Maritime Domain Awareness System

Multiplication of Radar Modules for Tracking Silent Vessels

Final Report

OSE4952-EEL4915 Senior Design II

Group 4

University of Central Florida
Department of Electrical and Computer Engineering
CREOL
Jet Chen, CpE
Alexander Drew, EE
Derek Florer, CpE
Gianna Montevago, PSE
Marco Plasencia, CpE
Garrett Miller, CS
Devin Parham, CS

Sponsored by United States Southern Command (SOUTHCOM)
Reviewers: Dr. Q.Z. Sun, Dr. Yannick Salamin, Dr. Heinrich, and Mr. Wubker
Mentors: Dr. Chan, Dr. Leinecker, Dr. Kar & Dr. Gerber

Table of Contents

Chapter 1: Executive Summary....1
Chapter 2: Project Description....3
2.1 Background....3
2.2 Review of Prior Related Work....4
2.3 Goals and Objectives....5
2.3.1 Basic Goals:....5
2.4 Project Features and Functionalities....7
2.4.1 Hardware....7
2.4.2 Software....8
2.4.3 Required Specifications....9
2.4.4 Hardware Block Diagram....11
2.4.5 Software Block Diagram....12
2.4.6 Optical System....13
2.5 Prototype Illustration....13
2.6 House of Quality....15
Chapter 3: Research and Part Selection....16
3.1 Radar Systems....16
3.1.1 Fundamentals of Radar Systems....16
3.1.1.1 Wavelength and Frequency Considerations....17
3.1.1.2 Sea State, Clutter, and Environmental Effects....17
3.1.1.3 Radar Mounting Height....18
3.1.2 Types or Radar....20
3.1.2.1 Comparison of Radar Technologies....20
3.1.2.2 Radar Form Factor Comparison....22
3.1.3 Current Radars on the Market....22
3.1.3.1 Furuno DRS6ANXT....23
3.1.3.2 Garmin GMR Fantom 126....23
3.1.3.3 Raymarine Cyclone Pro 6ft....23
3.1.3.4 Simrad HALO® 2006 Open Array....24
3.1.3.5 Raymarine Cyclone 6 ft....24
3.1.3.6 Comparison of Radar Models and Selection....25
3.2 Maritime Environmental Challenges....26
3.2.1 Stainless Steel (316)....27
3.2.2 Aluminum Alloys (5083 & 6061)....28
3.2.3 Fiber Reinforced Polymers....28
3.2.3.1 Fiber Reinforced Plastics Study....29
3.2.3.2 Epoxy-glass FRPs....29
3.2.3.3 Fiber Glass Reinforced Polyurethane....30

3.2.4 Material Comparisons....31
3.2.4.1 Weight vs Strength....32
3.2.4.2 Corrosion Resistance....32
3.2.4.3 Maintenance and Durability....32
3.2.4.4 Formability and Cost....32
3.2.5 Material Selection for MDAS Components....33
3.2.5.1 Main Outer Frame....33
3.2.5.2 Electronics Enclosure....33
3.2.5.3 Turbine Mounting Bracket Support....33
3.2.5.4 Base Plate....34
3.2.6 Protective Coatings....34
3.2.6.1 Epoxy Coatings....35
3.2.6.2 Polyurethane Coatings....35
3.2.6.3 Anti-fouling Coatings....36
3.2.6.4 Protective Coating Material Comparisons....36
3.2.7 Mounting Hardware....38
3.2.7.1 Flanged Mount....38
3.2.7.2 Cantilever Mount....39
3.2.7.3 Deck-penetrating Mount (Through-bolt Mount)....39
3.2.7.4 Mounting Hardware Comparisons....40
3.3 Optical Technology Substitution Feasibility....41
3.3.1 LiDAR Imaging System....41
3.3.1.1 LiDAR Imaging vs. Radar Imaging....43
3.3.2 Camera System....45
3.3.2.1 CCD Camera Sensors....46
3.3.2.2 CMOS Camera Sensors....49
3.3.2.3 CCD vs. CMOS Sensors....52
3.3.2.4 Camera vs. Radar....55
3.3.3 Fiber Optics Communication....55
3.3.3.1 Fiber Optics Communication vs. Radar....56
3.4 Additional Ship Identification System....56
3.4.1 Camera Sensor Selection....56
3.4.2 Lens System....58
3.4.2.1 Keplerian Telescope System....60
3.4.2.2 Galilean Telescope System....60
3.4.2.3 Achromatic Doublet Lens....60
3.4.2.4 Lens Systems Comparison....61
3.4.3 Adaptive Optics....61
3.4.4 Gimbal....64
3.4.5 Component Selection and Comparison....65
3.5 Embedded Systems....67

3.5.1 Single Board Computer....67
3.5.2 Programming Languages....69
3.6 Technology Stack....70
3.6.1 Frontend....71
3.6.1.1 Vue....71
3.6.1.2 Angular....72
3.6.1.3 React....72
3.6.2 Backend....72
3.6.2.1 Django....73
3.6.2.2 Nest....73
3.6.2.3 Express....74
3.6.2.4 AWS Lambda....74
3.6.3 Database....74
3.6.3.1 Redis....75
3.6.3.2 PostgreSQL....75
3.6.3.3 MongoDB....75
3.6.4 Server....76
3.6.4.1 AWS....77
3.6.4.2 DigitalOcean....77
3.6.4.3 Azure....77
3.6.4.4 Google Cloud....77
3.6.4.5 Network-Attached Storage....77
3.6.5 Internal API's....78
3.6.5.1 REST API....78
3.6.5.2 WebSockets....78
3.6.5.3 AWS API Gateway....79
3.6.6 External API's....79
3.6.6.1 Leaflet.js....79
3.6.6.2 Raymarine SDK....79
3.6.7 Authentication Methods for Maritime Web App....79
3.6.7.1 Password Hashing....80
3.6.7.2 JSON Web Token....80
3.6.7.3 OAuth 2.0....81
3.6.7.3 Multi-Factor Authentication....81
3.6.7.4 AWS Cognito....81
3.6.7.4.1 Comparison to Traditional Methods....82
3.6.7.4.2 Advantages of Using Cognito....82
3.6.7.5 Firebase....82
3.6.7.6 Auth0....83
3.7 Communication Protocols....84
3.7.1 NMEA 2000 and Marine Communication Protocols....84

3.7.2 Starlink Communication..... 87
3.7.3 Wireless Communication..... 89
3.7.3.1 Local Network Remote Access..... 89
3.7.3.2 Starlink Uplink for Cloud-Based Commands.....89
3.7.3.3 Gimbal Camera and Bluetooth Control.....90
3.7.3.4 Environmental Considerations..... 90
3.8.3 Wired Communication.....90
3.8.3.1 Radar System Integration..... 91
3.8.3.2 Starlink Mini Connectivity.....91
3.8.3.3 Serial Peripheral Interfaces.....91
3.8.3.4 I2C..... 92
3.8.3.5 UART..... 92
3.8.3.6 SPI..... 93
3.8.4 Serial Communication Usage.....93
3.9 Analysis of the Power System.....94
3.9.1 Implementation of a Wind Powered System.....96
3.9.2 Comparison of Vertical and Horizontal Wind Turbines.....98
3.9.3 Wind Turbine Part Selection..... 99
3.9.3 Charge Controller Selection.....104
3.9.4 Power Storage and Battery Selection.....106
3.9.5 Identification and Selection of Regulators.....109
3.9.6 System Voltage Comparison..... 111
3.9.6.1 12V DC System..... 111
3.9.6.2 24V DC System..... 111
3.9.6.3 AC System..... 111
3.9.6.4 System Voltage Conclusion..... 112
3.9.7 Expected System Strain.....113
3.10 Machine Learning Techniques.....115
3.10.1 Gradient Boosting..... 115
3.10.2 Density-Based Spatial Clustering of Applications with Noise (DBSCAN)..... 116
3.10.3 Random Forest..... 117
3.11 Classification Algorithms..... 118
3.11.1 Multi-Feature Ensemble Learning Classification Method (MFELCM)..... 118
3.11.2 LightGBM.....121
3.12 Anomaly Detection Algorithms..... 125
3.12.1 Traffic Route Extraction and Anomaly Detection (TREAD)..... 125
Chapter 4: Standards and Design Constraints.....128
4.1 Standards..... 128
4.1.1 FCC 47 CFR Part 80 - Maritime Services and Radar Compliance.....129
4.1.2 IEEE C95.1-2019 - Electromagnetic Radiation Safety.....130
4.1.3 IEEE 1547-Inspired Considerations for Off-Grid Power Integration Using Wind

Turbines....132
4.1.4 Code of Federal Regulations - Automatic Identification System....133
4.1.5 Waterproofing and Ingress Protection....134
4.2 Constraints....135
4.2.1 Environmental Factor Constraints....135
4.2.2 Weight and Form Factor Constraints....136
4.2.3 Raspberry Pi Constraints....137
4.2.5 Budget Constraints....138
4.2.6 Power Generation and Energy Storage Constraints....138
4.2.7 Lightning and Surge Protection....139
4.2.8 Cyber Security Constraints....140
Chapter 5: Comparison of ChatGPT with other Similar Platforms....141
5.1 LLM Comparison....141
5.1.1 Case Studies - ChatGPT....142
5.1.1.1 Embedded Systems Perspective....142
5.1.1.2 Optical System Perspective....142
5.1.2 Case Studies - Gemini....143
5.1.2.1 Radar Perspective....143
5.1.2.2 Material Section Perspective....144
5.1.3 Case Studies - Deep Seek....145
5.1.3.1 Turbine Perspective....145
5.1.4 Case Studies - Claude....146
5.1.4.1 Web App Perspective....146
5.2 Project Impact....146
5.2.1 Benefits....147
5.2.2 Detriments....148
5.3 Conclusion....149
Chapter 6: Hardware Design....151
6.1 Power System Design....151
6.1.1 Understanding the Implementation of a DC System....153
6.1.1.1 Generating Usable Power....156
6.1.1.2 Utilizing Regulators to Supply Power....157
6.1.1.3 Wind Turbine Implementation....159
6.2 Sensing Hardware: Raymarine Radar....160
6.3 Onboard Processing Unit: Intel NUC....160
6.4 Satellite Communication Unit: Starlink Mini....160
6.5 System Monitoring Accessory: Smartphone Interface....160
6.6 Wiring and Network Adapters....161
6.6.1 Power Wiring Considerations....161
6.6.2 Network Connections and Adapters....162
6.7 Imaging System....162

6.7.1 Lens System..... 163
6.7.2 Gimbal..... 164
Chapter 7: Software Design..... 166
7.1 Dark Vessel Classification Algorithm..... 166
7.1.1 Automatic Identification System..... 168
7.1.2 Retrieving AIS Data..... 168
7.1.3 Parsing and Formatting of Data..... 169
7.1.4 Subtracting Known Vessels..... 170
7.1.5 Classifying Dark Vessels..... 170
7.1.6 Known Challenges..... 171
7.1.7 Alternative Applications..... 172
7.2 Use Case Diagram..... 173
7.3 Activity Diagram..... 174
7.4 Component Diagram..... 175
7.5 UI Design Goals..... 176
7.5.1 Login Display..... 177
7.5.2 Homepage Design..... 178
7.6 Security Design Goals..... 179
7.7 Scalability and Maintainability of Maritime Web App..... 180
7.8 Web App Plans and Goals..... 181
7.8.1 Short-Term Goals..... 181
7.8.2 Mid-Term Enhancements..... 181
7.8.3 Long-Term Expansion..... 182
7.9 Embedded System Software Design..... 182
7.9.1 Software Architecture..... 182
7.9.2 Flow of Execution..... 183
7.9.3 Data Structures..... 184
7.9.4 Communication Handling..... 184
7.9.5 State Machine for Anomaly Detection..... 184
7.9.6 Failsafes and Watchdog..... 185
Chapter 8: System Fabrication/Prototype Construction..... 185
8.1 MDAS System Fabrication..... 186
8.1.1 MDAS System Prototype..... 186
8.1.2 MDAS System Redesign and Structural Revision..... 189
8.1.3 MDAS System Structural Material Design..... 191
8.2 Custom 12DC to VCM Cable Design..... 192
8.2.1 Cable Fabrication Process..... 192
8.2.2 Observations and Challenges..... 193
8.3 Optics System..... 193
8.3.1 Lens System..... 193
8.3.2 Gimbal..... 196

Chapter 9: System Testing and Evaluation.....197
9.1 Prototype Setup..... 197
9.1.1 NUC-Based Prototype Platform..... 197
9.1.1.1 Software Setup.....198
9.1.1.2 Account and Credential Management..... 198
9.1.2 HP Base Station Integration.....198
9.1.2.1 Software Setup.....199
9.1.2.2 Port-Forwarding and Wireguard.....199
9.1.3 Starlink Mini Satellite Link..... 199
9.1.3.1 Subscription Setup.....200
9.1.3.2 Physical Connection..... 200
9.1.4 Raymarine Cyclone Radar + VCM100.....200
9.1.4.1 Power Path.....201
9.1.4.2 Ethernet/ DHCP Connection..... 201
9.2 Hardware Testing.....201
9.2.1 Radar Power Integration and Troubleshooting..... 202
9.2.2 Starlink Connectivity Testing.....203
9.3 Software Testing.....204
9.3.1 Radar Command Interface and SDK Testing.....204
9.3.2 Starlink Communication and Data Exchange..... 207
9.3.3 Web Application Testing.....209
9.4 Optical Feasibility and Testing..... 211
9.4.1 Lens System..... 211
9.4.2 Angular Resolution..... 215
9.4.3 Senior Design 2 Testing..... 216
9.5 Overall Integration.....218
9.6 Plan for SD2..... 218
Chapter 10: Administrative Content.....220
10.1 Budget.....220
10.2 Bill of Materials.....220
10.3 Distribution of Workload.....222
10.4 Project Milestones..... 223
Chapter 11: Conclusion..... 225
Appendices.....227
Appendix A - References..... 227
Appendix B - Copyright Permission..... 236
Appendix C - Data Sheet.....236
Appendix D - Software Code..... 240
Appendix E - ChatGPT prompts..... 259

List of Tables

Table 2.1 - Engineering Specifications..... 9

Table 2.2 - Optical Engineering Specifications.....10

Table 2.3 - House of Quality..... 15

Table 3.1 - Radar Technology Comparison.....21

Table 3.2 - Radar Form Factor Comparison.....22

Table 3.3 - Radar Part Comparison..... 25

Table 3.4 - Material Comparison for MDAS Structural Components..... 31

Table 3.5 - Protective Coating Comparisons.....36

Table 3.6 - Mounting Hardware Comparisons..... 40

Table 3.7 - Radar vs LiDAR comparison.....43

Table 3.8 - CCD Comparison.....48

Table 3.9 - CMOS Comparison.....51

Table 3.10 - CCD vs CMOS Comparison..... 54

Table 3.11 - Mounting Camera Comparison..... 58

Table 3.12 - Keplerian vs Galilean Comparison..... 61

Table 3.13 - Adaptive Optics Comparison..... 63

Table 3.14 - CMOS Sensor Comparison..... 65

Table 3.15 - Gimbal Comparison..... 66

Table 3.16 - Lens System Comparison..... 66

Table 3.17 - SBC Comparison..... 68

Table 3.18 - Comparison of Programming Languages..... 70

Table 3.19 - Frontend Technology Comparison.....71

Table 3.20 - Backend Technology Comparison..... 73

Table 3.21 - Database Technology Comparison.....74

Table 3.22 - Server Technology Comparison.....76

Table 3.23 - Authentication Technology Comparison..... 80

Table 3.24 - Key Fields of NMEA 2000 Message..... 86
Table 3.25 - Satellite Comparison..... 88
Table 3.26 - Environmental Considerations..... 90
Table 3.27 - Serial Communication Usage..... 93
Table 3.28 - Renewable Energy Comparison..... 97
Table 3.29 - Wind Turbine Mounting Comparison..... 99
Table 3.30 - Regulator Comparison..... 103
Table 3.31 - Wind Turbine Comparison..... 105
Table 3.32 - Battery Comparison..... 108
Table 3.33 - Regulator Comparison..... 110
Table 3.34 - System Voltage Comparison..... 112
Table 3.35 - Maritime Radar System Power Requirements Analysis..... 114
Table 3.36 - System Power Totals..... 115
Table 7.1 - AIS and ARPA Data..... 169
Table 7.2 - Proposed Modules..... 182
Table 9.1 - Zemax Lens Testing..... 213
Table 10.1 - Bill of Materials..... 220
Table 10.2 - Workload distribution..... 223
Table 10.3 - Project milestone plan..... 224

List of Figures

Figure 2.1 - Hardware block diagram..... 11

Figure 2.2 - Software block diagram..... 12

Figure 2.3 - Optical Sensor Concept Diagram..... 13

Figure 2.4 - MDAS Prototype CAD [93]..... 14

Figure 3.1 - Radar vs LiDAR data comparison (61)..... 43

Figure 3.2 - How cameras create a digital image (64)..... 45

Figure 3.3 - Front v back illumination v stacked (68)..... 51

Figure 3.4 - NMEA 2000 Network Topology..... 85

Figure 3.5 - NMEA 2000 message (PGN and Data Fields)..... 86

Figure 3.6 - Real-world NMEA 2000 Data (Fast Packet Trace example)..... 86

Figure 3.7 - MFELCM Machine Learning Flowchart..... 119

Figure 3.8 - Process of dynamic features extraction..... 120

Figure 3.9 - Performance evaluation of base classifiers and MFELCM..... 120

Figure 3.10 - (a) Confusion matrix of samples (MFELCM); (b) Confusion matrix of ships (MFELCM)..... 121

Figure 3.11 - Dataset Preprocessing..... 122

Figure 3.12 - LightGBM Machine Learning Flowchart..... 122

Figure 3.13 - LightGBM Classification Results Bar Chart..... 123

Figure 3.14 - LightGBM Classification Confusion Matrices..... 124

Figure 3.15 - Comparison of MFELCM (Literature) and LightGBM (Comparative) Performance Metrics..... 124

Figure 3.16 - TREAD Process Flowchart..... 125

Figure 3.17 - (a) Color-coded routes (b) showing potential outliers (red) (c) Kernel Density Estimation..... 126

Figure 3.18 - TREAD Probability of Anomalous Behavior..... 127

Figure 6.1 - MDAS Radar System Diagram..... 154

Figure 6.2 - Power System to deliver DC Voltage to the Pi MCU and Starlink Mini Terminal... 158

Figure 6.3 - Optical System Layout..... 163

Figure 6.4 - Zemax Data..... 163
Figure 6.5 - Gimbal Apparatus..... 165
Figure 7.1: Dark Vessel Classification State Diagram..... 167
Figure 7.2- Use Case Diagram for Maritime Web Application..... 173
Figure 7.3 - Activity Diagram for Maritime Web Application..... 174
Figure 7.4 - Component Diagram for Maritime Web Application..... 175
Figure 7.5 - Login Display for Maritime Web Application..... 177
Figure 7.6 - Homepage Display for Maritime Web Application..... 178
Figure 7.7 - Embedded System Software Logic..... 183
Figure 8.1 - MDAS Initial Prototype Sketch..... 187
Figure 8.2 - MDAS Initial Prototype CAD..... 188
Figure 8.3 - MDAS 5 Beam Frame CAD..... 189
Figure 8.4 - Iterative CAD models showing the step-by-step addition of MDAS system components..... 190
Figure 8.5 - 12VDC to VCM fabricated cable..... 193
Figure 8.6 - Prototype Lens System v1 and v2..... 194
Figure 8.7 - Lens Tube Models..... 195
Figure 8.8 - Printed Lens Tubes..... 195
Figure 9.1 - Complete Prototype Setup..... 196
Figure 9.2 - Multimer Reading 12VDC Input and ~0VDC Output..... 201
Figure 9.3 - Multimer Reading ~42VDC with VCM100 Green ‘ON’ LED..... 202
Figure 9.4 - Starlink Mini Terminal..... 202
Figure 9.5 - DHCP Server With Radar Populated..... 203
Figure 9.6 - Quantum Interface, CLI from Raymarine SDK With Radar’s Instantiation..... 204
Figure 9.7 - Wireshark Packets When Connected to Radar..... 204
Figure 9.8 - ePowerDown Command Sent to Radar Through Quantum Interface..... 205
Figure 9.9 - Wireshark Transmission After Radar Was Sent Power Down Command..... 205
Figure 9.10 - eWakeUp Command Sent to Radar Through Quantum Interface..... 205
Figure 9.11 - Wireshark Transmission After Radar Was Sent to Wake Up..... 206

Figure 9.12 - Tailscale Interface With NUC and HP Base Station Connected..... 207
Figure 9.13 - Postman API Login Test.....208
Figure 9.14 - Login Page Prototype..... 209
Figure 9.15 - Dashboard Map Prototype.....209
Figure 9.16 - Original Image Test..... 211
Figure 9.17 - Final Image Test..... 213
Figure 9.18 - Test Image (left) vs. Intensity Plot (right)..... 215
Figure 9.19 - Test Image at 10 Feet.....216
Figure 9.20 - Final Test Image..... 217

Chapter 1: Executive Summary

Modern maritime law enforcement is confronted with a growing challenge: vessels operating outside of legal oversight by turning off or simply not having Automatic Identification System (AIS) transponders. These "dark ships" are being utilized more frequently for purposes such as narcotics trafficking, unauthorized immigration, and illegal fishing, and their communications unavailability renders most conventional surveillance techniques futile. Though AIS was established to enhance navigation safety and situational awareness, it is by design contingent on the voluntary cooperation of the vessel. When such collaboration falters, visibility also falters. Accordingly, huge areas of the open sea become blind spots where illicit activity flourishes. The issue is not one of insufficient technological resources; it is one of the fact that the technologies we depend on were not created to handle non-compliance well.

To overcome this, our team set out to develop the Maritime Domain Awareness System (MDAS), a standalone, low-cost, and scalable system that modernizes maritime surveillance. Unlike conventional approaches, which rely on government-run patrols and large-scale infrastructures, MDAS uses available commercial maritime traffic to its advantage. Instead of adding fleet capacity or buying expensive infrastructure, we propose to install small, weather-tolerant radar nodes on commercial ships and cargo vessels navigating the seas. The modular nodes will be completely powered off-grid from wind energy and onboard energy storage systems, with long-term performance and low maintenance. The system consists of an advanced marine radar unit that detects vessels over a large area of operation, together with a low-power processor unit dedicated to radar preprocessing and filtering, as well as satellite communication unit that enables data to be transmitted in real time even in remote maritime locations.

When it receives returns from radar, the information is routed to a base station and an AIS subtraction algorithm is employed to remove cooperative traffic. The outcome is an improved list of likely silent ships which by intention or design, do not broadcast their identity. Those targets, when found, would be updated on a web application almost immediately and following action to shore-based operators can take their ideal next step. The ability to screen and classify suspect ships remotely cuts down the incidence of false positives considerably and dispenses with the necessity for constant patrol. Equally significant, every radar unit possesses the capability for remote configuration via software command. Operators at the base station have the ability to modify scan intervals, power settings, or place a unit in a standby mode, all while conserving energy within a resource-limited context. This implies that during periods of inactivity, the system conserves energy but can be activated at will.

The power of MDAS lies not so much in what it does, but how it does it. It is self-sustaining. It is portable. It hopes to convert the global merchant fleet into an active network of peer-to-peer radar units, with each unit contributing to a better overall picture of real-time maritime activity. As the number of MDAS units deployed grows, so does the degree of coverage, fidelity of information, and reliability. In a context in which unlawful maritime activity is ever more difficult to detect, MDAS does not try to outmatch the sophistication of the problem; rather, it tries to work around it. The system

we aim to developed does not require the utilization of a naval vessel for target identification. It does not even depend on satellites being utilized to find irregularities. Instead, it needs no more than wind from maritime travel, an open horizon, and snug mounting to become active.

The MDAS architecture is composed of four integrated subsystems: the radar unit, the onboard computing platform, the communication uplink, and the inland base station. Each maritime node hosts a Raymarine Cyclone radar connected to an Intel NUC that executes command logic and telemetry processing. A Tailscale-enabled Starlink Mini uplink provides secure global connectivity between the node and the base station, allowing real-time data transmission. Power is supplied by a wind turbine and onboard energy storage system to ensure autonomous, off-grid operation. Onshore, the base station consists of a server laptop running a web-based application where operators can log in, receive messages, issue commands, and eventually visualize radar returns. The system supports modular configuration, multiple nodes can be deployed across a fleet, with each contributing independently to situational awareness. This distributed model ensures coverage is scalable, redundant, and adaptable to changing mission profiles.

The report begins by establishing the scope of the maritime surveillance problem and defining the role MDAS aims to fulfill within that context. It then outlines the system architecture, detailing how each major component, from radar hardware and embedded processing to satellite communication and base station infrastructure would ideally work simultaneously. Design constraints such as regulatory compliance, off-grid power, and environmental durability are considered alongside functional requirements like real-time communication and operator control. The document moves on to describe the selection, configuration, and integration of hardware components, followed by a breakdown of the software architecture that handles radar control and remote communication through Starlink. Prototype construction and testing efforts are documented to demonstrate the feasibility of each subsystem, while analysis of the current system integration highlights remaining challenges and outlines a clear path forward for full deployment. Ultimately, MDAS represents a major leap in the deployment of maritime security strategies, moving from a paradigm of reactive enforcement to one marked by proactive and distributed situational awareness.

Chapter 2: Project Description

2.1 Background

The Importance of Maritime Border Security

The security and stability of the United States depend heavily on protecting its maritime borders. With over 95,000 miles of coastline and extensive shipping routes, the nation's waters serve as vital channels for trade, transportation, and national defense. However, these same waters are also exploited for illicit activities such as drug trafficking, human smuggling, illegal fishing, and unauthorized vessel entry. Traditional maritime security efforts rely on government-operated naval assets, but these resources are often stretched thin across vast oceanic regions, creating blind spots where illegal activities can go undetected.

To address these challenges, the U.S. has implemented Maritime Domain Awareness (MDA) initiatives, integrating surveillance technologies to monitor vessel activity in real time. The Maritime Domain is defined as "*All areas and things of, on, under, relating to, adjacent to, or bordering on a sea, ocean, or other navigable waterway, including maritime-related activities, infrastructure, people, cargo, and vessels and other conveyances."* The Maritime Security Policy established the first-ever Maritime Security Policy Coordinating Committee to oversee the development of NSMS and eight supporting plans-the main one being a national plan to achieve MDA. This plan lays the foundation for all matters related to the maritime domain that could impact the security, safety, economy, or environment of the United States and also focuses on identifying threats as early and as distant from our shores as possible. A key component of MDA is the Automatic Identification System (AIS), which requires vessels to transmit their identity, position, speed, and course to prevent collisions and enhance navigational safety. AIS allows port authorities, coast guards, and naval forces to track legitimate vessels; however, ships engaging in illicit activities frequently disable or manipulate their AIS transponders to evade detection. This creates a significant gap in maritime security, as vessels without AIS cannot be easily identified or tracked using conventional methods.

Project Motivation and Cost-Saving Measures

Given the limitations of current MDA strategies, there is a critical need for a cost-effective, scalable solution to detect and monitor non-cooperative vessels operating without AIS. Our project aims to deploy self-sustaining maritime surveillance units that integrate radar technology to track and transmit data on vessels operating without AIS transponders. These units will filter out compliant ships and focus on identifying unregistered or suspicious vessels, relaying this information to an inland server for analysis and military response.

A major innovation of this approach is leveraging commercial vessels as hosts for these radar systems. Rather than relying solely on government ships or satellite-based coverage, which are costly to operate and limited in number, deploying radar-equipped units on commercial vessels expands the military's radar network while significantly reducing operational costs. This not only increases the density of surveillance coverage

but also ensures a more persistent and widespread monitoring capability. A critical aspect of this system is its ability to function independently from the host vessel. The radar unit is designed to be self-powered, ensuring minimal impact on the host's regular operations. The radar system is designed to be non-intrusive, increasing the acceptance from commercial operators to allow radar integration on maritime vessels. By integrating advanced radar systems with existing commercial fleets, this project enhances border security, resource efficiency, and military intelligence capabilities seamlessly without requiring additional government-operated vessels.

2.2 Review of Prior Related Work

To learn more about this project, research was conducted online to understand the requirements of similar solutions and gain insight into how to complete the objectives. This research led to a notable example of another commercial off-the-shelf (COTS)-based radar system. A surveillance systems company, Surface Search Radar Engineering, designed the AN/SPS-78 radar system to replace the aging AN/SPS-73 systems aboard U.S. Coast Guard vessels. Their project addresses reliability and obsolescence issues and details commercial requirements for the International Maritime Organization, along with capabilities necessary for tactical missions required by coast guards and navies. The new AN/SPS-78 is fully compatible with the U.S. Coast Guard's SeaWatch—an enterprise command, control, and navigation system—and other naval systems. This prior work was extremely helpful in explaining how to create a radar processing and tracking system that meets commercial and tactical needs. Open-source information from this project will assist in development efforts.

Another discovery was ThayerMahan's 'SeaPicket' system, an energy-harvesting, autonomous surveillance platform that integrates acoustic, radar/EW, video, and AIS sensors. SeaPicket is designed for long-term deployment and offers persistent maritime domain awareness by collecting and processing data on vessel movements. This system uses artificial intelligence and machine learning to analyze sensor data and provide real-time insights to port security organizations and field commanders. Overall, this project illustrates the integration of multiple sensors in a modular design with autonomous operation, which has been beneficial in understanding the goal of developing a plug-and-play MDA system. Similar energy-harvesting techniques and data analytics methodologies will be incorporated into the design to enhance system autonomy and efficiency.

Lastly, Kelvin Hughes' SharpEye S-Band Radar was examined due to its relevant characteristics. This commercially available solid-state radar system is designed for MDA and navigation. It is used on a variety of naval, Coast Guard, and commercial vehicles due to its low power consumption, high reliability, and advanced tracking capabilities. SharpEye uses Doppler processing, which detects and classifies targets based on their movement patterns, helping differentiate between stationary, slow, and high-speed vessels. As a result, clear radar imaging can be achieved even in heavy rain or rough seas, making it ideal for persistent MDA applications. The system also eliminates the need for high-voltage components, significantly reducing maintenance requirements. Studying this work provided valuable insights into how radars process data, optimize

power efficiency, and operate in challenging conditions. Additionally, it highlighted the potential for Doppler-based tracking to enhance real-time detection of non-AIS vessels. These techniques establish a strong baseline for design choices in the development of a radar-based MDA system.

2.3 Goals and Objectives

The self-contained radar system is designed to enhance maritime domain awareness by addressing critical challenges such as detecting non-AIS vessels, operating independently in harsh marine environments, and maintaining reliable long-range communication. By prioritizing portability, self-sustainability, and affordability, the system will enable continuous monitoring and real-time data transmission.

Through a combination of hardware and software innovations, the MDAS aims to create a durable and efficient solution for vessel detection and tracking in remote maritime regions. The following goals and objectives outline the key functionalities required to bring this system to completion.

2.3.1 Basic Goals:

- Develop a self-sufficient radar system that does not rely on the ship's power supply.
- The system must detect vessels transmitting AIS, and vessels without AIS transmission.
- Establish a reliable method for transmitting data from the radar system to the base station.
- Design the system to operate autonomously for long periods with minimal maintenance.
- Make the system lightweight and easy to install on various maritime setups.
- Provide on/off power control for the radar via a physical switch and remotely through Starlink from a remote device.
- Develop a user-friendly interface that integrates radar overlays with AIS Marine Tracker.
- Attach an enhancing optical system to a camera to increase overall performance at distance.
- Use a gimbal-mounted camera system to take long-distance photos of potential threats.

Advanced Goals:

- Enable continuous data transmission from radar ensuring live updates without delay.
- Reduce power consumption by turning off the radar when ported.
- Allow the radar to operate intermittently rather than continuous operation.
- Improve the lens attachment for better overall performance of the optical system.

Stretch Goals:

- Ensure expandability by allowing multiple modules to connect, improving load balancing and synchronized data processing.
- Obtain processable images in addition to the visible light camera for operation in moonlight conditions.
- Research and integrate adaptive optics into the system.

Hardware Objectives:

- Construct a self-sustaining power source with integration capabilities for wind energy.
- Utilize a microcontroller to analyze radar data and process it for transmission back to base station via Starlink protocol.
- Integrate a GPS module with the microcontroller to accurately slew the camera towards targets.
- Configure the radar to enter a low-power standby mode when not actively in use.
- Ensure the use of sturdy, waterproof, and corrosion-resistant materials to protect the radar module and connecting wires.
- Include a physical power switch that allows the module to be turned on or off by the operator.
- Select a CMOS sensor optimal for the operating distance and ambient conditions.
- Design and implement a Galilean lens system for fixed magnification.
- Integrate a gimbal into the optical system to provide a larger range of motion.

Software Objectives:

- Develop an algorithm to process radar data and subtract AIS transponder information, isolating non-AIS targets.
- Create a user-friendly interface for monitoring and control, integrating AIS Marine Tracker with radar overlays.

- Develop a server accessible via Starlink to send commands that remotely power the radar module on and off.
- Enable the server to remotely send commands to the mounted camera for capturing images on demand.
- Enable automatic radar shutdown when the vessel is docked and restart upon departure.

Stretch Goal Objectives:

- Sustain a scalable module database with intra-connected communications allowing for radar proximity efficiency.
- Integrate an additional infrared camera capable of operating with low light for target identification.
- Integrate adaptive optics technologies to be able to see targets clearly at the horizon.

2.4 Project Features and Functionalities

In this chapter, we describe the inherent qualities and processes that facilitate the standalone and dependable operation of the radar unit in remote maritime conditions. By explaining the hardware, software, and optical subsystems, we demonstrate how each of the components plays a role in the overall efficacy of the system ultimately delivering sensing functionality, power efficiency, and effective data communication. The subsequent subsections offer an overview of how these components integrate to form a compact, autonomous radar solution that is ideal for long-term marine environment deployment.

2.4.1 Hardware

The hardware components of the self-contained radar unit are carefully selected to meet the project's requirements of compactness, low power consumption, and minimal maintenance. Each component plays a crucial role in ensuring the system's functionality, self-sufficiency, and long-term reliability in a marine environment. At the core of the system is the marine radar, which provides the primary sensing capability, detecting objects and mapping the surrounding environment. Complementing this is a camera mounted on a gimbal, which captures visual data of either the radar unit itself or nearby objects of interest using its gimbal. Together, these components allow for a comprehensive understanding of the surroundings, with the radar providing detection and the camera offering visual verification.

To power the system sustainably, a wind turbine is employed to generate renewable energy. This energy is regulated by a charge controller, which ensures that the current fed into the battery remains within safe limits, protecting the battery from overcharging or damage. The battery serves as the central energy storage unit, supplying power to all system components when renewable sources are insufficient or unavailable.

To meet the diverse voltage requirements of the system, DC-DC regulators are used to step up or step down the battery's output to provide stable power to each component. This includes powering the microcontroller, which serves as the system's brain, and managing data from the radar, camera, and other peripherals. The microcontroller is also responsible for facilitating communication between the radar data and external devices via cellular connectivity, ensuring that the data is transmitted to both servers and mobile devices for remote monitoring and analysis.

Additionally, the system includes a mobile device paired with the radar apparatus, acting as a local on-off switch for the entire system. This allows for on-site control and rapid deployment adjustments. To enhance adaptability, a versatile mounting system has been designed to house all components in a relatively compact manner while ensuring compatibility with various ships, allowing for flexible installation across different marine vessels.

Each of these components interacts seamlessly to create an efficient system. The wind turbine and charge controller feed renewable energy into the battery, which acts as a central hub of power distribution. The DC-DC regulators ensure that each component receives the appropriate voltage for operation. The microcontroller acts as the central coordinator, handling data processing and communication while the radar and camera perform their respective sensing tasks. This carefully designed interaction between components ensures that the radar unit remains operational over extended periods, even in remote locations, while maintaining its self-contained nature.

2.4.2 Software

The software for the self-contained radar unit is designed to enable remote monitoring, control, and data processing while ensuring seamless operation with minimal user intervention. At its core, the system continuously transmits the GPS location of the radar node to a central server, allowing users to track its position in real time. When needed, users can send commands from the server to turn the radar on or off, enabling remote control of power usage and system operation. The system is integrated with a bay station database that plays a crucial role in managing radar operations and data storage.

The bay station database is responsible for issuing instructions to radar units to turn on or off as required for data collection, allowing for efficient control over the system. Once activated, the radar collects raw radar data and transmits it to the server via a Starlink connection, ensuring low-latency and reliable data delivery, even in remote maritime locations. The software then employs an AIS subtraction algorithm to filter out all vessels that are broadcasting AIS signals, ensuring that only non-AIS vessels remain in the collected data set. This improves detection accuracy and operational efficiency while reducing false positives.

The processed radar data is then securely stored in the database for further analysis, long-term tracking, and potential integration with other maritime monitoring systems. Users can remotely trigger the system to capture a camera image for additional verification. The image can be used to visually inspect either the radar unit itself—confirming operational status—or the area where an unidentified vessel was

detected, providing potential visual confirmation of the target.

The software integrates these functions into a streamlined workflow, ensuring efficient communication between the radar node and the bay station database while enabling users to monitor and control the system remotely with minimal latency.

2.4.3 Required Specifications

The specifications for the MDAS are divided into subsections each detailing specific aspects of the system's requirements and implementation.

Table 2.1 *- Engineering Specifications*

Portability	
Total Weight	≤ 60kg
Enclosure Volume	6.5" by 6.5" by 5"
Radar	
Detection Range	50 to 60 nautical miles (57.5 to 69 miles)
Data Transmission Rate	2 to 12 times per minute
Maximum Targets Tracked	20 to 40 Vessels
Full System	
Total Wind Energy Production	400 to 500 Watts
Power Consumption	≤ 275 Watts
Maximum Voltage	48 V
Cold Start Time	≤ 120 seconds
Operating Temperature	5°C to 50°C
Response Time (Radar Module to Base Station PC)	≤ 3 minutes
Maintenance Interval	~6 months
Total Cost	≤ $15,000
Software/ Communication	
AIS Filtering Algo Accuracy	≥ 99% accurate
Remote Shutdown Latency	≤ 20 second

Optical System	
Field of View (H/V)	0.74/0.50 degrees
Optimal Operating Distance	~ 5 M
Magnification	3-5x

Table 2.2 - *Optical Engineering Specifications*

Component	**Parameter**	**Specification**	**Unit**
Allied Vision Alvium 1800 U-050c, 1/3.6" 0.5MP C-Mount, CMOS Sensor	a) Sensor Size b) Pixel Size c) Sensor Type d) Shutter Type e) Resolution	a) 33 x 29 x 29 b) 4.8 c) CMOS d) Global e) 0.50	a) mm b) um c) — d) — e) Megapixels
AC254-300-A Achromatic Doublet Lens	a) EFL b) Diameter c) Radius of Curvature d) Coating e) Material	a) 250 b) 25.4 c) 165.2, -137.1, -557.4 d) AR coated e) N-BK7/SF2	a) mm b) mm c) mm d) — e) —
FeiyuTech SCORP-C Three-Axis Gimbal	a) Pan Range b) Tilt Range c) Roll Range	a) 360 b) +120 ~ -201 c) +20 ~ -20	a) Degress b) Degrees c) Degrees

2.4.4 Hardware Block Diagram

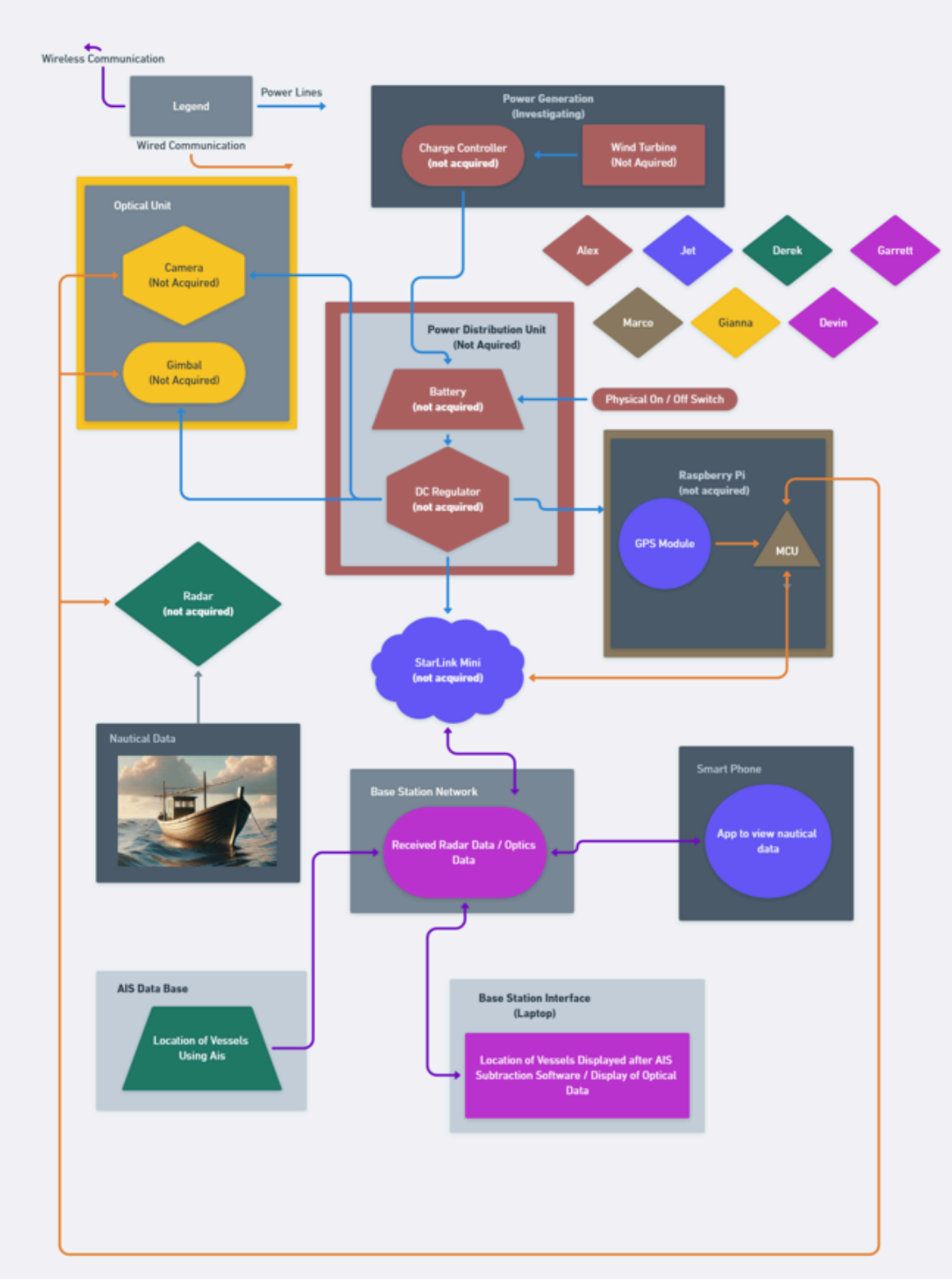


Figure 2.1 *- Hardware block diagram*

2.4.5 Software Block Diagram

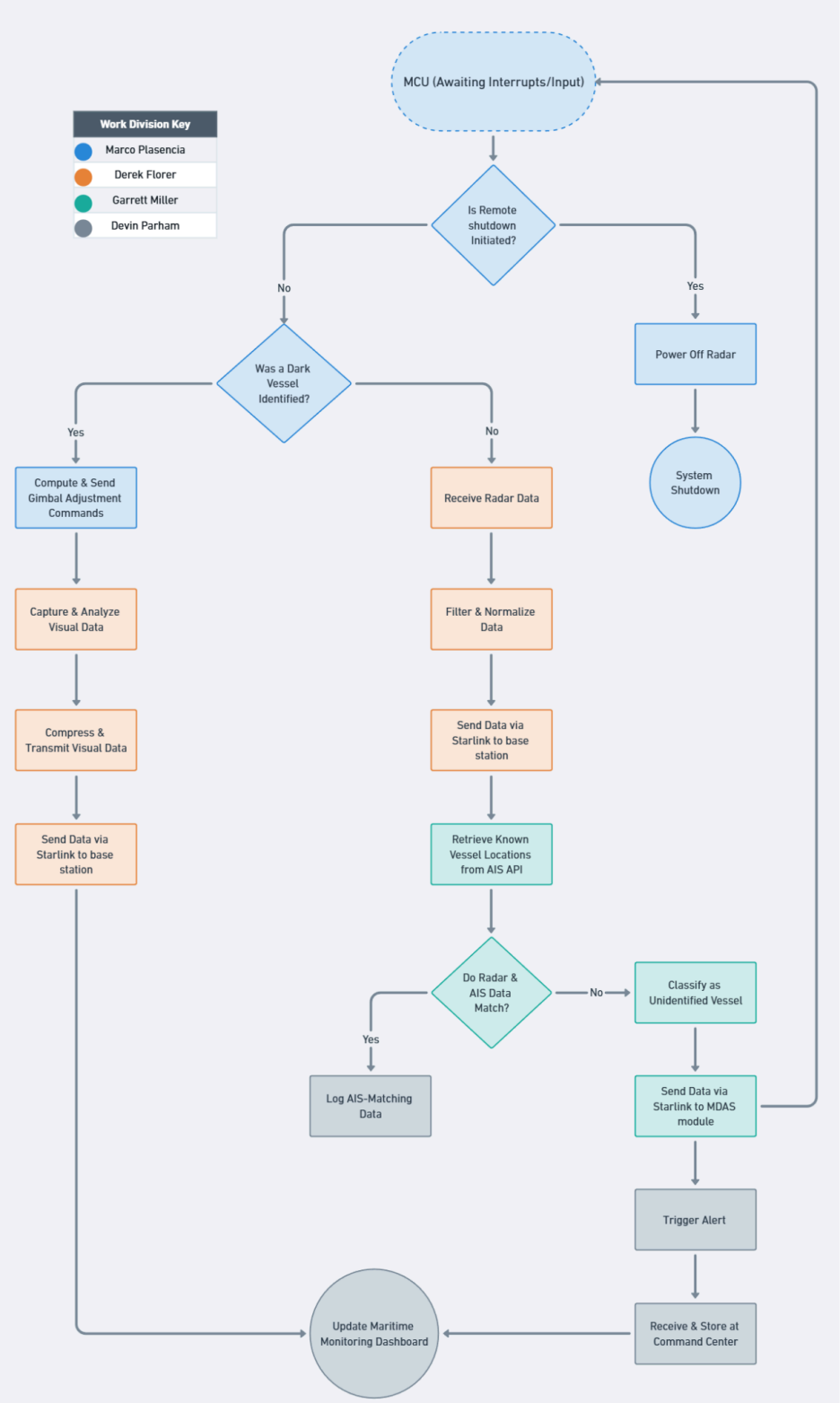


Figure 2.2 *- Software block diagram*

2.4.6 Optical System

In addition to the included radar system, we are also integrating an optics system. We are implementing a long-distance camera system that can help with the identification of neighboring ships that may be encountered on the open ocean. The camera will be a commercially available CMOS camera, mounted on a gimbal for an increased range of motion. A fixed telescope lens system will be attached to the front of the camera to increase magnification and image quality. This system will be an achromatic doublet lens, which decreases spot size. This system will help the performance increase at long distances.

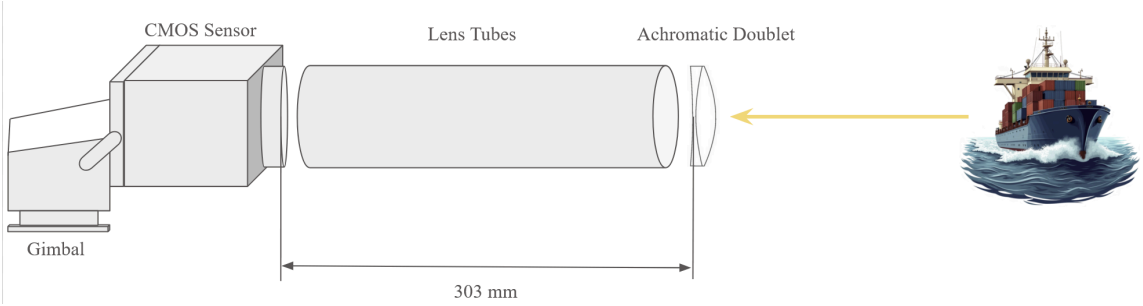


Figure 2.3 *- Optical Sensor Concept Diagram*

2.5 Prototype Illustration

The Computer-Aided Design (CAD) model in Figure 2.4 displays the current prototype configuration of the Maritime Domain Awareness System (MDAS). The compact, self-contained unit combines a number of important components to provide correct operation in isolated maritime locations. A Raymarine Cyclone radar is situated at the top of the assembly, guarded by a radome that encloses the radar and protects it from the elements without compromising signal integrity. Beneath the radar is a vertically mounted wind turbine, housed in a mesh cylindrical structure that provides ongoing airflow, while simultaneously shielding the turbine from potential debris.

Mounted on the side of the frame is a sealed protective cover that encloses major electronic components. This includes a Raspberry Pi, battery for power storage, and other electrical components necessary for operation. The design highlights durability and simplicity ultimately ensuring that the system can operate autonomously in various maritime conditions.

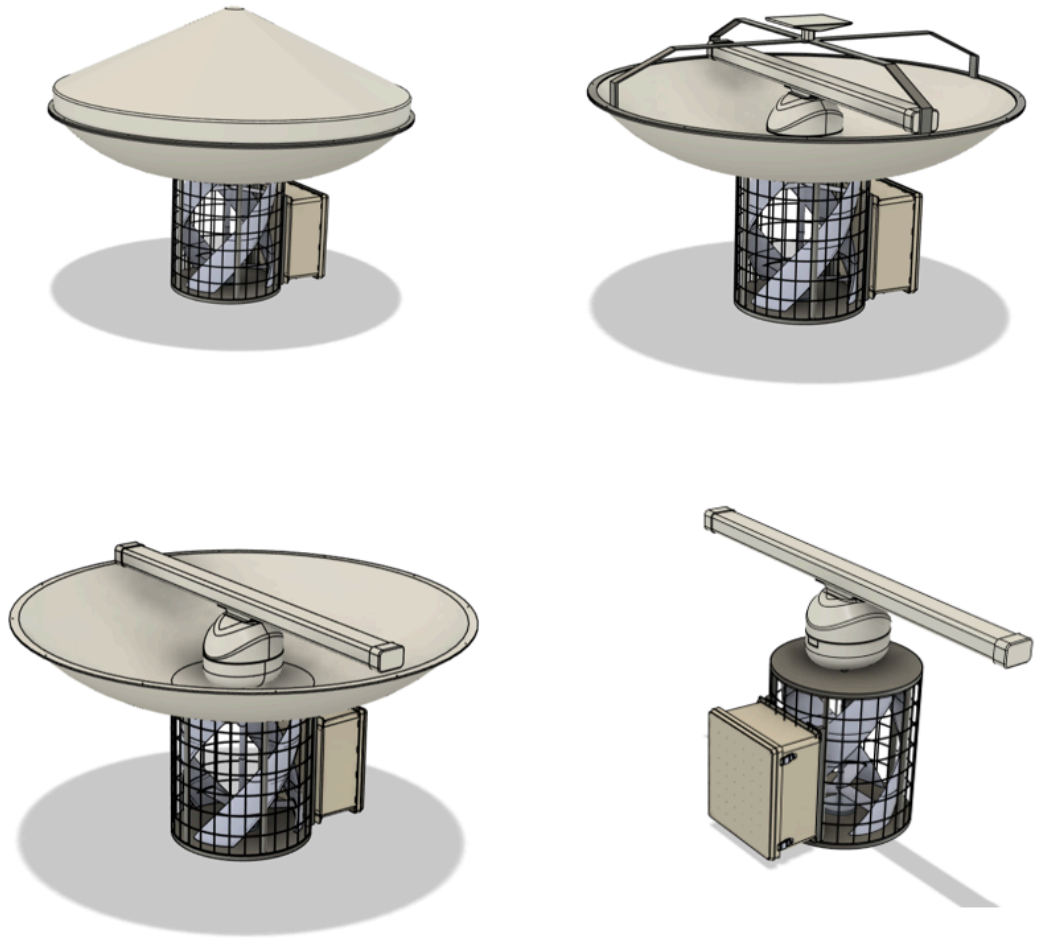


Figure 2.4 *- MDAS Prototype CAD [93]*

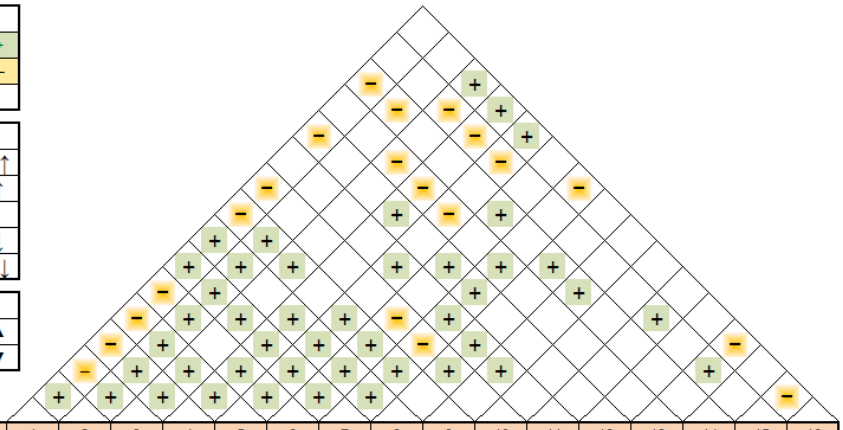
2.6 House of Quality

Table 2.3 *- House of Quality*

Correlations	
Positive	+
Negative	–
No Correlation	

Relationships	
Strongly Positive	↑↑
Positive	↑
Moderate	
Negative	↓
Strongly Negative	↓↓

Direction of Improvement	
Maximize	▲
Minimize	▼



Column #		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Marketing Requirements \ Engineering Requirements	Direction of Improvement	Weight	Size	Detection Range	Data Transmission Rate	Concurrent Targets Detected	Total Energy	Power Consumption	Maximum Voltage	Cost	Cold Start Time	ISO Range	Operating Temperatue	Response Time	Maintenance Interval	Algorithm Accuracy	Latency
Direction of Improvement		▼	▼	▲		▲		▼		▼	▼	▲		▼	▲	▲	▼
Portability	▲	↑↑	↑↑				↓		↓	↑					↓		
Ease of Use	▲					↑					↑	↑		↑↑		↑	↑↑
Affordability	▼	↓↓	↓	↓↓			↓↓	↓↓		↓↓	↑	↓↓	↓	↓			
Reliability	▲			↑		↑↑	↑			↑	↓	↑↑	↑↑	↓↓	↑	↑↑	↓
Energy Efficiency	▲				↑		↑↑	↓↓		↑							
Weather Resistance	▲									↑		↓			↑		
Safety	▲	↓↓	↓↓						↓				↓		↑		
Target Values		≤ 25kg	6.5" by 6.5" by 5"	50 to 60 nautical miles	4 to 10 times per second	40 to 60 Vessels	400 to 500 Watts	≤ 275 Watts	48 V	≤ 15,000	≤ 120 seconds	100 to 3200	5°C to 50°C	≤ 3 minutes	~6 months	≥ 99% accuracy	≤ 1 second

Chapter 3: Research and Part Selection

Developing our Maritime Domain Awareness System requires both deliberate planning and in-depth investigation. Before settling on any final design or starting the prototyping phase, we first laid out our specific requirements, surveyed potential solutions, and measured how each one would align with our overall project goals. This chapter looks at the step-by-step research that guided our technological choices. We divided this chapter down into key subcategories involved in the MDAS project. The discussion is broken into key subcategories of the MDAS project: radar systems, power and energy considerations, materials and structural design, electronics, and base station AIS-related information.

Throughout this process, cost, performance, and durability played significant roles in selecting power-management solutions and peripheral components. Our objective throughout this process was to find a balanced blend of reliability and adaptability which are essential for maritime surveillance. By explaining our decision-making criteria and how each piece fits into the larger system, we show how even the smallest parts contribute to maintaining consistent vessel tracking and reliable data transfer over long distances at sea.

3.1 Radar Systems

As part of our broader goal of maritime domain awareness, radar technology provides a straightforward and powerful method for detecting both cooperative and non-cooperative vessels. Many modern solutions rely on satellite tracking or the Automatic Identification System (AIS), but vessels looking to avoid detection often disable AIS or are not equipped with AIS transponders. In contrast, radar technology does not depend on the voluntary transmission of identifying data by a vessel. Instead, it relies on electromagnetic waves to find and follow targets based on reflected energy. This makes radar especially important for naval surveillance, where security resources might be thin at the coast or the area to be monitored is vast, like the open ocean.

3.1.1 Fundamentals of Radar Systems

Radar, or Radio Detection and Ranging, is one of the most important aspects of maritime surveillance because it has the capability to actively scan and probe the environment, unlike depending on cooperative emissions such as AIS signals. In practical terms, a radar system emits pulses of electromagnetic energy and calculates returning echoes to identify and follow targets in maritime scenarios, including vessels that may be operating “silently” without AIS transponders [6]. The two-way time between pulse transmission and echo reception identifies the range of the target, and signal-processing techniques can interpret speed and bearing.

Within the framework of this project, the Maritime Domain Awareness System (MDAS), having the capacity to detect both cooperative and non-cooperative vessels is essential. Solid-state pulsed radar technology, through the utilization of techniques such as pulse compression and Doppler filtering, provides improved clutter rejection and increased resolution without necessitating unrealistically high transmit power levels [6]. Solid-state

pulsed radar brings a key benefit to off-grid maritime systems: it significantly lowers the peak power needed for operation. This is especially valuable when the system draws from renewable energy sources, where power availability is often limited. Equally important, next-generation radar architectures commonly include integrated networking and remote-control interfaces, giving operators real-time command over parameters like gain, clutter filters, and range scales, even from a distance. These adaptive features are crucial in an off-grid, wide-area surveillance setting, since they allow the radar to respond to changing conditions while ensuring reliable performance under constrained power budgets.

3.1.1.1 Wavelength and Frequency Considerations

Operating frequency, which is also wavelength, significantly affects a radar's performance in maritime applications. Common marine radars fall into two primary bands: S-band (approximately 3 GHz, ~10 cm wavelength) and X-band (9–10 GHz, ~3 cm wavelength).

S-band radars, operating at a wavelength of approximately 10 centimeters, are particularly utilized for having a very fine capability to penetrate heavy rain, thick fog, and other forms of atmospheric obscurants which can affect the visibility. They often maintain detection ranges even during severe weather, although their longer wavelength typically results in less fine resolution and broader beamwidth. These characteristics make S-band radars primarily the radar technology of choice, for large-ship installations or where strong precipitation is the dominant factor [6].

X-band (~3 cm wavelength) radars provide excellent detail on small and closely spaced targets. This makes them particularly effective for detecting low-profile craft or distinguishing multiple vessels in congested waters. The inherent shorter wavelength of X-band radars allows for the creation of narrower beamwidths and narrower resolution cells, a characteristic which often results in higher accuracy in terms of monitoring on-the-move targets. However, it must be noted that a significant drawback comes into play, as the propagation of X-band is much more susceptible to attenuation when faced with heavy precipitation conditions, a factor which could then result in a reduction of the maximum distance at which these types of radars can effectively transmit.

In modern shipboard practice, bigger ships that travel transoceanic routes often carry both bands. S-band is carried for improved performance in heavy rain, and X-band is used for high-resolution scanning. For an MDAS intended to detect illegal "silent" ships and with a desire to minimize the hardware footprint, an X-band system offers a great compromise on size, power, and target discrimination.

3.1.1.2 Sea State, Clutter, and Environmental Effects

Beyond electromagnetic propagation, maritime radars must contend with the dynamic conditions of the sea surface. Sea clutter develops when radar pulses reflect off waves, whitecaps, and turbulent water, producing random echoes that can mask genuine targets

[12]. Higher sea states, especially those with breaking waves, have more severe clutter and demand sophisticated filtering techniques in order to be sensitive to small vessels.

- Clutter Suppression Algorithms:

  Doppler-based techniques, such as Moving Target Indication (MTI) and pulse-Doppler processing can be particularly effective in discriminating between moving targets and stationary or slowly moving reflectors that are typically found on the surface of the sea [6]. By isolating echoes with distinct Doppler shifts, the radar processor is able to emphasize the returns from vessels and, simultaneously, suppress the backscatter from the waves.

- Weather and Rain Clutter:

  Heavy rain works to introduce another source of clutter to the radar systems because the raindrops are capable of scattering the transmitted radar pulses. Even though it is true that S-band radars tend to have superior rain penetration because they operate at a lower frequency, the fact must not be lost that even X-band systems can reduce the impact of rainfall clutter. They do this by using shorter pulse widths, the use of pulse-compression waveforms, and by employing adjustable gain controls to enhance their performance during rainfall.

- System Settings for Changing Conditions:

  According to Furuno’s basic radar guidelines, operators must make frequent adjustments to gain, sea clutter filters, rain clutter filters, and range scale to maintain consistent situational awareness [11]. Automated or remotely configured filter profiles, which are particularly relevant for our untouched MDAS, enable the radar to adapt on the fly to rapidly changing sea states and is of great importance for optimizing maritime domain awareness.

Taking into account frequency band selection, wavelength-specific propagation characteristics, and clutter management makes a radar-equipped MDAS a powerful tool for detecting and tracking silent vessels. The integration of such complex signal processing, controlled transmit power, and adaptive clutter filtering ensures that a system can maintain reliable vigilance over maritime domains despite harsh sea states.

3.1.1.3 Radar Mounting Height

Aside from frequency selection and clutter mitigation, antenna height is a key factor in maritime radar performance. Mounting the radar antenna higher up on the ship's mast extends the line-of-sight horizon [11], enabling detection of low-flying ships at greater ranges. Ship roll and pitch, on the other hand, can momentarily tilt the radar beam, creating a "bobbing" blind zone during rough sea states. However, putting a radar up on the mast can bring a host of practical issues. Beyond the structural reinforcements needed to handle the added weight and wind loading, maintenance and vessel impact also become greater factors. Mast mounted setup becomes less desired and mounting to the pilot house is far more practical. Even a modest increase in elevation can result in a

significantly larger surveillance area, which is important when trying to spot smaller, low-profile vessels on the open sea.

- Extended Detection Range

  Line-of-Sight Geometry: The distance to the radio horizon generally increases with the square root of the antenna height. In practical terms, a radar mounted higher on a vessel's mast can see farther over the Earth's curvature, which means it can pick up targets that would be invisible to a deck-level antenna [6].

  Obstacle and Surface Clutter Clearance: Raising the radar's altitude helps reduce reflections due to nearby obstacles as well as mitigate the effect of sea-surface clutter produced by subsurface waves. As a result, this adjustment can produce more distinct target returns and reduce the processing load on the signal processor responsible for eliminating unwanted echoes.

- Impact on Small Target Detection:

  Low-Profile Boats: Small boats run lower in the water and often come in just above the top of the waves. By placing the radar at a sufficient height, operators can maximize their detection range for these vessels, thus increasing overall situational awareness [11].

  Wave Masking: In high sea states, a beam from an elevated position is less likely to be completely absorbed by breaking waves due to which targets are still detectable despite the occasional rise of wave crests between the radar and the sea surface.

- Manufacturer Recommendations and Practical Limits:

  Optimum Mounting Heights: Radar makers, such as Furuno, generally provide minimum and optimum mounting height recommendations. These considerations include extended range along with practical concerns like wind loading and stability [11]. Extremely high installations can compromise vessel balance and contribute to center-of-gravity problems, especially on small platforms.

  Antenna Size Implications: The larger the antenna array, the narrower the beamwidths at a specific height but with greater windage and weight. The site chosen for installation will need to be sufficiently robust to take this load and clear of obstructions from items like masts, funnels, or sensors.

- Compromises and Operations Modifications:

  Vessel Dynamics: Elevating the radar on a mast or the wheelhouse produces added motion due to roll and pitch, which can lead to transient blind sectors. Electronic or mechanical stabilization can minimize these [6].

  Maintenance and Accessibility: An elevated mounting location might introduce additional inconvenience when carrying out on-site maintenance. This is an

important factor as these will be going on commercial vessels and there will not be a crew on deck to correct anything that is wrong with the system.

In summary, the height of a maritime radar antenna is crucial for maximizing detection range and ensuring consistent performance. This is especially important for an autonomous system designed to track dark vessels. By paying close attention to the structural and operational implications of antenna elevation, along with considerations of frequency band, power settings, and clutter control, an MDAS can become more robust and capable of delivering wide-reaching maritime domain awareness [12]. As previously explained, this benefit comes with many hesitancies which leads to mounting on the pilot house which remains high enough while making installation and maintenance significantly more reasonable.

3.1.2 Types or Radar

In order to select the most suitable radar system for the Maritime Domain Awareness System (MDAS), it becomes highly necessary to possess thorough knowledge of all the various types of radar technologies currently in use, along with the varied physical form factors available in the maritime industry. Different radar systems yield variation not just in the way they generate and process signals, but also greatly in their physical composition, total power consumption requirements, detection range, and their durability under different environmental conditions [6]. In order to facilitate the design selection process for maritime surveillance mission sets such as MDAS, this section classifies radar technology into various operational types and structural form factors.

3.1.2.1 Comparison of Radar Technologies

Radar systems are broadly categorized by the specific internal operating mechanisms which they employ. There are several tradeoffs to be considered for each different radar system, including the power consumed, the detection fidelity, the total cost, and the maintenance demands, these latter ones being particularly critical to the functionality and performance of remote and autonomous systems.

Table 3.1 *- Radar Technology Comparison*

Radar Type	Description	Key Advantages	Limitations	Typical Cost
Magnetron Radar	Traditional radar type using vacuum tubes to emit high-powered microwave pulses	Long range, mature tech, inexpensive upfront	High power consumption, short life span, requires warm-up, poor low-speed perf.	Low
Solid-State Radar	Uses transistor-based circuits to generate pulses; often employs pulse compression	Low power, compact, long life, better with Doppler & signal processing	More expensive initially than magnetron; lower absolute peak power	Moderate to High
CHIRP Radar	“Compressed High-Intensity Radar Pulse” that sweeps through frequencies	High resolution, better clutter rejection, clearer images	Typically more complex to implement and integrate	High
Pulse-Doppler Radar	Measures target motion via Doppler shift; often integrated into solid-state sets	Excellent target separation from sea clutter; detects speed/direction	Computationally intensive, benefits from advanced processors	High
Dual Range Radar	Allows simultaneous display of short and long-range scans	Operator sees near and far targets without manual switching	Requires more powerful processors and slightly higher power draw	Moderate to High

These radar types are typically coupled, for example, solid-state radar with Doppler and CHIRP, to achieve better performance while keeping cost and power in check. For MDAS, the use of solid-state pulse-Doppler radar with CHIRP technology offers the most desirable balance between effectiveness, detection integrity, and environmental compatibility.

3.1.2.2 Radar Form Factor Comparison

Alongside signal technology, the physical design or form factor of a radar plays a vital role in its suitability for maritime deployment. Dome and open-array are the two dominant structural styles used in commercial marine radar systems.

Table 3.2 *- Radar Form Factor Comparison*

Form Factor	Description	Advantages	Limitations	Ideal Application
Dome Radar	Enclosed radome, often with shorter antenna arrays (12–24")	Enclosed radome, often with shorter antenna arrays (12–24")	Enclosed radome, often with shorter antenna arrays (12–24")	Enclosed radome, often with shorter antenna arrays (12–24")
Open-Array Radar	Exposed rotating array (3–6 ft), higher gain antennas	Superior range (up to 96 nm), narrow beamwidth, better separation of targets	Heavier, higher power use, larger footprint, requires robust mounting	Larger vessels, ocean crossings, wide-area surveillance
Hybrid Modular	Less abundant, customizable radar units with removable/expandable elements	Tailored size/ weight/ power tradeoffs	More expensive, complex integration	Military/commercial custom integrations

The MDAS project requires a radar system that is powerful enough to identify tiny vessels at a distance and small enough to be easily installed on a variety of commercial vessels. This critical balance is particularly well-served by open-array radars, particularly the 6-foot solid-state models. Especially when advanced technologies like Doppler processing and CHIRP are included into their architecture, they offer enhanced detection without requiring unduly high power needs.

3.1.3 Current Radars on the Market

In order to select a suitable radar for an overseas maritime surveillance project, we must consider performance in heavy weather, power demands for off-grid or wind-based power systems, integration flexibility for custom software, and of course, cost and range specifications. Below are five leading solid-state X-band radars commonly found on commercial vessels. Each system operates with pulse compression for improved detection and minimal maintenance, but they differ in how much power they require, how

readily they integrate with custom applications, and how extensively they support cost-effective deployments.

3.1.3.1 Furuno DRS6ANXT

Furuno's DRS6ANXT uses the company's "NXT" pulse-compression technology, designed to enhance target detection while containing overall power draw and weight. Its range performance varies depending on whether you opt for a smaller dome or a larger open-array antenna, with the dome version typically reaching out to around 36–48 nautical miles, while open-array setups can push farther, closer to 72 nm [7] In terms of cost, the DRS6ANXT can start around $3,000 for dome models and rise to over $8,000 for open-array configurations, making it a solid mid-tier option. Despite its moderate wattage draw (about 45 W nominal, peaking around 100–120 W in certain modes), it remains feasible for wind or solar power installations if properly sized. Furuno primarily uses NavNet TZtouch MFDs and can see packaged data via their SDK. Still, from a weather-resilience standpoint, Furuno radars are widely respected in commercial shipping and typically handle salt spray, heavy rains, and moderate gales with no issue.

3.1.3.2 Garmin GMR Fantom 126

Garmin's GMR Fantom 126 is well-known for its powerful doppler-based MotionScope technology, delivering excellent short- to medium-range target discrimination. With a six-foot open array, it can detect targets at distances up to around 96 nautical miles, which satisfies the requirements for broad-area maritime domain awareness. The purchase price often falls in the $7,000–$8,000 range, placing it in a higher bracket once you factor in the array hardware. In exchange, users gain sophisticated detection of high-speed or distant vessels–an advantage if wide coverage is a priority. However, the radar's power usage can climb to nearly 200 W under higher rotation speeds and transmit power, which poses challenges if your system must remain self-sustaining on wind power alone. Its open-array design also demands extra deck space and careful mounting at suitable height for best performance. Though Garmin provides user-friendly interfaces for end consumers, there is no official software development kit (SDK) for developers aiming to extract or process raw radar data. As a result, those wanting custom software integration face some hurdles.

3.1.3.3 Raymarine Cyclone Pro 6ft

The Raymarine Cyclone Pro is a top-tier open-array radar featuring a transmit power rating near 170 W. This higher power, along with its advanced pulse-compression signaling, enables reliable detection ranges up to around 72 nautical miles and more robust resolution in challenging weather. Cost-wise, the Cyclone Pro setup usually exceeds $6,000, sometimes reaching $8,000 or more depending on antenna length and accessories. Because it is a solid-state design, average power consumption remains lower than a traditional magnetron-based radar, but it can still climb above 150 W when scanning at higher speeds. That said, many overseas operators appreciate its heavy-duty construction, sleek footprint for minimal wind loading, and proven operation in high seas. Another critical benefit is the availability of Raymarine's LightHouse SDK, which allows direct software-level access to radar data and controls, a significant advantage in a project

requiring custom detection or remote management. For an energy-constrained MDAS platform, though, the Pro model's higher wattage output may strain a smaller wind-powered solution, making it better suited to scenarios with ample energy or bigger vessels.

3.1.3.4 Simrad HALO® 2006 Open Array

Simrad's HALO® 2006 radar delivers a capable 6-foot open-array solution for mid- to long-range detection, typically up to about 72 nautical miles. Its pulse compression and VelocityTrack doppler technology help operators distinguish actual moving vessels from clutter like waves or precipitation. Pricing is comparatively competitive in the open-array category, averaging in the $4,000–$5,000 range, though additional costs for accessories may apply. Power draw generally hovers around 65 to 130 W depending on rotation speed, which is relatively moderate for a large open array. This level of consumption is still a notable factor when using only wind power, but intermittent or scheduled scanning modes can help conserve energy. Simrad does not offer an official open SDK, so more advanced integrations often rely on partial open-source efforts or custom reverse engineering. Still, for standard "drop-in" commercial deployments, HALO radars have proven weather resilience and reliability at sea.

3.1.3.5 Raymarine Cyclone 6 ft

Raymarine's Cyclone (non-Pro) 6 ft open array offers a balance of performance and energy efficiency that makes it particularly attractive for the MDAS concept. Its typical output hovers closer to 110 W at peak, and many scanning modes operate below that figure, which significantly eases the burden of off-grid power generation using a wind turbine. The cost generally falls below $10,000, inclusive of the 6-foot antenna kit, placing it at the high end of dome radars but within reach for robust maritime security needs. It shares the same sealed, low-profile design as the Cyclone Pro, featuring an aerodynamic, weatherproof enclosure that withstands salt spray, heavy rains, and higher wind loads which is critical when mounting on external structures. Coupled with Raymarine's LightHouse SDK, integrators can directly manipulate scanning parameters, retrieve raw or semi-processed returns, and feed the data into custom tracking algorithms that filter out AIS targets. This level of software flexibility aligns perfectly with a self-contained, overseas-deployed MDAS platform that must autonomously detect silent vessels. Although its maximum range is slightly below that of some high-power open arrays, the Cyclone still comfortably meets the ~60–72 nm target range for mid-size maritime surveillance, making it an excellent tradeoff between coverage, cost, and operating power.

3.1.3.6 Comparison of Radar Models and Selection

Table 3.3 *- Radar Part Comparison*

	Furuno DRS6ANX T-XN13A	**Garmin GMR Fantom™ 126**	**Raymarine Cyclone Pro**	**Simrad HALO® 2006**	**Raymarine Cyclone**
Frequency Band	X-band (~9.4 GHz)	X-band (~9.4 GHz)	X-band (~9.4 GHz)	X-band (~9.4 GHz)	X-band (~9.4 GHz)
Form Factor	Solid-Sate, Open Array	Solid-State, Open Array	Solid -State, Open Array	Solid -State, Open Array	Solid -State, Open Array
Antenna length	6 ft	6 ft	6 ft	6 ft	6 ft
Output Power	25W	120W	110 W	50 W	55 W
Max Range	72 Nm	96 Nm	96 Nm	72 Nm	72 Nm
Supply Voltage	12, 24 VDC	10, 32 VDC	12, 24 VDC	12, 24 VDC	12, 24 VDC
Nominal Power (Standby / Typical)	45W / N/A	45W / N/A	35W / 95W	N/A / 50W	30W / 70W
Cost	$5,295.00	$8,999.00	$9,749.99	$7,499.00	$8,250.00
Rotation Speeds	24, 36, 48 rpm	24, 48 rpm	12, 24, 36, 48, 60 rpm	16 - 48 rpm	12, 24, 36, 48, 60 rpm
Horizontal Beamwidth	1.35°	1.25°	1.32°	1.2°	1.32°
Vertical Beamwidth	22°	23°	25°	25°	25°
Minimum Detection Range	6 m	6 m	6m	6 m	6 m
Weight	64.25 kg (41 lbs)	23kg (50.7 lbs)	19 kg (41.85 lbs)	31.6 kg (69.7 lbs)	28 kg (61.73 lbs)

Software Development Key	Yes	No	Yes	No	Yes
Additional Key features	-Pulse compression -Doppler target analyzer -Beam sharpening	-”MotionScope” Doppler tracking -Pulse compression	-CHIRP pulse compression -RangeFusion and beam sharpening -Advanced Doppler target tracking	-”VelocityRack” doppler tracking -Dual range mode -Beam sharpening	-CHIRP pulse compression -RangeFusion -Doppler target tracking

After evaluating multiple radar models, we chose Raymarine’s Cyclone 6 ft Open Array for its excellent range, performance, and integration capability in our Maritime Domain Awareness System (MDAS). Although Garmin’s GMR Fantom lacks an official SDK, disqualifying it for custom data extraction and control. Furuno’s DRS6ANXT does offer an SDK but it requires more extensive configuration and integration effort compared to Raymarine’s more accessible LightHouse Developers SDK, which is designed for smoother third-party application support. Simrad's HALO® 2006 also has minimal documentation, making necessary reliance on reverse-engineering for more extensive integration which lags from it being a COTS plug and play system.

In contrast, Raymarine provides an official SDK to enable third-party software to control radar functionality. Being able to do so is valuable for collecting data on vessels in foreign waters to subsequently transmit for AIS subtraction. The Cyclone's 60–110 W power consumption is considerably lower than the Cyclone Pro's 150+ W, a valuable benefit for off-grid installations run on renewables. Power to the system needs to be as little as possible and the improvements the Raymarine Cyclone Pro models offer are not achievable with self sustainment.

Using a 6 ft open-array antenna, the Cyclone can detect vessels at a theoretical range of 72 nautical miles, which is sufficient for monitoring small craft crossing transoceanic routes. Its aerodynamic design reduces wind resistance, thus easing stress on mounting points, while the corrosion-proof housing resists harsh maritime environments. All these features combined make Raymarine's Cyclone 6 ft Open Array the perfect radar choice for our MDAS, offering a balanced combination of wide coverage, simple integration, and efficient power consumption.

3.2 Maritime Environmental Challenges

The MDAS is set to face several challenges out at sea. Issues such as salt spray, high winds, vibrations, fluctuating temperatures and moisture were of high concern, which can degrade materials over time. The primary issue we discussed was corrosion in the materials going to be used to design the system’s structure. Consisting of two parts, the

radar system requires an open-air frame that houses the vertical wind turbine and a sealed enclosure to protect the system's electrical components. Both of these parts require corrosion resistant material to fight against common maritime issues like pitting and crevice corrosion where metal deteriorates at certain points causing weak spots in the design's structure. Selecting materials that can survive over maritime travel is crucial towards the MDAS' ability to maintain operation for an extended period of time. The next phase involved looking at the most popular materials to house a project like this and ensuring its durability given marine conditions.

3.2.1 Stainless Steel (316)

Stainless steel is well known for its strong mechanical properties and high resistance to corrosion in salt-heavy maritime environments. The most common grades are 304 and 316. Each type differs slightly in their elemental makeup which greatly impacts their performance.

Type 304 stainless steel is composed of majority iron, part chromium, and part nickel while type 316 is essentially the same aside from an additional bit of molybdenum to help it achieve a robust oxide film on its surface which improves resistance to chlorides and harsh chemicals. [77] As a result, it develops a great resistance to pitting and crevice corrosion. Type 304 is more commonly used in kitchen equipment and industrial cases, and we will shift our focus towards type 316 being the preferred choice in maritime applications.

316 Stainless steel's yield strength, which measures how much force can be exerted on a material before it permanently deforms, is 30 kilopounds per square inch (ksi). Its tensile strength, which measures how much force can be exerted on a material before it breaks, is 80-90 ksi. It has a density of 0.29 pounds per inch cubed. [84] These properties ensure that stainless steel can provide reliable structural performance under substantial loads.

Stainless steel offers long-term durability and low maintenance in specific maritime conditions. Its resistance to rust and corrosion allows structures to withstand the continuous exposure to saltwater, moisture, and varying temperatures at sea with minimal upkeep. Commonly used for designing boat hulls, piping and fittings, it fulfills its role for a while which can save cost in the long run. Furthermore, it also is very structurally stable with how much weight it can withstand and is well suited for environments experiencing constant vibration and heavy winds like on a boat.

However, despite these advantages, stainless steel has some major drawbacks. The first one is cost, when compared to other materials like some aluminum alloys and fiber reinforced plastics. Its complex composition and manufacturing process drives up the price of materials and definitely influences how we approach a material leaving other options to be considered. The second drawback is the weight of the material, some projects focus on specific tasks that require minimizing weight and stainless steel wouldn't be a great pick as it is quite heavy. Extra weight may cause components and machines to work harder leading to compromised efficiency in exchange for stability. Overall it is important as engineers to understand that although stainless steel is a viable option for its great durability and corrosion resistance, these factors must be balanced by

its drawbacks.

3.2.2 Aluminum Alloys (5083 & 6061)

After our consideration of stainless steel, Type 316, with its great corrosion resistance in seawater environments, the second material that we looked at was aluminum. Aluminum alloys are widely used for marine applications due to their greater strength-to-weight ratio, corrosion resistance, and versatility in both structural and enclosure designs. Given the weight limit and environmental requirements of the MDAS, we examined 2 different marine-grade aluminum alloys like types 5083 and 6061 to see if they would fit our needs.

Starting with type 5083, this alloy is widely used in marine systems where corrosion resistance is a top priority. It's made mostly of aluminum with small amounts of magnesium and chromium, allowing it to perform especially well in saltwater.[76] It's not as strong as some of the other aluminum alloys, but it also develops an oxide layer on the surface when exposed to air and water which holds up well against pitting and crevice corrosion. It achieves a yield strength of 31 ksi, a tensile strength of 44 ksi and has the density of 0.096lb/in^3. Type 5083 can retain good structural performance after welding which is useful when working with frames that require long sides which need to be created in sections. We can see the use of 5083 in many ship hulls, propellers, and pumps.

Type 6061 is another type of aluminum alloy examined and is mainly used in structural and architectural applications. While it is corrosion resistant, it doesn't perform as well as 5083 where the material is constantly exposed to saltwater. The addition of silicon in the elemental composition weakens its corrosion resistance but adds the benefits of higher overall strength and better machinability. It measures a yield strength of 40 ksi, a tensile strength of 45 ksi and density of 0.098 lb/in^3. It is often used in making items that are mass producible like internal automotive components, brackets or small frames.

In addition to the performance of the materials, aluminum is affordable and easier to machine. It is also much lighter than other metals which can reduce strain on the system but at the tradeoff of structural less strength.

3.2.3 Fiber Reinforced Polymers

After examining traditional marine-grade metals like stainless steel and aluminum, we explored Fiber Reinforced Polymers (FRPs) as a viable option. FRPs are different from metals as they are composite materials created by combining two materials together: a resin (which is a type of plastic) and reinforcing fibers (commonly glass or carbon). The resin is what holds everything together while the fibers are responsible for the material's strength and stiffness.

The key advantage of FRPs is that they don't behave like metals. Instead of relying on the raw strength of a single material, FRPs can be engineered by changing the way fibers are arranged. This allows parts to be strengthened in a particular way where it's needed most and fit in a multitude of projects. The flexibility that this material provides in its design is substantial and a valuable option worth considering.

3.2.3.1 Fiber Reinforced Plastics Study

To understand how FRPs perform under real marine situations, we analyzed a research study conducted in 2021 ‘Effect of Time-Real Marine Environment Exposure on the Mechanical Behavior of FRP Composites’. FRP samples were submerged in seawater for six to twelve months in real marine environments that had varying salinity, temperature ,exposure to algae and microorganisms.

Over time, the samples absorbed moisture and their surfaces showed signs of biological growth. This had a direct effect on the strength of the material. The scientists observed that voids formed inside the resin which resulted in many samples losing up to 40-50% of their original tensile strength depending on the type of resin and fiber configuration used. [82]

As a result of this experiment, it was concluded that Epoxy-based FRPs reinforced with E-glass performed better than polyester. Their structure remained more intact and the epoxy matrix was better at keeping water out. Therefore we can consider epoxy-glass FRPs a viable option for the MDAS system.

3.2.3.2 Epoxy-glass FRPs

Epoxy-glass composites are a quite popular FRP category. They are highly regarded for their mechanical stability and corrosion resistance especially in humid marine environments.

One of the most significant advantages of epoxy as a matrix is that it has very good adhesion to fibers, facilitating efficient load transfer from resin to reinforcement. This improves tensile strength and structural integrity over time. Additionally epoxy resins rarely shrink as they cure, reducing the risk of internal cracks and warping of parts that can hurt the structural integrity of the FRP. Results from the previous study in section 3.3.3.1 show that when compared to vinyl-ester and polyester, epoxy resins are more water resistant and can be more reliable in marine applications where material degradation from moisture is a concern.

The tensile strength of some Glass-Epoxy composites research ranged from 20,000 - 60,000 psi, converted to around 20 to 60 ksi. The yield strength or flexural strength found in the data sheet ranged from 40 - 80 ksi. This helps portray it as a strong option not just because of its resistance to corrosion but great structural strength comparable to its metal counterparts. From the design perspective, epoxy-glass FRPs have good stiffness-to-weight ratios and are relatively easy to mold. They appear to be a shining option as parts of the support legs or frame on the MDAS as it needs to be lightweight and very stable.

Nevertheless, epoxy-glass composites are not perfect. They perform well in withstanding mechanical loads but are also brittle and will crack upon impact or excessive flexing. In applications like on a boat where there is an abundance of vibration, it wouldn’t be very viable. Another disadvantage of epoxy resins is that UV exposure caused by the sun can slowly degrade the FRPs epoxy matrix unless specific coatings are applied. Because of

these negatives, epoxy resin may not be the best option and must require more research into FRPs.

3.2.3.3 Fiber Glass Reinforced Polyurethane

Another form of FRP materials that we also considered is fiberglass-reinforced polyurethane (FRPU). This composite uses a polyurethane resin, which is softer and more elastic than the stiffer epoxy or polyester resins. While this increased flexibility may at first seem to be a disadvantage for structural application, it can in fact be a tremendous benefit in dynamic applications where impact, vibration, and movement are constant conditions. As the project's structure will be exposed to dynamic winds, torque generated from the vertical turbine and wave-induced motion, having a material that can absorb stress without cracking is highly desirable. While the fiberglass is responsible for structural support, the polyurethane resin matrix helps it fight fatigue over time.

In real-world marine applications, FRPU composites have already proven to be a viable material solution. A good example of this is evident in Coosa's line of structural panels. Their Bluewater and Nautical series consist of polyurethane-infused panels with continuous strand fiberglass reinforcement. These panels are used across the marine industry in boats, yachts, and other seafaring vessels, particularly in areas that are exposed to water and mechanical stress over extended periods of time. Many use cases include bulkheads, which are internal walls within the hull that divide the boat into different compartments to contain water in case of floods. This material is also used as a core in the boat’s internals which increases its rigidity without adding much additional weight and is also reliable for reinforcing their hulls.

Weight also plays an important consideration. Coosa's Bluewater panels are available in three standard density choices: 20, 26, and 32 pounds per cubic foot, or approximately 0.011, 0.015, and 0.019 pounds per cubic inch. Coosa's Nautical panels are available in 15, 20, and 24 pounds per cubic foot density and are distinguished by being lightweight and economical, although they are generally more suitable for non-structural uses because of their lower fiberglass content.

We would look more into the Bluewater series due to its make up of a greater quantity of fiberglass reinforcement which adds stiffness and load-carrying capability. When compared to previous metal options, fiberglass seems a fraction of the weight.

When compared to traditional plywood, FRPU boards like Coosa’s are highly resistant to rot, mold and water absorption. Because of its closed-cell polyurethane matrix, even if the surface gets damaged or scratched, water would not effectively enter the material and weaken it over time. With a tensile strength ranging from 500 - 560 pounds per square inch. It is much weaker structurally than metals; however in the scope of this project, much of the system would be above water and not affected by much weight aside from mounting a radar on top of the systems frame and a turbine inside. Its strong durability,longevity and lightweight abilities portray it as a viable option.

It has limitations like the previous FRPs as prolonged exposure to ultraviolet (UV) radiation will still cause degradation of the polyurethane resin, resulting in discoloration

of the surface and eventual loss of integrity of the matrix. Thus, we must consider different applications of coatings to protect the material that is selected for the project. Additionally, its low stiffness makes it less suitable for components under heavy load without extra support.

In spite of these limitations, FRPU offers a great mix of advantages for marine operations. Its low-density, water resistance, and flexible nature allows it to be used for a variety of applications. If the correct supporting materials are used, Coosa FRPU boards might be a good choice for implementing onto the MDAS design.

3.2.4 Material Comparisons

Table 3.4 *- Material Comparison for MDAS Structural Components*

	316 Stainless Steel	**5083 Aluminum Alloy**	**6061 Aluminum Alloy**	**Epoxy Fiberglass Reinforced Polymer**	**Polyurethane Fiberglass Reinforced Polymer**
Density	0.29lb/in^3	0.096lb/in^3	0.098lb/in^3	0.065lb/in^3	0.011 - 0.019 lb/in^3
Yield Strength (kilopounds per square inch)	30 ksi	31 ksi	40 ksi	20-60 ksi	14-53 ksi
Tensile Strength (kilopounds per square inch)	70 ksi	46 ksi	45 ksi	40-80 ksi	~0.5ksi
Pitting / Crevice Corrosion Resistance	Excellent	Good	Fair	Excellent	Excellent
Maintenance	Low	Low	Low	Low (needs UV protection)	Low (needs UV protection)
Formability	Poor (hard to shape)	Good	Excellent	Good	Good
Cost	High	Medium	Medium	High	Medium

Following a thorough analysis of stainless steel, aluminum alloys, and fiber-reinforced

polymers, we needed to decide what the best options were in our design. Each material presented its own advantages and trade-offs across the multitude of metrics like weight, strength, corrosion resistance, formability, and overall cost. These factors greatly influence the development of the MDAS structure, especially when broken down into the main design components: the open air external frame, the electronics components enclosure, mounting hardware for radar on the frame and the mounting of the whole MDAS system to the naval vessel it would reside on.

3.2.4.1 Weight vs Strength

In mechanical strength testing, 316 stainless steel is the strongest with tensile and yield strengths of 70 ksi and 30 ksi, respectively. This is, however, traded off against a much denser 0.29 lb/in³, which disqualifies it from applications where weight is a factor. The 5083 and 6061 aluminum alloys, however, provide a high strength-to-weight ratio with densities between 0.096 and 0.098 lb/in³. Although not as high as steel, the aluminum alloys can be a possible option as structural materials. Epoxy-glass FRPs, having tensile strengths of 40–80 ksi and densities of approximately 0.065 lb/in³, are as strong as metal but much lighter. Polyurethane-glass FRPs like Coosa's panels are not very viable in supporting the whole system with the lowest tensile strength but instead are very corrosion-resistant and light which is ideal for lighter load-carrying tasks.

3.2.4.2 Corrosion Resistance

The most important factor to address in the marine environment is corrosion. Stainless steel 316 possesses excellent crevice and pitting corrosion resistance and thus is highly suitable for external frames that are exposed to seawater but the issue of weight disqualifies it from being applicable in much of this design. Aluminum alloys are comparatively good, with 5083 being the one that forms a similar protective oxide layer. However, it is outshined by the Fiberglass Reinforced Polymers, both the epoxy and polyurethane reinforced variants. Because they are non-metallic and chemically stable in nature in marine environments, this makes them ideal use cases for those components which are in direct contact with water or those components that must be sealed for extended periods.

3.2.4.3 Maintenance and Durability

Stainless steel and aluminum metals are low maintenance but stainless steel has a longer lifespan and better resistance to mechanical deterioration. Epoxy and polyurethane composites also have low maintenance requirements but their lifespan is highly dependent on UV protection as their resin matrices can deteriorate overtime under constant exposure to sunlight without protective coatings. They would require an outer protective layer like an epoxy or paint for long-term deployment with minimal or no maintenance.

3.2.4.4 Formability and Cost

By considering various manufacturing processes, 6061 aluminum stood out with its ease of machinability and convenience in shaping. On the other hand, while stainless steel is

highly resilient and resistant, it turns out to be very problematic in the fabrication process and is very difficult to work with, thereby incurring high costs associated with its production. Composites turned out easier to mold and customize for specific shapes, especially for components that require high water resistance or enclosed configurations. Regarding cost, stainless was the most expensive material. both in terms of raw materials and fabrication processes. Aluminum is in the middle of the road with medium cost and acceptable performance. Composite materials like epoxy FRPs, are more costly initially but can be cost-saving in terms of weight and corrosion maintenance. Finally, FRPU composites, such as Coosa boards, are competitively priced and find a balance between performance and price. They are also easily customizable for a multitude of tasks.

3.2.5 Material Selection for MDAS Components

One of the main key aspects of the MDAS system is its self enclosed vertical design with every component having a lightweight and compact design. All of this has to be structurally stable and mounted on the respective vessels where they will be tested. As a result, every element has to be particularly chosen with a specific purpose in mind and work together to be successful.

3.2.5.1 Main Outer Frame

In order to fulfill the critical role as the primary external structure that would effectively sustain both the radar and turbine and ensure system integrity overall, epoxy fiberglass reinforced polymer (Epoxy-GFRP) was selected. This material possesses a very high tensile strength of 40 to 80 ksi combined with a very low density of approximately 0.065 lb/in^3. Also it is a materially strong, lightweight material, which is particularly helpful in avoiding top-heaviness and improving handling when facing unexpected sudden movements on water. Additionally, its excellent resistance to corrosion guarantees that it can withstand continuous exposure to the marine environment without serious degradation.

3.2.5.2 Electronics Enclosure

The enclosure for the electrical components is located in the bottom section of the frame and should be designed ideally with polyurethane fiberglass reinforced polymer. We were specifically recommended for the Coosa's Bluewater panel to be utilized in designing parts of the structure and it seemed like a good fit. As the enclosure does not experience severe mechanical stress, it is important that it has a proper seal to protect the internals from water and other marine environmental threats. FRPU's closed cell composition and high water resistance properties make it ideal. Although it is quite weak in the tensile strength department (~0.5 ksi), its environmental resistance, reliability and machinability compensate for its weakness. Also its low weight greatly benefits the system as it reduces unwanted stress on the bottom frame.

3.2.5.3 Turbine Mounting Bracket Support

For the mounting bracket that supports the turbine, it was determined that Coosa's Bluewater panel would be utilized again. Our decision was made not based on its water

resistant capabilities but rather how we deemed the specific mechanical factors that might negatively impact the enclosed turbine. This component is subjected to cyclic loads and repeated vibrations generated from the torque of its vertical nature. Unlike rigid materials which have the tendency to crack or deform when exposed to prolonged periods of oscillatory stress, the FRPU possesses flexible characteristics. These characteristics help absorb energy and redistribute stress throughout the material rather than a single point. The polyurethane matrix in this composite plays an important role as a dampening factor to reduce fatigue-caused damage over time. The strength of this material is determined to be strong enough to support the vertical wind turbine chosen. Not to mention again its light-weight capabilities, we deemed it a very useful part of our systems design where routine maintenance may be difficult and infrequent.

3.2.5.4 Base Plate

The base plate is the most important part of the whole design. As the anchor that attaches the MDAS to the ship's deck, dockside platform or other maritime structures, it needs to be made from the strongest and most durable material that was researched. Based on the research findings, 316 stainless steel appeared to be the most compatible based on the requirements we decided. Its appealing tensile strength reaching up to 70 ksi, ensures the reliability for withstanding not only the static load of the system but also the dynamic loads from the wind turbine, gusts of high winds, ship movements and sea-state vibrations. Not to mention it was the densest option researched when compared to our other corrosion resistant materials.The deployment surface is key in determining the success of the project as one major collapse can render the system useless.

Furthermore, coupled with its excellent resistance to pitting and crevice corrosion, a 316 stainless steel base plate can withstand coming in contact with salt spray and trapped moisture. Its ability to resist degradation over time can be great as long as it keeps its outer protective oxide layer from the added molybdenum in its chemical makeup. Although from the looks of it, one can see that having metal on a system that would be in maritime environments for a long while is a cause for disaster. However, if it is properly prepared with protective coatings, the drawbacks can be negated and its advantages can be utilized.

3.2.6 Protective Coatings

Even with the careful selection of corrosion-resistant materials, long-term exposure to the harsh maritime environment will have a significant and negative impact on the structural integrity of the components over time. Saltwater, UV radiation, and biological fouling are major factors that contribute to surface damage and performance loss. In order to enhance the durability and reliability of a material over time, there must be coatings applied, which serve to add additional layers of protection for both the metal and composite materials utilized in the design and construction of the MDAS.

Protective coatings have two important functions: corrosion protection and fouling resistance. Anti corrosive coatings work by establishing a physical barrier to project materials against moisture, oxygen and salt ions. Antifouling coatings are designed to stop the buildup of marine organisms such as algae, barnacles and bacteria that stick to

underwater surfaces. Without the protective layers, these factors can hurt the performance of structures that utilize compromisable material.

3.2.6.1 Epoxy Coatings

Epoxy coatings are primarily used as barrier coatings to protect materials against the environment. As explained previously, an epoxy is the formation of a tightly connected polymer that helps prevent moisture, oxygen, salts and other corrosive elements from entering a material. This is especially prominent in marine environments where this protection is crucial.

The most significant advantage of epoxy coatings is their excellent adhesion to metal and composite surfaces. This is really important as a variety of materials can go into designing a structure and having the ability to choose the best fit for the situation at hand. The flexibility allows for other design aspects to be considered before applying a protective coating.

Another advantage would be its ability to add tensile strength and integrity to the applied material. In situations of high vibrations and winds which can cause mechanical stressing, applying an extra coating can combat these factors. A study confirmed that "high-performance epoxy coatings exhibit low permeability and high durability when applied at adequate thickness and with other surface preparation".[59] Meaning that the coating's characteristics reduce the chances of microcracks forming, increasing a material's water resistance.

Moreover, epoxy coatings can also be utilized in multi-layered protective layer coatings. It can be paired with a polyurethane coating, which is going to be researched in the following section, and improve a material's UV resistance alongside providing great resistance to corrosion. This can be a very viable option to apply to metal materials discussed previously.

3.2.6.2 Polyurethane Coatings

Polyurethane coatings are another popular type of protective finish for materials in marine environments. Typically applied as topcoats over epoxy primers. The difference is that epoxy coatings help establish a layer of corrosion protection and polyurethane provides UV protection, mechanical flexibility and extra durability.

Its UV protection is a key point as polyurethane has exceptional durability when subjected to sunlight. Epoxy coating, despite being chemically and mechanically superior, undergoes degradation in UV environments. It results in the surface breaking down which releases fine powder known as chalking and begins discoloration over time. On the other hand, polyurethane excels at UV stability, maintaining its color and protection.

Polyurethane coatings also allow materials to be more abrasion and impact resistant, qualities which are really appealing when deciding on finishes for components subjected to constant motion from winds and waves.

In real-world marine systems, polyurethane topcoats have seen widespread use on offshore platforms, ship hulls, coastal facilities and many more. Because of their low maintenance costs and longer periods between services which is an ideal condition for the MDAS that would probably be installed in remote areas and rarely serviced. Properly applied over a strong epoxy primer, polyurethane coatings can greatly extend the life and functionality of system elements that are exposed to the severe conditions of the ocean environment.

3.2.6.3 Anti-fouling Coatings

Although epoxy and polyurethane coatings cover the corrosion and ultraviolet degradation factors that might impact maritime applications, they do not protect against the biological threats that come with prolonged exposure to seawater. Marine structures are very vulnerable to biofouling, which is the buildup of unwanted biological material, including algae, barnacles, and microbial organisms on their exteriors. This can cause material deterioration and decrease overall system performance. To combat this, an extra protection layer like other coatings is typically required.

Antifouling coatings are necessary to provide long-term efficiency and integrity of marine systems. There are two main choices of antifouling coatings and they include: biocide-based coatings (BCs) and foul-release coatings (FRCs). Biocide coatings operate by releasing active chemicals agents like copper or zinc compounds, it can help drive away marine growth by creating a toxic environment on the materials surface. Although it can be very effective, this approach raises environmental concerns and loses its effectiveness over time as the coating slowly deteriorates to release its chemical ingredients.

Foul-release coatings on the other hand can provide a non-toxic solution. It is commonly made from silicone or a fluoropolymer-base. Instead of using chemical deterrents, these coatings create a surface layer with very low adhesion possibilities. Together with the movement through water, any adhered material would gradually lose its grip and brush away. This was supported by a 2024 analysis that claimed foul-release coatings performed better in the long term in minimizing biomass and organic materials, especially in areas of heavy marine growth. Their drag and foul reduction applications make it extremely useful in maritime applications.

3.2.6.4 Protective Coating Material Comparisons

Table 3.5 *- Protective Coating Comparisons*

	Epoxy	**Polyurethane**	**Anti-fouling**
Primary Function	Barrier against moisture, oxygen and salts	UV resistance, abrasion and impact resistant	Prevents marine growth on structural components

Application Layer	Primer/base	Topcoat over epoxy	Topcoat
Advantages	Excellent adhesion and mechanical strength Water resistant	Durable under motion and weathering	Prevents biofouling Prevent material decomposition
Limitations	UV degradation without topcoat	UV stability depends on material quality. Expensive equals better	Needs reapplications
Cost	Moderate	High	Moderate/High

Given that the MDAS would be mounted to commercial ships sailing through marine environments, the use of protective coatings are essential in providing important layers of defense from extreme environmental threats. Though biofouling can be a concern, the MDAS is more of a surface-mounted surveillance system. Mainly placed on elevated positions where it can utilize its radar to the highest potential, it becomes more exposed to sunlight, salt spray and weather. Biofouling is more common in areas where components are constantly submerged underwater where algae and microorganisms are more of a threat to the object's exterior.

To address these issues, the design would prioritize the long-lasting corrosion and ultraviolet degradation rather than antifouling. Our coating design would most likely rely on the both epoxy and polyurethane working together as they have a great synergy when used in a multi-layer system. Epoxy is used as the base layer where needed, providing enhanced adhesion, mechanical strength, and corrosion resistance. Polyurethane is used as the topcoat, offering high UV protection, color retention, and weathering resistance. Antifouling coatings may be used sparingly to combat areas where water is prone to collect or where many crevices are found after designing.

Each structural element of the MDAS has to be customized with a set of coatings based on its material composition, mechanical function and environmental exposure. The primary outer framework, which is composed of epoxy fiberglass reinforced polymer (Epoxy-GFRP), offers great strength-to-weight properties and has natural corrosion resistance. The downside is that its epoxy matrix is prone to ultraviolet degradation over time. That being said, although it doesn’t require a base coat, a topcoat in the form of UV-resistant polyurethane should be applied to all exterior surfaces to shield from prolonged exposure to sunlight as well as surface abrasion.

The electronics enclosure, situated at the bottom of the structure, is constructed from polyurethane fiberglass reinforced polymer (FRPU), specifically Coosa's Bluewater panel. Although the material provides improved water resistance and reduced weight, it does not have sufficient tensile strength and is prone to ultraviolet (UV) radiation. An epoxy primer should be applied first to close the surface and ensure adhesion then topped with a polyurethane topcoat that will prevent UV and environmental degradation to the composite.

The turbine mounting bracket, also designed from FRPU, supports the vertical wind turbine. The component is subjected to continuous torsional and vibrational stress. While the flexible characteristic of the composite diminishes fatigue, it still benefits from protective coatings. The material will be strengthened with a surface-sealing epoxy base coat, and a polyurethane topcoat will protect it from weathering and UV degradation.

The most structurally significant component is the base plate, which is suggested to be constructed out of 316 stainless steel. Despite the corrosion resistance of stainless steel, protective coatings will enhance its life and reliability even more. A high-build epoxy primer will be used to create a strong barrier to moisture and salt, followed by the application of a polyurethane topcoat to provide resistance to ultraviolet radiation. Possibly in areas where water can be trapped, specifically at bolt holes, crevices or brackets, antifouling coatings can be applied if needed.

Overall, by examining each material's strengths and weaknesses and analyzing which structural components they would be applied to, this coating process helps in properly ensuring some of the most critical performance goals of the MDAS system: long-term durability, structural integrity, and minimal maintenance in harsh marine environments.

3.2.7 Mounting Hardware

Initially when we were presented with this project, our team thought the MDAS was going to be mounted on top of the mast of a naval vessel. However after questioning existing captains of research vessels and receiving advice from our sponsor, it was clear that it was more appropriate on top of the captain's boathouse. This led to two different renditions of the MDAS structural design shown in Figure 2.5.

3.2.7.1 Flanged Mount

A flanged mount is a form of structural support that employs a flat plate, most commonly either square or circular, along with pre-drilled holes to offer stability to the equipment that gets fixed on top. This mounting method is commonly used in engineering applications where stability, simplicity of installation, and durability against environmental stresses are of the most important. The flange plays a similar role to the base plate of the MDAS as it is another point of structural stability. It offers several points of attachment, thus enabling weight and mechanical loads to be distributed over a large surface area, reducing the risk of localized stress that could damage the mounting surface. In areas where structures are exposed to wind, vibration and other environmental factors, flanged mounts present a strong and stable solution to preparing a piece of equipment to be mounted.

They are also particularly useful in applications where the mounted equipment experiences motion or generates torque as in the case of systems with rotating components like radar arrays or wind turbines. When bolted, the flange connection can successfully resist displacement and are typically designed out of corrosion-resistant materials like stainless steel or anodized aluminum.

Within the context of the MDAS, the flanged mount is a great option if the system is meant to be placed on the roof of a marine boathouse. Through the use of a standard bolt pattern, the system can readily be installed on different commercial or naval platforms without the need for extensive modifications. This aligns with the project's goal of a scalable, plug-and-play marine surveillance system that can be relatively deployed.

3.2.7.2 Cantilever Mount

The flanged mount provides a solid method of attaching the MDAS system to flat surfaces, including rooftops or reinforced decks. However, there are certain deployment situations in which flat mounting surfaces are not available or are impractical. In these cases, some equipment must be attached to poles, masts or vertical structural supports to achieve stability. This is where the cantilever mount reveals itself to be a viable option, it offers both structural integrity but also spatial flexibility to install items with the stated constraints.

A cantilever mount creates a bracket-type support system extending outward horizontally from a vertically placed surface, strengthened by an angle brace diagonal for supporting off-center loads. This design offers the MDAS unit with its radar array, wind turbine and optical components to be positioned clear of potential obstructions. The radar can now with a greater unrestricted line of sight, can provide more accurate readings for our system. The wind turbine has access to uninterrupted airflow when up high. Therefore the system has improved visibility, performance, and power generation efficiency.

The cantilever mount also improves the modularity and scalable aspects of the MDAS system as it allows for a greater range of surfaces and orientations, increasing the range of possible deployable locations without compromising system stability. Typically the mounts are machined out of marine-grade aluminum or coated steel to prevent corrosion and stress. Which is required in these marine environments.

3.2.7.3 Deck-penetrating Mount (Through-bolt Mount)

A deck-penetrating mount, or through-bolt mount as more commonly referred to, is a robust and durable mounting technique used in naval applications where structural strength is the top priority. This type of mount involves securing the equipment by fastening bolts directly through the mounting surface, most often use cases revolve around ship decks, platforms or reinforced docks. Through this process, it allows loads to be distributed throughout the entire thickness of the structure instead of being limited to the surface. The bolts are secured from below using lock nuts and washers creating a mechanically solid connection that will resist shear, uplift and twisting motion. In marine environments where equipment can be exposed to high winds, wave impacts or vibrations, this method can effectively provide a foundational support. The

deck-penetrating mount provides a higher level of mechanical security for the mounted equipment compared to the other options. It is intended for mounting equipment intended for extended high-performance operation over a long period of time.

3.2.7.4 Mounting Hardware Comparisons

Table 3.6 *- Mounting Hardware Comparisons*

	Flanged	**Cantilever**	**Deck-penetrating**
Portability	Semi-permanent - can be removed with tools	Semi-permanent - can be repositioned with effort	Permanent
Installation Complexity	Low - Requires access to top surface only	Moderate - Requires bracing and precise alignment	High - Requires access to underside and sealing
Vibration Resistance	Moderate	Moderate	High
Surface	Flat, reinforced rooftops or horizontal platforms	Vertical posts, masts, or side structures	Reinforced deck, dock, or platforms with underside access
Advantages	Simple to install, evenly distributes load	Maximizes clearance Good for radar visibility and wind turbine airflow	Strongest mount; ideal for high wind/wave environments Very secure
Limitations	Requires flat surfaces May transfer vibrations into surface	Higher stress on joints Requires rigid support structure	Difficult to relocate or service
Cost	Moderate	Moderate - High	Low - Moderate

After careful evaluation of the mounting options: flanged, cantilever, and deck-penetrating, along with the functional and environmental requirements specific to

the MDAS system, the flanged mount was decided to be the most suitable option for deployment. This decision was made from a combination of technical, practical, and operational considerations that follow the project's core goals of being lightweight, portable, and self contained. The system must also be self-sufficient and not rely on the vessel's power.

The main advantage of the flanged mount is its simplicity in design. As a simple plate of metal, it is suitable for mounting on flat surfaces which are often the roofs of naval boat houses or observation platforms. With already pre drilled holes, it allows for quick mounting using standard fasteners and requires access only to the top surfaces of areas. This greatly reduces installation time and complexity which is especially worthwhile in areas where knowledgeable people or complex tools may be needed.

Moreover, the flanged mount distributes the structural load of the equipment over a greater area which helps in decreasing stress on specific areas and provides the system another stable point of support during high winds and torque caused from the enclosed wind turbine.

When comparing the flanged mount with the other options, the deck-penetrating mount provides better structural support but it requires access to the underside of the mounting surface. This entails a lot of manual labor in its installation and makes it less ideal for simple or immediate use. Similarly when compared to the cantilever mount, which is suited for mounting on vertical poles or masts to offer more clearance, it too requires more work and setup. Specific mechanical stress points for attachment must be calculated and designed for the MDAS unit thereby complicating the whole process.

Finally, the flanged mount supports the modularity and plug-and-play capability demanded by the project's scope. It employs a standard bolt pattern which simplifies mounting the system on most platforms without the requirement for a custom adapter or part. Such flexibility is essential for a scalable solution that can be used across different maritime applications, from coastal structures and harbors to fixed on maritime vessels. Through these decisions, the flanged mount enables the MDAS unit to be easily deployable, have structural stability and be durable over some period of time out in maritime conditions with minimal maintenance.

3.3 Optical Technology Substitution Feasibility

For this project, the primary objective is to utilize radar technology to carry out our main goal of expanding maritime domain awareness. Overall, our team chose radar because of its proven reliability in harsh conditions. However, we also explored the option of using other optical technologies to accomplish the same goal.

3.3.1 LiDAR Imaging System

One of these options would be using a Light Detection and Ranging system or LiDAR system. In the case of our project, it is not feasible to replicate a system in the allotted time and budget, but the following details the ideal solution and its working principles (59). The LiDAR system has a wide variety of applications in different fields, including

scientific, industrial, and militaristic. The main principle of LiDAR is that it uses laser pulses fired into the atmosphere in which radiation is continuously scattered back toward the laser and then collected and measured by a telescope with a photomultiplier (57). This information is then processed and provides the distance traveled by the laser pulse. Laser pulses are optimal for this process because of their narrow bandwidth, which makes it possible to measure the concentration of molecules along the path. The main advantage of using LiDAR is that it can provide precise positioning over large areas, as well as the quick transmission timing of the laser pulses (60). The detail and speed that are accomplished using this system are almost otherwise unmatched. LiDAR works using a principle called the lidar equation, or $P = \frac{E_0 TAc\beta(r)}{2r^2} exp[-2\int_0^r a(r')dr']$ which is derived from the energy, dE, received at the telescope after the backscatter is given by $dE = E_0 TA\frac{\beta(r)dr}{r^2} exp[-2\int_0^r a(r')dr']$. In these equations, Eo is the initial laser pulse energy, T is related to the optical efficiency of the receiver, A is the telescope area, $\beta(r)$ is the volume backscatter coefficient of the atmosphere at range r, and $a(r')$ is the volume extinction coefficient at range r' (57). For the basic working principles, two measurements need to be taken into account, the position of the system and the distance of the object. There are two main types of LiDAR imaging: airborne and terrestrial. For airborne LiDAR the system is normally installed at the bottom of the aircraft looking downwards. Within this category, there are two subtypes: topographic and bathymetric. Topographic systems are mainly used for landscapes, while bathymetric systems are used for measuring elevation and depth of water. Terrestrial LiDAR systems can either be one- or two-dimensional and scan in either 180 or 360 degrees. There are two subtypes for topographic modeling as well, static and mobile modeling, which relates to whether the scanning system is mounted on a stationary tripod or a moving vehicle. For the case of our project, we would most likely be looking at a mobile terrestrial system that would be mounted on the mast of the ship. With this system there are some drawbacks, especially with the conditions we require it to function under. With the conditions of the open ocean, an incredibly precise system is required in order to stabilize the laser pulse. There are also issues of impeded operation due to precipitation in the atmosphere, given that the added water particles increase scattering and absorption and degrade the accuracy of receptors. Especially for using the Starlink satellite system as specified in the constraints, there is an incredibly small target that needs to be hit, which requires a multitude of corrective measures in both software and hardware. For a LiDAR detection system, there are some advantages over a radar-based or visual-based detection method. When looking at the current market options, there is usually a camera system for distances less than a kilometer, or radar for distances greater than a kilometer. The camera-based method can be quite capable in the near field but suffers from light distortion from the surface of the water. There is also the fact that real-world positioning cannot be provided without a third-party system running code and doing data analysis. Overall, research relating to LiDAR systems on maritime vessels is quite limited at the current moment, but it has been shown that object detection for both camera and LiDAR systems is heavily influenced by the motion of the vessel. This increased and erratic motion leads to misfires and discontinuities from the detection system (59). Another disadvantage of LiDAR

imaging systems is that there is a higher cost of equipment and maintenance since the parts are more complex, which is something we are trying to avoid in our project.

3.3.1.1 LiDAR Imaging vs. Radar Imaging

The basic purpose behind LiDAR and Radar are the same, detecting the presence and volume of distant objects (61). However, the process behind how they each work is very different. While LiDAR uses infrared laser beams, radar uses radio waves to generate an image. Using the infrared laser, a 360-degree scan is made and then a point cloud of the surrounding environment is created. This point cloud is then manipulated by software to create an accurate 3D map of the environment with high resolution. Since radar is less precise, it is more useful for larger object detection (i.e. ships and aircraft).

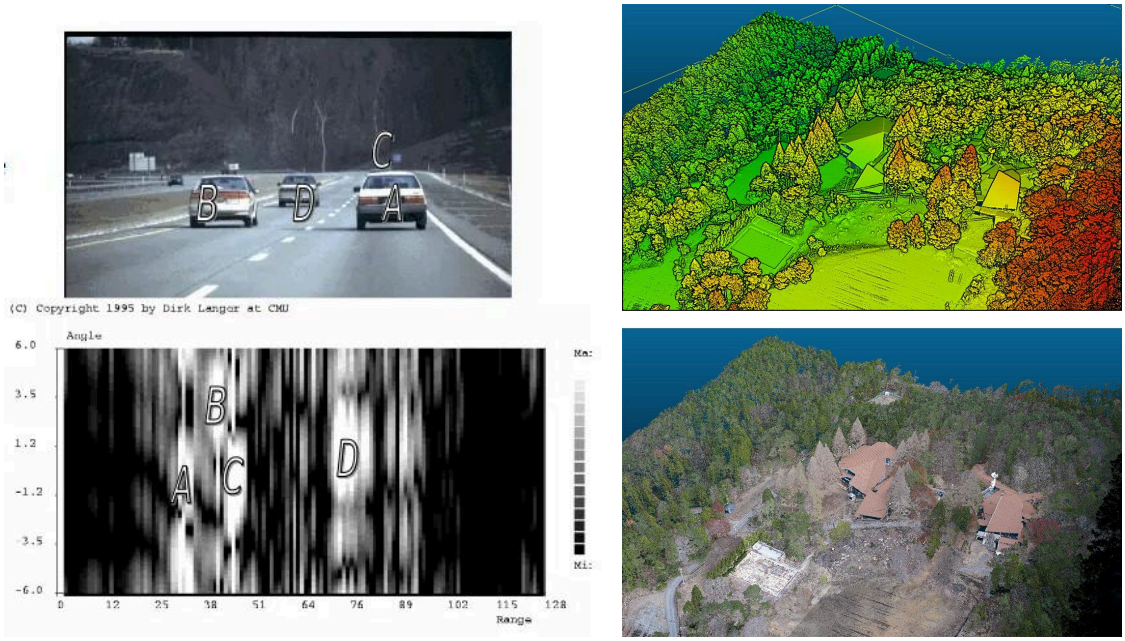


Figure 3.1 *- Radar vs LiDAR data comparison (61)*

While LiDAR may be more precise and have higher resolution, radar has the great advantage of being able to operate in adverse weather conditions, which is imperative for our project. In order for our system to work properly, the signal must be able to transmit through intense waves, salt spray, thunderstorms, and high wind conditions.

Table 3.7 *- Radar vs LiDAR comparison*

Category	**LiDAR System**	**Radar System**
Operating Principle	Uses short laser pulses to measure distances using time-of-flight calculations	Uses radio waves to detect an object's range, speed, and angle

Operating Wavelength	Typically 905-1550 nm (near infrared)	Radio Waves: HF (0.003 - 0.03 GHz) VHF (0.03 - 0.3 GHz) UHF (0.3 - 1 GHz) L (1 - 2 GHz)
Resolution	High spatial resolution; produces detailed 2D and 3D maps	Lower spatial resolution; more adept at detecting larger objects
Active or Passive Sensor?	Active	Active
Accuracy	Very accurate distance measurements (within 1-3 cm)	Accurate, but less precise measurements (10-30 cm)
Effective Range	Short to medium range (up to approx. 250 m)	Long range (500+ m); size of antennae limits lateral resolution
Weather Sensitivity	Affected by any precipitation, as well as dust	Still performs well in deteriorating weather conditions
Light Sensitivity	Direct sunlight can affect performance; reflections from the surface of bodies of water can impede imaging	Unaffected by lighting conditions
Object Detection	Good for detecting smaller shapes and textures	Good for larger objects and motion tracking
Data Output Formatting	High-density point cloud	Lower resolution but includes movement data
Processing Requirements	Higher due to complex data	Lower since data isn't as complex
Cost Analysis	More expensive to build and maintain	More affordable and developed technologies are available
Size and Form Factor	Larger system, specifically with rotating types	Smaller systems are available

LiDAR vs. Radar Comparison Chart

3.3.2 Camera System

Another optical technology that was considered was the use of cameras to create the tracking overlay. Cameras make use of a passive sensor to create images. Passive sensors detect the natural radiation (in this case sunlight) from the environment to produce the output (62). The nature of the passive sensor prevents it from functioning perfectly in both direct sunlight and total darkness and therefore requires extra lighting or add-ons to correctly process images. Cameras only capture images in two dimensions, meaning there is no depth to the produced images. This lack of depth perception can be a bit challenging when analyzing images, as it can often identify false positives. The two main kinds of image sensors that cameras use are Complimentary Metal Oxide Semiconductor (CMOS) and Charge-Coupled Device (CCD) sensors. The sensor itself is a semiconductor that converts optical images into digital signals for further processing (63). The imaging process itself begins when the input light passes through the lens and hits the sensor. In order to convert the light into an electrical signal, the photoelectric effect is used for the conversion, which is proportional to the intensity of the incoming light source. The electrical signals from all the photosites in the sensor are passed to the camera's image processor, which interprets all the information to determine each pixel's individual color and brightness (64).

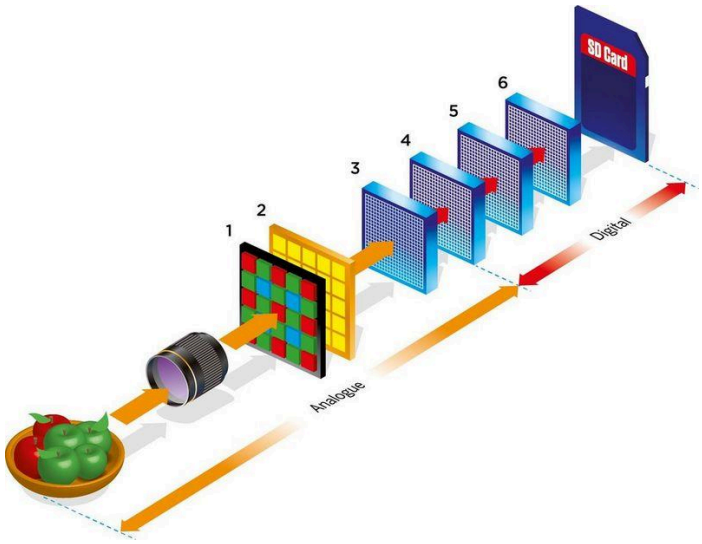


Figure 3.2 *- How cameras create a digital image (64)*

While a sensor's megapixel count is important for image resolution, the physical size of the sensor is also important. For example, if two sensors have the same total pixel count but one is physically larger than the other, then each photosite on the larger one must be bigger. While cameras are great for creating clear visuals of the surrounding environment, for our project's distance requirements radar is the clear choice moving forward. However, we will still be integrating a camera system into our project. This imaging system will be used to take pictures of identifying features of ships encountered offshore for surveillance purposes.

3.3.2.1 CCD Camera Sensors

The CCD sensor was the first advancement that allowed images to be captured without using film, which revolutionized photography at the time. The CCD sensor is comprised of a silicone surface on which an integrated grid of semiconductor capacitors that are capable of holding an electric charge are etched (64). The capacitors act as photosites, with the charge being directly proportional to the light that hits it. The photoactive region of the pixel is known as the depletion region, in which the photons are converted to electric charges (63). The charge from each photosite is then held in a series of potential wells by gate electrodes and then transferred from well to well through the sensor grid. The charge collected in each pixel is relayed from pixel to pixel in columns first and then converted to a horizontal register using clock pulses. The optimization of a single readout has a uniformity advantage, but this also causes the speed at which the image is acquired to be limited. In terms of image accuracy, the image maintains order by reading the electrons as they arrive at the output node, which can lead to a few limitations which in turn slows down the image acquisition process. The first of which is there is only one output node per sensor, meaning each pixel passes through the one node. The second is if the electrons are transferred too fast they end up repelling each other, which can easily be prevented by slowing the transfer speed but this further delays the readout. The third is that the entire sensor must be cleared of all electrons before the next exposure to light.

- Time Delay and Integration (TDI) Operation

  The Time Delay and Integration, or TDI, is a method of image acquisition that achieves high speed, high resolution, and high sensitivity scanning over a wide area. This method of scanning has evolved from line scanning, in which a camera captures line after line of single rows of pixels. These cameras are plagued by low resolution, while TDI cameras maintain high resolution by scanning multiple lines. These lines are integrated from a single pixel row onto the next as the object passes through the sensor. The resulting image quality is greatly improved and does not have as concrete lighting requirements.

- Front and Back Illuminated CCD

  The main difference between a front and back-illuminated CCD imaging sensor relates to how the incident photons travel to the photoactive region of the sensor. In a front-illuminated sensor, the photons strike the same side of the device as the gate electrodes. This design leads to a limitation on how many photons reach the photosensitive region, due to there being a chance of gate electrodes absorbing or reflecting the present photons. Since there are fewer photons reaching the photosensitive region, this can typically limit the Quantum Efficiency (QE) to around 50% - 60%. In contrast, the back-illuminated CCDs have the photons striking the back side of the device. In order for this to work properly, the silicone substrate is thinned which allows access to the photosensitive region. Since there is no direct contact with the gate electrodes, there is less of a possibility of the electrons reflecting or absorbing, which results in a QE of up to 95%. These sensors are often characterized by a higher sensitivity and low noise (63).

- Multiport CCD

  One main disadvantage of using a CCD sensor is related to the limited readout speed as discussed above. The multiport CCD is a way to remedy this setback, designed for higher resolution cameras. This multiport (or multi-tap) technology divides the camera sensor into several different image areas which can be processed in parallel for increased readout speed. While this decreased processing time is enticing, this also comes with a trade-off of needing additional amplifiers and ADCs per additional sensor.

- Thermoelectrically Cooled CCD

  Within the CCD sensor, there is a linear relationship between increasing temperature and dark current. This in turn increases noise in the image and can lead to decreased image quality unless the heat is dissipated. An effective way to dispel heat from the sensor is using a built-in thermoelectric cooler. These coolers come in single- and multi-stage models and can maintain temperatures down to -70 degrees Celsius, depending on the amount of stages present. It is imperative the number of stages reflects the ambient temperature conditions.

- Full-Frame CCD

  In a full-frame CCD (FF CCD), the semiconductor region functions as both a photoactive element and a charge-transfer device. During the process of integration, the pixel location accumulates the electric charge in response to incident photons. After integration has been completed, the charge packets are vertically transported through the pixel locations toward a horizontal shift register. This specific sensor allows the entire surface of the CCD to be light sensitive, which in turn maximizes the number of pixels that are included, and therefore also maximizes the portion of each pixel that is capable of converting the photons into electrons. While this is a pretty big advantage, a major limitation of this system is a mandatory mechanical shutter to block the incoming light after the completed exposure period. This is due to the fact that all light must be drained from the photoactive areas, or else the charge packets would corrupt between readouts.

- Frame-Transfer CCD

  The Frame-Transfer (FT) CCD allows the camera to reap partial benefits of the FF CC, while almost entirely eliminating the need for a shutter. This can be accomplished by dividing the CCD into two equal sections, comprised of a normal light-sensitive imaging array and a storage array shielded from the incident light. After initial integration, the pixels' charge packets are quickly transferred to the storage array where the readout occurs. As the readout is processed, the active pixels can begin accumulating the charge for the next image. This leads to a higher frame rate compared to the full-frame CCD. Although these are functional without the shutter, they are subject to an effect called vertical smear, which comes from the transfer of charge packets not being fast enough, therefore leaving the image slightly altered. While this model comes with

advantages, a pretty major disadvantage is that the image quality is decreased, since the number of pixels is essentially cut in half.

- Interline-Transfer CCD

  This type of CCD camera builds on the FT CCD, except it aims to reduce the vertical smear as much as possible. This system works to eliminate the vertical smear by providing a more thorough storage network of regions that are adjacent to each photoactive region. After the initial exposure is complete, every charge packet in the sensor is simultaneously transferred to a non-light-sensitive vertical shift register (65). This system type creates an electronic shutter rather than a mechanical one and produces a very minimal smear. Additionally, integration occurs in tandem with the readout, maintaining a higher frame rate. Although these do not require the larger and more expensive storage options of the FT CCD, they are overall less efficient in the means of converting photons to electrons. This is due to each pixel location now having both a photodiode and a portion of a vertical shift register. This is because the pixel is now not entirely light-sensitive, and less electric charge is generated.

Table 3.8 *- CCD Comparison*

CCD Sensor Type	Key Features	Spectral Range	Applications
Back-Thinned	Visible, UV, Near IR; Line and area sensor options; High-speed type; Low noise type	Visible, UV, Near IR, DUV	Line scan cameras, spectrometers, image readout
Back-Thinned TDI	TDI gives high sensitivity; high-speed, continuous image acquisition; back-thinned structure ensures high sensitivity from UV to IR; multiple ports for high-speed line rate	UV to Near IR	Sequential imaging of moving objects; flow cytometry; semiconductor inspection; inspection tasks
Front-Illuminated	Line and area sensor options; high-speed response; high sensitivity; TDI operation	Visible, UV, Near IR, DUV	Foreign object screening; high-speed imaging; industrial inspection; low light level detection; spectrometers
Back-Illuminated	Enhanced NIR sensitivity; low readout noise	Enhanced NIR	Space telescope

X-Ray	100% fill factor; TDI operation; Low dark current; wide dynamic range; fiber optic plate with/without scintillator	X-Ray	X-ray imaging; non-destructive inspection; dental panorama
Multiport	Built in high speed readout amplifier; TE-cooled options; line rate 1000-10000 lines/s; data rate 5 - 40 MHz	Visible, UV, Near IR, DUV	Foreign object screening; machine vision; DNA sequencing; Raman Spectroscopy
Thermoelectrically Cooled	Single and multi-stage options	Visible, UV, Near IR, DUV	Intensive imaging systems
Full-Frame	Larger photoactive area, high image quality, mechanical shutter	Visible to Near IR	Astronomy, low-light imaging
Frame-Transfer	Resistive gate structure for high-speed transfer; low image lag	Visible to Near IR	Spectrophotometry
Interline-Transfer	Reduces vertical smear; fast non-mechanical shutter; high frame rate, lower fill factor	Visible	Consumer cameras, surveillance video

3.3.2.2 CMOS Camera Sensors

Unlike CCD sensors, the CMOS sensor concerts charge into voltage directly in the pixels, where the voltage amplification and quantization create the digital output (66). These camera sensors are known for high frame rates and excellent image quality and have often replaced CCD sensors since they can typically achieve these results with a lower required power. These sensors can read the electric signals much quicker, which is imperative for high-speed shooting sequences (64). In terms of manufacturing, CMOS sensors have the same basic structures found in computer microprocessors allowing for more efficient mass production efforts at a lower cost.

The CMOS sensor itself is made up of a grid of millions of small pixels, or photosites. Each pixel has different capacity levels that correlate to the amount of electrons that can be stored in it. At full-well capacity, this means that the maximum amount of electrons that can be held are stored within the pixel. This in turn produces the maximum amount of photons and a higher amount of electrons. When the pixel is at saturation capacity, this is the optimal level of photons and electrons to produce a quality image. When the image is oversaturated, it contains unnecessary information which can degrade the image quality. The absolute sensitivity threshold of the pixel is the minimum number of photons

that produce an image that can be distinguished from the noise gathered from the background. The lower this threshold is the more sensitive the camera sensor is, which can be helpful for low-light situations. With most electronic sensors, there is still a value the sensor picks up when there is no light present. This is called temporal dark noise, and as exposure and time increase electrons are generated without the presence of light. For most applications, a lower dark noise value is preferred. Once photons from incident light enter the photosite, the photodiode converts the light into an electrical current corresponding to the intensity of the detected light. After the signal is amplified, it is sent to an analog-to-digital converter (ADC), and the resulting digital information is sent to an image processor. The processor takes all of the digital signals and translates them into a cohesive final image, which is in part due to each pixel having an assigned value. This value correlates to the amount of light and the resulting electric charge. In order for color images to be created, a color filter array must be added, since the sensor only detects light, not color. The filter array contains a pattern of RGB filters in a grid formation, with one filter for each pixel. The filters are laid on top of the photosensitive areas so that each pixel is only exposed to red, green, or blue light.

- Front illuminated sensor

  In terms of illumination, there are two types of basic CMOS sensors. For the front-side illuminated (FSI) CMOS sensor design, all of the wiring and circuitry that is necessary for the electronic functions are on top of the photodiode. This was one of the original designs and was proven to be not as effective when designing newer models. The incident light has to travel between the wires to get to the photodiode beneath, which creates more noise in the images due to the interference.

- Back illuminated sensor

  In order to increase the image quality and overall effectiveness of the CMOS sensor, the back-illuminated (BSI) sensor was designed. By flipping the original FSI design, the need for the light to work its way through the wires is eliminated. Since there is no longer to obstacle of circuitry, there is a much greater surface area exposed to the incoming light. This results in a quicker readout speed and makes the most of an electronic shutter. This leads to much higher image quality, lower noise readout, and maximizes efficiency from the sensor.

- Stacked sensor

  Another variation of the CMOS camera sensor is the stacked sensor. While the BSI greatly increased the quality of the produced images, the goal of the stacked sensor is to increase the speed at which the images are produced. In original CMOS designs, the signal readout from each individual pixel must travel the strips of wiring to the other side of the sensor for proper processing. By introducing the stacked processors, the distance that the signals need to travel is drastically decreased which results in faster processing speeds. These stacked sensors consequently have better autofocus due to the faster readout and

heightened processing power. These qualities all result in the stacked sensors outperforming the BSI CMOS sensors in almost all basic performance functions.

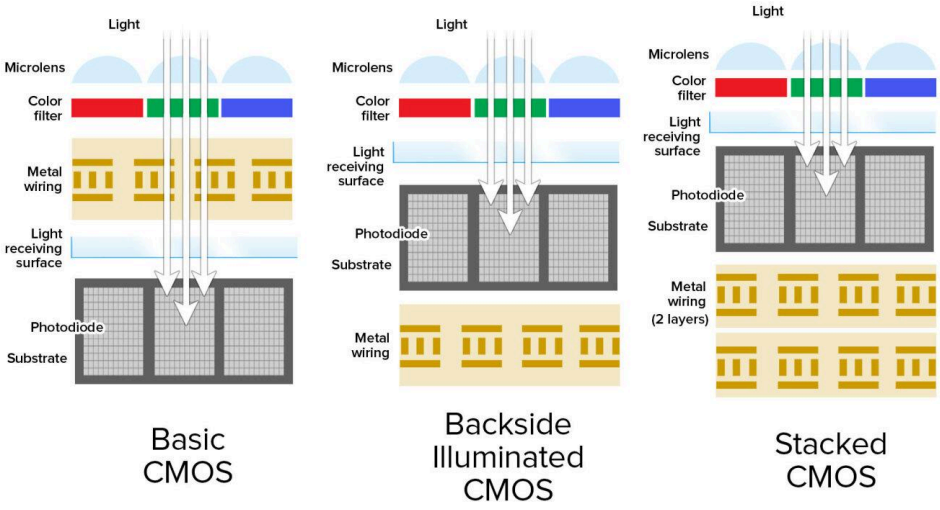


Figure 3.3 *- Front v back illumination v stacked (68)*

- Global shutter

  The global shutter system is normally added onto a stacked CMOS camera sensor. It has all of the benefits of heightened resolution and accuracy, and also has almost instantaneous scanning speed. The shutter comes close to having a 1/80,000-second capture speed, resulting in crisp photos at incredibly high resolutions. The tradeoffs for such high speed is a louder shudder noise, which in comparison is not as big a deal compared to low quality (68).

Table 3.9 *- CMOS Comparison*

CMOS Sensor Type	**Structure Description**	**Speed / Frame Rate**	**Noise Performance**	**Best Uses**
Front-Illuminated	The light must pass through internal wiring before reaching the photoactive area	Moderate	Moderate	General photography ; common in phone cameras
Back-Illuminated	Light directly hits the photoactive area; the inverse of the FSI system	High	Good	Low-light settings; astrophotography

Stacked	CMOS sensor and circuitry are stacked	High	Very Good	High-speed video settings; more advanced photography
Global Shutter	The image frame is captured simultaneously	Very high	Excellent	Fast motion capture; machine vision

3.3.2.3 CCD vs. CMOS Sensors

The CCD and CMOS camera sensors complete the same function quite similarly, but they both have distinct differences which can cause either one to excel in various applications. Both of the sensors convert light to an electric charge as the basic function. In the CCD sensor, all of the pixels' charges in the photoactive zone are sent into limited output nodes. More often than not, there is only one output node for voltage transfer into an analog signal (69). Output uniformity is a key component in achieving the high image quality. In regards to the CMOS sensor, each pixel undergoes the charge-to-voltage conversion, with add-ons built into the sensor to maximize the quality of the output digital bits. The addition of the other functions (amplifiers, noise-correction, digitization, etc.) makes the CMOS sensor more complicated, and therefore less of the photoactive region is available for the light capture process. Also since each pixel does its own conversion process, the uniformity is not present like in the CCD sensor but the parallel functionality still allows for effective high-speed processing. When these two sensors were introduced to the photography market in the late 1960s - early 1970s, the CCD sensor became the dominant method because of the higher quality images it produced with the available technology. Currently, CMOS and CCD sensors are both mainstream technologies each producing excellent results. For our project, we will be integrating a stacked CMOS camera sensor. The CMOS sensor is more compact, cheaper, and has better power consumption.

- Consumer Availability

  In the case of consumer applications, such as cell phone cameras, the majority of research has been focused on fine-tuning the performance of smaller camera sensors that can produce high-quality results. This development resulted in vast improvements in image quality along with decreased pixel size. Due to the ever-growing population as well as ever-evolving smartphone technology the demand for better camera quality has also increased. This intense demand increase has led to a high volume of imaging systems being put into production for smartphones and other camera equipment. With lower power consumption and higher integration capabilities, CMOS imaging sensors outperform CCD sensors in almost every performance parameter.

- Machine Vision Systems

  In terms of machine vision, many of the advancements that made consumer cameras excel have been invested in current technologies. This means that the majority of CCD sensors have also been passed up for more streamlined CMOS sensor technologies. Key factors in this selection are capture speed and noise production. In the case of CMOS sensors, the parallel linescan data pathway allows the amplifiers to maintain a low bandwidth. Once the signal reaches the interface between the imager and off-chip circuitry all of the data is digitized (69). High-speed CCD sensors do have parallel performance capabilities, but since they do not perform at this same level the resulting bandwidth is high which leads to a higher noise level.

- Near Infrared Imaging Systems

  To obtain images in the near-infrared (NIR) wavelength (700 nm to 1000 nm) the image sensor needs to have a thicker photon absorption region. This is due to the infrared photons absorbing deeper into the semiconductor's silicone than visible light photons. In this case, the CCD models specifically made for near-infrared capabilities are more sensitive to infrared photons than CMOS models. Since CMOS sensors are fabricated to high volume processes, they are designed to be mainly insensitive to the NIR range. Increasing the thickness of the photoactive sensor region degrades the ability to resolve spatial features if the thicker layer isn't paired with a higher pixel bias or a lower epi doping level. When you change the voltage or the epi doping level however, this affects the operation of the CMOS sensor in both the analog and digital circuitry.

- UV Imaging System

  When it comes to the UV spectrum, both options come with considerable trade-offs, however, the CMOS sensor is a slightly better option than the CCD sensor. Unlike the NIR sensor, ultraviolet photons are absorbed very close to the surface of the silicone image sensor. This means that a UV imaging system must not have thicker silicone layers, such as polysilicon, nitride, or thick oxide layers that would prevent proper photon absorption. In order to combat this, modern UV imaging systems are normally back-thinned with a thin AR coating on top of the imaging surface. Unfortunately, there is a very fine balance between an effective back-thinned model and a working UV imaging system. In order to produce a quality UV response, the surface of the imager must be carefully crafted and require specialty treatments, regardless of the sensor type. Back-thinned imaging sensors often have thicker oxide layers, which is not suitable for the production of UV images. Another option is the back-thinned surface is passivated with a highly doped boron layer, which goes too deep into the silicone layer (69). This in turn leads to most of the UV photons being lost in recombination, leading to a very poor image quality. As a general rule of thumb, a UV imaging response can be achieved in all varieties of line scan imaging systems, but not in all area imaging systems. When it comes to global shutter systems, the CCD sensor can not be back-thinned while the CMOS sensor can. This doesn't come without drawbacks,

unfortunately, as the storage node must be shielded when in a back-thinned sensor but only if the system is computing both visible and UV images. This shielding is not effective in protecting all parts of the pixel from the incident illumination without negatively affecting the sensor's fill factor ability. The other options for back-thinned global shutter CMOS sensors do not produce high-quality images, due to high noise, rolling shutter, or a lower full-well capacity.

- Time Delay and Integration Imaging System

  When it comes to time delay and integration (TDI) imaging systems, typically CMOS sensors produce higher-quality images. These image sensors are most commonly used in machine vision systems and work very similar to line scan sensors, but they often have hundreds of lines. These sensors are often implemented when signals are weaker since the multiple snapshots can be superpositioned to create a stronger signal. CCD TDI sensors have great sensitivity, but they eventually reach a speed limit. CMOS sensors have the speed advantage, but the voltage-domain TDI sensor is more complicated and therefore more costly to produce on a reasonable manufacturing level. Overall, the voltage-domain centered CMOS cameras provide the highest quality images.

- Cost Considerations

  In terms of the project, it is important to consider the cost breakdown of the two types of camera sensors. Camera sensors that are already available on the market are always cheaper than custom options. If there are customizations that are imperative to the system function, it is generally cheaper to develop a CCD sensor. Since CMOS sensors use more expensive components and typically have more circuitry, they are more expensive as a result. Since the goal of our project is to be consumer-friendly, we will be choosing an already existing camera sensor to keep costs more reasonable.

Table 3.10 *- CCD vs CMOS Comparison*

Feature	**CCD**	**CMOS**
Image Quality	Good image quality	Superior image quality in multiple scenarios
Readout Speed	Slower compared to most CMOS models	Faster readout due to parallel functionality
Power Consumption	Higher power consumption	Lower power

Cost	Overall less expensive; faster discontinued models can lead to more expenses down the line	More expensive components for custom sensors; high production volume makes market sensors cheaper
Size	Sensor modules are larger	Compact models are more readily available
Shutter	Some models can avoid rolling shutter effects	Global shutter should be in place to prevent rolling shutter issues
Lighting	Back-thinned for low-lighting Daylight operation is optimal	Back-illumination for low light Daylight operation is optimal
Hardiness	With proper casing, general weatherproofing can be achieved	More compact module design makes weatherproofing easier
Motion	A frame-transfer model can help balance out the ship's constant movement	Global shutter is best for constant ship motion

3.3.2.4 Camera vs. Radar

The main difference between camera sensors and radar sensors is that cameras are passive technology while radar is active. The camera sensor detects the natural radiation from the environmental light and creates the resulting image (61). While radar can function perfectly fine in complete darkness, a camera sensor needs ambient light to perform basic imaging functions. In addition to lighting being a key component of the effectiveness of a camera system, weather also plays a big part in the quality of images produced. If rain beads up on the lens, it can affect the resulting image and prove ineffective for the main goal of the project. That being said, a camera system would be more energy efficient, but since it is not as effective in other crucial parts of this project we will be using a radar system.

3.3.3 Fiber Optics Communication

Another method that was considered for this project was using fiber optics for communication methods. Optical fibers are typically made from glass, and use transmitted light to convey information. Fiber Optic system are often regarded as being superior to their metallic-conductor counterparts due to much larger bandwidth capabilities (70). Additional advantages to fiber-based communication systems are electrical isolation, exemption from electromagnetic interference, low power loss, and

being generally more lightweight compared to other transmission systems. Since fiber optics are a light-based system, a grounding connection does not need to be established. As a result, the transmitter and receiver are isolated from each other and do not experience the constraints of a ground loop. Some of the general applications of fiber optics systems are communications, sensing, power delivery, and illumination.

3.3.3.1 Fiber Optics Communication vs. Radar

The main difference between fiber optics and radar communications is the transmission medium. For radar communication, the signal is transmitted through space or the air. The transmission relies on the signal being reflected between the transmitter and receiver to function properly. In terms of fiber optic transmission, the visible light is internally reflected in the optical fibre. In terms of our project's goals, fiber optic transmission is not a feasible option. The main goal is to create the AIS system overlay using the radar sensor, and with fiber optics this is not possible. Radar is dependent on the signal reflecting off of surrounding objects, while fiber optic networks are based on total internal reflection. Also, since ships are not stationary, a fiber optic transmission network for communication would not be feasible either.

3.4 Additional Ship Identification System

The original desired outcome of optical integration into this project was the intention to reproduce the same outcome that was outlined by the radar system. However, with our time and budgeting constraints, as well as overall greater disadvantages, it is not feasible to recreate the system using optical technologies. In an ideal scenario, we could work to replace the radar with a LiDAR scanning system however this comes with an extensive amount of troubleshooting, specifically relating to weather-dependent functionality.

In an effort to still include optics in the project's design, we are adding an additional imaging system to the main system, in addition to the AIS tracking overlay. The goal of this secondary imaging system is to provide additional information to operators that highlight suspicious vessels while offshore. The purpose of the camera will be to identify and capture images of the names and physical attributes of ships that may be deemed suspicious or do not have AIS tracking enabled. This imaging system will be comprised of a stacked CMOS camera sensor and a stationary magnification lens system, similar to a telescope setup. This camera will be mounted on a motor for a greater horizontal range of motion, which allows for greater surveillance capabilities.

3.4.1 Camera Sensor Selection

As stated within the contents of section 3.4.2 there are different advantages and disadvantages to the multitude of camera sensors available on the market. When considering maritime environments, as well as mounting height, we must take into account a property called atmospheric turbulence. Similar to how heat waves create shimmery distortions above high-temperature surfaces, images of distant objects are distorted in telescopes due to variations of heat in the Earth's atmosphere (71). Due to the temperature variations, the light moves in slightly different directions in different places on the entrance aperture, otherwise known as random fluctuations of the refractive index

of the light. These variations create distortions through the lens onto the final image. This effect must be kept in mind when taking into account the height of the imaging system and the operating distance we are aiming for. In terms of the sensor itself, there is no damage to the hardware, the distortion effect takes place as the light hits the lens, but this does still lead to the sensor capturing the distorted image.

The average height of a merchant marine vessel is around 35 meters. This height correlates to the ground layer of the atmosphere, where atmospheric distortion is the strongest (72). This means that it is incredibly important to keep the atmospheric turbulence in mind. In order to combat atmospheric turbulence, post-processing can be done using software, adaptive optics can be integrated into the sensor, or we can choose a sensor that operates in the NIR spectrum. For the optical system, it would be best to utilize a stacked CMOS sensor, due to the higher image quality and more compact design. In order to select a specific CMOS sensor, a few qualities need to be kept in mind: pixel size, sensor size, resolution, and camera mounts (73).

- Pixel Size

  Each camera sensor is made up of individual pixels that all collect incoming photons. The sensor being made up of individual pixels also helps the sensor determine where the incident photon is coming from (74). Pixels come in different sizes, which has an effect on the sensitivity and resolution. For larger pixels, more photons are collected which increases the sensitivity of the sensor but this decreases the resolution. For smaller pixels, the sensor is able to obtain a higher resolution but does not have as high of sensitivity. Generally speaking, a higher pixel count does make better image quality.

- Sensor Size

  Sensor size is determined by the number of pixels present. By taking the surface area of the sensor and dividing it by pixel size, you are able to find the size of the sensor. Similar to pixel size, the general rule of thumb for sensor size is also a larger sensor and better image quality. This is due to the fact that a greater surface area of the sensor normally correlates to larger pixel sizes, and more photons being collected.

- Resolution

  The resolution of a camera is the ability of the imaging device to resolve two points that are close to each other (74). When the resolution is higher, the smaller the details that can be obtained in the final image. Resolution is affected by pixel size, magnification, the lenses, and the Nyquist limit. The following equation can help determine resolution: $Resolution = (\frac{pixel\ size}{magnification}) * 2.3$, where the 2.3 correlates to the Nyquist Limit. This limit determines whether a sensor can differentiate between two neighboring objects. This quality is going to be very crucial in choosing a sensor for our system design, since we are working with farther distances the resolution should be high to be able to make out smaller details on the ship.

- Camera Mount

  There are different mounting methods when it comes to attaching a lens system to a camera sensor. The four main types are C-Mount, CS-Mount, F-Mount, and Mxx-Mount. The lens flange focal distance (FFD) and the camera mount flange distance must be exactly the same or else there could be focusing issues that arise within the system (73).

An important part of imaging through the telescope is understanding angular resolution. This refers to the smallest angle that can be perceived within an image [94]. This is incredibly important for the details that we are aiming to capture.

Table 3.11 *- Mounting Camera Comparison*

Mount Type	**Structure Description**	**Mechanical Depth (FFD)**	**Diameter**	**Best Uses**
C-Mount	The most common mount;	17.526 mm	12.54 mm x 32 TPI	Very common
CS-Mount	5 mm shorter than C-Mount	12.526 mm	12.54 mm x 32 TPI	Precise back focal distance
F-Mount	Bayonet-style mount; designed for 35 mm format cameras; no back focal adjustment possible	Vaires	Varies	Cameras with bigger sensors
Mxx-Mount	MXX relates to the diameter, thread pitch, and FFD (i.e. M72)	Varies	Varies	Alternative to F-Mount for larger sensors

3.4.2 Lens System

For the proposed imaging system, we are implementing a stationary lens magnification system. Our ideal parameters for this system are: 7.5x magnification, a telescope-based lens system, and an operating distance of 5 -7 km. The goal is to take pictures of ships that the system encounters while offshore as an additional safety net, so the system should be able to operate at a medium distance range, we are not looking to image the horizon. In order to effectively choose a lens system there are a few parameters we need to further explore: field of view, lens resolution, magnification, focal length, and working distance.

- Field of View

  The field of view (FOV) references the viewable area that is able to be imaged by the lens system. This area is the portion of the object that fills the sensor of the camera (75). It can either be a horizontal/vertical distance (mm) or an angular distance [AFOV] (degrees). This viewable area is influenced by the focal length of the lens and the size of the sensor. The FOV can be determined using the following equations: $Horizontal\ FOV = 2 * WD * tan(\frac{AFOV}{2})$ (76) $AFOV = 2 * tan^{-1}(\frac{Sensor\ Horizontal}{2 * f})$ (76). Since our lens system will have a fixed magnification value, the equation to find FOV is $FOV = \frac{H}{M}$, where H is the sensor size and M is the fixed magnification value (75).

- Angular Resolution

  While it is important to consider the resolution of the camera sensor itself, there is also the resolution of the camera lens that must be considered as well. A lens's ability to accurately resolve an image is limited by diffraction. As light travels through the lens aperture, it diffracts before hitting the sensor which creates a diffraction pattern. This pattern is known as an Airy pattern, with the bright spot in the middle being called the Airy disk. This disk has an angular radius given by $\theta = 1.22(\frac{\lambda}{D})$, where D is the diameter of the lens in meters. It is important to note that different points on an image produce different Airy patterns and the distance between points affects its ability to be resolved. If the angular separation between two points is greater than the angular radius of their Airy disk resolution is possible, otherwise the image merges (74). Consequently, the diameter of the lens determines resolution and has an inverse relationship to the angular radius of the Airy disk.

- Magnification

  The magnification (M) of an optical system is defined as the ratio between the image (h') and object (h) size (73). Magnification plays a big part in enhancing image clarity and resolution which helps bring smaller details into focus. Our system will have a fixed magnification value of 7.5x which means the ratio of the two lenses will have to equal 7.5.

- Focal Length

  The focal length of a lens is the foundation for a clear and focused image. It is defined as the distance over which collimated rays coming from infinity converge to a point along the optical axis (73). Lenses that have positive focal lengths (convex) converge the light while lenses with negative focal lengths (concave) diverge the light (75). A fixed focal length lens has a fixed AFOV. If the lens is focused with different working distances, different sized FOVs can be achieved

but the viewing angle will always remain the same. Focal length can be typically found by the following equation: $f = \frac{(H * WD)}{FOV}$ using working distance, the object size, and FOV.

- Working Distance

  The working distance of a lens system is the distance from the object to the lens where the image is at its sharpest focus, often shown as *s*. There is a very useful relation between working distance, magnification, and focal length: $s = \frac{f(M-1)}{M}$ (75).

3.4.2.1 Keplerian Telescope System

Out of the two types of telescopes, the Keplerian system is normally the easier one to construct. The two lenses that make up the system are both convex, meaning they both have a positive focal length. Since both of the focal lengths are positive, the overall length of the system can be found by adding the two values together (91). In order to increase the length of the system, a biconvex lens can be added in between the eyepiece and objective lenses. This telescope is typically considered to be a combination of an objective and a magnifier. One downside of these types of lens systems is that they produce an inverted image. This can be remedied with software, by flipping the camera sensor, or adding a third lens. In terms of our project, the extended length due to the addition of the focal lengths may also be a disadvantage. On the contrary, an advantage to using a Keplerian system is that it can produce a higher system magnification, which can be helpful for the longer distances we are looking to cover.

3.4.2.2 Galilean Telescope System

In a Galilean model telescope, there are normally only two lenses; one with a positive focal length and the other with a negative focal length. As a result of the opposing focal length signs, the distance between the lenses becomes shorter than in the Keplerian model. In terms of our project, this will be the better option since we can get a shorter lens system overall (91). In addition to the advantage of the shorter system, the output image is upright meaning no additional lenses or post-processing have to be done. This type of lens system can also help reduce spherical aberrations introduced by the lenses.

3.4.2.3 Achromatic Doublet Lens

Besides the two previously mentioned telescope lens formats, there is another lens system that can be brought into consideration. This would be a lens setup consisting of one achromatic doublet lens. An achromatic doublet lens is a single lens that comprises a biconvex lens and a biconcave lens. This lens properly focuses the incoming image onto the camera sensor and produces a high quality image. It also produces a smaller spot size

than the other two systems, and therefore it can better correlate to the pixel size of the camera sensor. This lens setup also helps reduce chromatic aberrations, and has a narrow field of view.

3.4.2.4 Lens Systems Comparison

Table 3.12 *- Lens Comparison*

Feature	Galilean Telescope	Keplerian Telescope	Achromatic Doublet
Objective Focal Length	Positive	Positive	Positive
Eyepiece Focal Length	Negative	Positive	N/A
Field of View	Narrow	Wider	Narrow
Relative Length	More compact	Longer	Longer
Magnification	Low magnification	Higher magnification	Lower magnification
Image Orientation	Upright image	Inverted image	Upright image
Aberrations	Chromatic (more at higher power); spherical	Chromatic; spherical	Spherical
Exit Pupil Location	Inside of telescope	Outside of telescope	N/A
Common Lens Count	2	3 (additional lens for flipping image)	1

For our project, the achromatic doublet lens system would be the most advantageous for the criteria we are aiming to fit.

3.4.3 Adaptive Optics

In order to clearly image boats along the horizon, we could implement incredibly sophisticated Adaptive Optics. The integration of these optics is a stretch goal of this project, as they are incredibly expensive and not easy to implement in a smaller system like ours. In an ideal scenario, these advanced technologies would be implemented into the Galilean telescope system to counteract atmospheric turbulence. Adaptive optics consist of deformable mirrors controlled by computers, which implement real-time correction for distortions caused by the Earth's atmosphere [92]. These corrective

systems can produce images just as sharp as space telescope systems, allowing for much finer details to be analyzed. In order to get the incredibly detailed images, there must be a seamless integration of hardware and software systems. Different modes of adaptive optic technologies have been developed, each achieving a different goal.

- Single Conjugated Adaptive Optics (SCAO)

  Standard adaptive optics performance is made possible by the Single Conjugated Adaptive Optics (SCAO) mode. In this format, the atmospheric turbulence corrected is limited to a specific cylindrical projection. This projection is the telescope aperture faced towards the guide star being viewed. This method produces a very high quality image, at the cost of a small field of view. Another downside of this method is the amount of the night sky you can clearly see is decreased, due to limitations on the amount of naturally available guide stars. SCAO can be combined with coronagraphy to achieve extreme spatial resolution as well as image contrast. This mode acts as a base for other modes of adaptive optics.

- Single Laser Adaptive Optics (SLAO)

  By adding a laser “guide star” to the base SCAO mode, we get Single Laser Adaptive Optics (SLAO). This mode is overall better than the SCAO mode, but comes with its own limitations due to the introduction of the laser into the system. One of these setbacks is the cone effect, which is due to the laser only being projected for a finite distance. Another performance limitation is anisoplanatism, which comes from only one turbulence slice being sensed and corrected.

- Ground Layer Adaptive Optics (GLAO)

  While the SCAO and SLAO modes employ turbulence correction along a single line of sight, the Ground Layer Adaptive Optics (GLAO) moed employs turbulence correction across the whole field of view at ground level. In the lower levels of the Earth’s atmosphere, the turbulence creates the most interference, making the GLAO mode very useful in image correction. In this mode, multiple off-axis wavefront sensors are used to obtain an average of the ground turbulence and this average is corrected by a single deformable mirror. With the advantage of a much larger field-of-view correction comes the trade-off of relatively low performance.

- Multi-Conjugate Adaptive Optics (MCAO)

  To go a step further in correcting both ground-level and high-altitude turbulence, Multi-Conjugate Adaptive Optics (MCAO) comes into play. This setup uses multiple mirrors to achieve this goal, employing quite a few off-axis wavefront sensors to read the turbulence in several conditions. The collected measurements are given to a tomographic reconstructor to estimate the 3D turbulence value, which is then projected onto the deformable planes to generate control commands. The MCAO mode produces a medium field-of-view and medium-to-high performance values. This mode of adaptive optics is laser-guide-star-based, which

means there is high sky coverage, but also a need for several natural guiding stars to stabilize the 3D turbulence.

- Laser Tomography Adaptive Optics (LTAO)

  This technique is comparable to taking a CT-scan of Earth's atmosphere, reassembling the image layer by layer. In Laser Tomography Adaptive Optics (LTAO) the telescope reconstructs the turbulence adjusted image above it piece by piece. This mode is a combination of the SCAO and MCAO modes, implementing several off-axis wavefront sensors but optimzed to analyze the center data of the field-of-view. The main advantage in regards to the SCAO mode is increased field-of-view. This is so the on-axis turbulence cylinder can be better sampled and analyzed, providing a correction using a single deformable mirror. All of this results in high performance images across a smaller field-of-view, with a trade off of tomographic error and low order residuals.

- Multi Object Adaptive Optics (MOAO)

  Similarly to the LTAO and MCAO methods, Multi Object Adaptive Optics (MOAO) uses the 3D turbulence to tomographically reconstruct the adjusted image. The overall goal of MOAO is to obtain multiple small-field turbulence corrections in an overall larger field-of-view. Compared to the other modes, in MOAO the turbulence is corrected for along the line of sight upon which the object is located. This is achieved using multiple wavefront sensors, natural and/or laser-guide-star, across the FOV. Each channel is corrected by its own designated deformable mirror which is closely conjugated to the ground level.

Table 3.13 *- Adaptive Optics Comparison*

Adaptive Optics Mode	**Corrected FOV**	**Performance**	**Coverage**
SCAO	Small (10 - 15 arcsec)	High	Small
SLAO	Small (10 - 15 arcsec)	Medium/High	Small
GLAO	Large (several arcmin)	Low	Large
MCAO	Medium (1-2 arcmin)	Medium/High	Large
LTAO	Medium (1-2 arcmin)	Medium/High	Large
MOAO	Large (several arcmin)	Medium/High	Large

3.4.4 Gimbal

In order for our optical system to be as effective as possible, we need to implement some range of motion for the camera. In order to do this, implementing a gimbal would be most effective. The gimbal can provide a sufficient range of motion, which is beneficial for the surveillance purposes of this system. In addition to adding maneuverability to the imaging system, the gimbal also acts as a counter to the constant motion of the ship on the waves. The gimbal constantly self corrects to keep the camera at a locked position. Something to integrate into future renditions of the design would be an algorithm to lock onto the ship, given a set of coordinates. While a gimbal with a full 360-degree rotation range would be incredibly useful in our project, finding one that fits the size of our camera sensor is incredibly difficult. These models are an exorbitant amount of money due to the precision they have to follow, and there are very few available at such a small size. There are two main types of gimbal models for us to consider, azimuth and elevation configurations. Azimuth represents the horizontal rotation of the gimbal, moving the camera from left to right. Elevation represents the vertical movement of the camera, tilting up and down. Gimbals are a two or more axis mount, and come in a variety of configurations. There are a few key differences in these two mounting methods, but they both serve the same purpose, and ultimately lead to the same end result.

- Altitude Over Azimuth (Alt-Az)

  This configuration is made up of a fixed azimuth axis and an altitude axis attached to the rotating part of the azimuth structure. There are two lines of rotation, one vertical and one horizontal. This specific configuration is most similar to how people use their eyes to scan the horizon. The Alt-Az system is most commonly used in surveying and landscape camera mounts.

- Altitude Over Altitude (Alt-Alt)

  This configuration is made up two structures; one that rotates along a horizontal axis (the “X” or “Major” axis) and a rotation stage that is attached to the major axis (the “Y” or “Minor” axis). This secondary rotation stage has an axis of rotation that is perpendicular to the major axis. The Major axis is normally in a North-South alignment while the Minor axis is typically in an East-West alignment. These two structures result in an up and down motion of the camera. This configuration is typically used for tracking satellites.

- Equatorial

  This configuration is very similar to the Alt-Alt configuration, however the Major axis is instead inclined at an angle that is equal to the site latitude, as well as the azimuth alignment is true North. These changes result in the Major axis having a rotational axis that is parallel to the Earth’s rotational axis. Due to this key characteristic, equatorial configurations are commonly found in astronomical telescopes since the rotation counteracts that of the Earth.

For our system, the base station will be quite high up on the ship, meaning there will need to be adjusting of the vertical tilt to account for height discrepancies. In addition to this, we will need to make use of the horizontal rotation in order to provide accurate photos. Therefore, we will need to implement both the Alt-Az and Alt-Alt configurations to provide as large of a rotational field as possible. The best way to implement this would be a simple pan-tilt gimbal, which we can directly mount the system on. This will provide horizontal and vertical movement capabilities and will result in better photos being taken. In addition to this, we can find a gimbal that directly connects to the Raspberry Pi for easy integration.

Although the imaging system is meant to be mounted under the base of the radar, for the purpose of the main demonstration of the project a handheld gimbal will be used to simulate the mounting environment.

3.4.5 Component Selection and Comparison

Taking all of the above information into account, we are able to make some decisions on the specific components to select for the imaging portion of this project. For the CMOS sensor we wanted something with higher resolution and a global shutter. By searching through the Edmund Optics database we were able to narrow down the sensors to three top options:

Table 3.14 *- CMOS Sensor Comparison*

	Specifications		
Parameters	iDS U3-3864LE-C-HQ-AF, USB 3.0, 1/2.8" Color Camera	Allied Vision Alvium 1800 U-050c, 1/3.6" 0.5MP C-Mount, CMOS Sensor	Basler ace2 a2A2590-60ucPRO Color USB3 Pro Camera
Sensor Format	1 / 2.8”	1 / 3.6”	1 / 2.8”
Resolution	2.10 MP	0.50 MP	5.00 MP
Power Usage	1 - 1.5 W	1.5 W	3.4 W
Frame Rate	135 fps	116 fps	60 fps
Shutter	Rolling	Global	Rolling
Price	$462.24	$312.12	$477.36

As shown above, we chose the Allied Vision Alvium camera. While it has a lower resolution, it has the global shutter, a midrange fps, and a lower power consumption. It is also not the most expensive of the sensors that were researched. The iDS sensor was another favorable option but the model that was used for testing was monochromatic, therefore the Alvium was the most favorable option.

Table 3.15 *- Gimbal Comparison*

	Specifications		
Parameters	Adafruit Mini Pan-Tilt Kit	SCORP-C Camera Stabilizer 3-Axis Mirrorless Gimbal	DJI RS 3 Mini, 3-Axis Gimbal Stabilizer
Connectivity	Raspberry Pi	Handheld	Bluetooth
Mount Dimensions	37mm x 33mm x 3mm	Adjustable slider plate	Adjustable slider plate
Rotation Range	180°	360°	80°
Tilt Range	150°	+120° ~ -201°	80°
Cost	$18.95	$249.00	$237.00

The model that was selected for the imaging system prototype was the SCORP-C 3-Axis Mirrorless Gimbal. This model was made to support heavy DSLR cameras with large lens systems, so it would be more than capable of supporting the smaller CMOS sensor and lens setup. It has a full 360 degree rotational range of motion, providing maximum maneuverability for any adjustments needed. While it was the most expensive option of the models researched, it was definitely the most equipped to produce the desired performance.

Table 3.16 *- Lens System Comparison*

	Specifications		
Parameters	Thorlabs AC254-400-A & Newport -75 mm Biconcave	Thorlabs AC254-300-A - Achromatic Doublet	Newport -75 mm Biconcave & 250 mm Biconvex
Diameter	Both 25.4 mm	25.4 mm	Both 25.4 mm
Simulated Spot Size	0.055 um	0.183 um	11.267 um
Simulated Image	Blurry	Clear	Very blurry
System Length	405 mm	303 mm	253.5 mm
Cost	$157.34	$95.34	$130.50

When choosing the lens system, Zemax played a huge part in helping to determine compatibility. By using the spot size and image simulation options, it was easy to choose

which system would work the best. In this case, the achromatic doublet lens proved to be the best option, with a small spot size and a clear simulated image.

3.5 Embedded Systems

3.5.1 Single Board Computer

When selecting a single board computer (SBC) for our project, we needed to consider multiple factors, including processing power, connectivity, energy efficiency, and compatibility with the other hardware components. The SBC is essentially the central control unit for onboard operations, managing data transmission from the radar system, interfacing with Starlink for communication and optical system for visual data, and handling the necessary computations for data processing and storage. With the environmental and power constraints of our system, it was crucial to choose a board that is power-efficient and capable of running the required software.

To ensure we select the best option, we researched several SBCs that are commonly used in embedded systems and even maritime applications. There are a few options that we compared and considered for our project: the Raspberry Pi 4 Model B, the Arduino Uno, and the BeagleBone Black. Review the table at the end of the section to view the comparison of these options. Each of these SBCs can and have been used in many different applications and they offer several unique advantages in terms of processing power, ease of integration, and peripheral support.

The Raspberry Pi 4 Model B is known as a powerful and versatile single-board computer that is frequently used in IoT and embedded applications. It has a Broadcom BCM2711 quad-core Cortex-A72 processor, up to 8GB of RAM, and multiple USB and GPIO ports for hardware interfacing. This makes it excellent for handling communication protocols such as NMEA 2000, processing radar data, and managing the software required for data transmission via Starlink. This board has many built-in quality of life features, including 4 USB ports, 2 micro HDMI ports, and a USB-C power supply. The Raspberry Pi also has a strong developer community, so there is always access to extensive documentation and software support. One of the biggest advantages of the Raspberry Pi is its ability to run a full Linux operating system, so there is a lot of flexibility in software development.

Another SBC we looked into is the Arduino Uno. It is a microcontroller-based board that makes it very efficient for low-power applications and sensor interfacing. This MCU board operates using an ATmega328P microcontroller and provides a set of digital and analog input/output pins that can be used for reading sensor data and triggering hardware responses. While the Arduino is excellent for simple control tasks, it is much less powerful than the other options because it lacks the processing power and multitasking capabilities that we need for complex data processing and communication tasks in our system.

The board we considered was the BeagleBone Black that is comparable to the Raspberry Pi in terms of performance and capabilities. It has a Texas Instruments AM335x ARM Cortex-A8 processor, onboard eMMC storage, and extensive I/O options. One of the

biggest advantages of this board is its real-time processing capability, making it useful for industrial and robotics applications. However, it has a much smaller community of engineers compared to the Raspberry Pi, limited software support, and lower processing power. Since we are looking for a widely supported and powerful computing platform, we determined that the BeagleBone Black was not the best fit for our project.

Table 3.17 *- SBC Comparison*

	Specifications		
Parameters	Raspberry Pi 4	Arduino Uno	BeagleBone Black
Processor	Broadcom BCM2711 (Quad-Core Cortex-A72)	ATmega328P (8-bit Microcontroller)	TI AM335x (Cortex-A8)
Clock Speed	1.5 GHz	16 MHz	1 GHz
RAM	2GB, 4GB, or 8GB LPDDR4	2 KB SRAM	512MB DDR3
Storage	MicroSD	32 KB Flash	4GB eMMC + microSD
Connectivity	USB, HGMI, GPI, Ethernet, Wi-Fi, Bluetooth	GPIO, I2C, SPI, UART	USB, HDMI, GPIO, Ethernet
Power Consumption	3W - 7W	0.5W	2W - 3W
Operating System	Linux-based (Raspberry Pi OS, Ubuntu, etc.)	No native OS (runs on firmware)	Linux-based (Debian)
Cost	$35 - $75	$20 - $30	$55 - $80

After evaluating these three options, we decided to use the Raspberry Pi 4 Model B as the main single board computer for our system. Its combination of processing power, community support, and compatibility with Linux-based operating systems makes it the most suitable choice for our needs. The Raspberry Pi will be responsible for interfacing with the radar system, handling data processing, and managing communication with Starlink. This MCU board also supports various programming languages, including Python and C++, which gives us some flexibility in software development. This combination of accessibility and power made it the ideal choice for our project's needs.

3.5.2 Programming Languages

The selection of an appropriate programming language is crucial when designing the software architecture for our maritime domain awareness system. The software running on the onboard Raspberry Pi will be responsible for managing data communication between the Raymarine Cyclone Radar, NMEA 2000 interfaces, Starlink connectivity, along with the optical system. This means that the programming language we use should be efficient, well-supported, and compatible with embedded systems. It should also be able to handle network communication, process real-time data, and support the necessary libraries for hardware interfacing.

In order to determine the best programming language for our system, we considered multiple factors such as performance, ease of development, compatibility with embedded systems, and community support. We chose to compare four commonly used languages for embedded systems and Raspberry Pi applications: Python, C++, Java, and Rust. View the table in this section for a comparison of these options.

The first language we considered was Python. It is one of the most popular languages for embedded systems because of its versatility, extensive libraries, and strong community support. It is frequently used for prototyping, scripting, and high-level automation. Python is also very flexible, making it easy to interface with the Raspberry Pi since it supports GPIO manipulation, network communication, and serial connections. Python also has libraries like PySerial, PiGPIO, and CANard which will allow for easy integration with our NMEA 2000 interface and optical system. However, Python does come with some drawbacks. Since it is an interpreted language, it is generally slower than compiled languages like C++ or Rust. This can sometimes affect real-time data processing, but since our system does not require ultra-low latency responses, Python remains a viable choice.

The second language we looked into was C++. It is known for its high performance and efficiency, making it a good choice for embedded systems development. C++ is a compiled language that provides low-level hardware control and direct memory management, which can be useful when interfacing with radar hardware and NMEA 2000 devices. Unlike Python, C++ is a better fit for applications that require real-time data processing and optimized performance. On the other hand, C++ has a steeper learning curve compared to Python, and development generally takes longer because you have to manually manage memory. Also, debugging and maintaining C++ code can be more complex, especially in a project like ours with multiple software modules.

The third programming language we considered is Java. This is a platform-independent language that is often used for networking applications and large-scale software projects. One of the most useful features that Java provides is garbage collection, but also includes built-in multithreading, and high security, making it a good choice for networked systems like ours. Java also offers networking libraries that would help with Starlink communication and remote monitoring features. However, Java is not as efficient for embedded systems compared to C++ or Python. It requires a Java Virtual Machine, which would increase resource consumption. Since our project is running on a Raspberry Pi with limited power, Java may not be the most efficient choice.

The final programming language we considered was Rust. This is a newer language that has gained popularity for systems programming and embedded applications. It is designed for high performance, memory safety, and concurrency, which is great for low-level hardware interaction. Unlike C++, you can avoid common memory management issues with Rust because it enforces strict compile-time safety checks. While Rust stands out in performance and reliability, it has a smaller developer community compared to Python or C++ since it is newer. The learning curve is also steep, and there are fewer libraries available for Raspberry Pi development compared to other languages.

Table 3.18 *- Comparison of Programming Languages*

Parameters	**Python**	**C++**	**Java**	**Rust**
Performance	Low	High	Moderate	High
Ease of Use	Very High	Moderate	High	Low
Memory Safety	Automatic	Manual	Automatic	Enforced
Embedded Suitability	Moderate	High	Low	High
Library Support	Very High	High	Moderate	Low
Power Efficiency	Moderate	High	Low	High
Community Support	Very High	High	High	Moderate

After comparing all four programming languages, we selected Python as the primary language for our Raspberry Pi software development. While C++ would provide better performance, Python's ease of development, wide library support, and strong integration with Raspberry Pi hardware make it the best choice. Additionally, Python's compatibility with NMEA 2000 communication, Starlink API integration, and embedded control aligns well with the needs of our system. Using Python will allow us to rapidly prototype, test, and deploy our software while maintaining flexibility for future modifications.

3.6 Technology Stack

Developing a web-based system for maritime tracking requires selecting the right combination of technologies to ensure real-time data processing, reliability, and scalability. The choice of a full-stack framework influences how data is collected, transmitted, stored, and displayed within the system. With multiple options available, selecting a technology stack depends on factors such as performance, ease of integration, documentation support, and compatibility with external APIs.

For this project, the technology stack must support real-time radar and AIS data processing, allow for secure ship-to-shore communication, and provide an interactive dashboard for visualizing detected vessels. In the following sections, we will evaluate different frontend, backend, and database technologies before justifying our final

selection. The goal is to ensure that the chosen stack not only meets the project's requirements but is also adaptable for future expansions and enhancements.

3.6.1 Frontend

The frontend serves as the user interface for the maritime tracking system, allowing visualization of real-time vessel data, interaction with maps, and monitor alerts. Because the frontend directly impacts the user experience, it is important to evaluate frontend technologies on both their usability and compatibility with our requirements. We focused on metrics that reflect development speed, visual performance, learning curve, and integration ease.

***Table 3.19** - Frontend Technology Comparison*

	Vue	**Angular**	**React**
Learning Curve	Low – beginner-friendly with clear documentation	High – steep curve due to TypeScript and framework complexity	Moderate – easier to start, but deeper concepts (hooks, state) add complexity
Scalability	Moderate – best for small to medium apps, scalable with effort.	High – full-featured and structured for enterprise-scale apps	High – flexible architecture supports large apps
Flexibility	High – can be used progressively or fully	Moderate – opinionated structure reduces flexibility	Very High – unopinionated, integrates with many tools

3.6.1.1 Vue

Vue is a popular frontend JavaScript framework known for its incremental adoption model, meaning it can be gradually integrated into an existing project or used for full-scale applications. Vue is designed to be lightweight, flexible, and intuitive, making it a strong contender for modern web development.

Vue supports routing through its official library called Vue Router. This is often included when setting up a project using Vue Command Line Interface or Vite. Vue Router allows developers to define routes tied to specific components and supports features like nested routes, route guards, and dynamic route matching. In more advanced use cases, such as multi-user dashboards or analytics views. The Vue Router offers the flexibility needed for complex navigation structures.

However, our software team has no prior experience with Vue, which was something we took great consideration. While Vue showcases a vast community and extensive documentation, the learning curve and potential adaptation time played a role in our evaluation. Given the tight project timeline and the need for rapid development, we opted for a framework with which we had more familiarity.

3.6.1.2 Angular

Angular is a TypeScript-based frontend framework developed by Google. It is widely used for building large scale enterprise applications due to its built-in features such as two-way data binding, dependency injection, and structured architecture. Unlike React and Vue, which are more flexible and lightweight, Angular follows a strict Model-View-Controller architecture, making it better suited for projects requiring strong code organization and maintainability.

Angular includes a built-in routing module as part of the core framework. Routing is configured using route definitions in the application module, which map URLs to specific components. Angular's router supports lazy loading, route guards, and deeply nested routes, making it suitable for large scale applications. While Angular's routing is powerful, it can also be more complex to set up compared to lightweight libraries like React Router or Vue Router. In a project like ours with relatively simple navigation needs, Angular may be more than necessary for frontend routing.

While Angular provides a robust and scalable framework, it has a steeper learning curve compared to React and Vue. Our team considered Angular due to its built-in routing, state management, and form validation features, but ultimately decided against it because of its complexity and higher development time. Given our project's need for quick iterations and real-time updates, we prioritized a framework that allows for faster prototyping and flexibility.

3.6.1.3 React

React is a JavaScript library for building interactive user interfaces, developed by Facebook. Unlike Angular, React follows a component based architecture, which allows for reusable User Interface components that simplify development and maintenance. It also uses a virtual DOM, which improves rendering performance by updating only the necessary parts of the UI instead of re-rendering the entire page.

One of React's key strengths is its large community and extensive third-party libraries, making it an excellent choice for integrating real-time radar and AIS data. React also provides easy integration with backend APIs, allowing seamless communication between the frontend dashboard and the radar data processing system. React does not include routing by default, so developers typically use a separate library like React Router to manage navigation between views.

Given our team's familiarity with React and its ability to handle real-time updates and dynamic UIs, we decided to use React as our frontend framework. Additionally, React's ecosystem includes Next.js, which could be useful for server-side rendering and performance optimization in the future.

3.6.2 Backend

The backend is responsible for handling data processing, API logic, and communication with the database. It must be capable of managing radar data flow and responding to user

requests quickly and reliably. To guide our evaluation, we identified general metrics related to performance, simplicity, and how well each backend technology aligned with our team's technical background and project goals.

Table 3.20 *- Backend Technology Comparison*

	Django	**Nest**	**Express**
Learning Curve	Moderate – clear documentation but steep for those unfamiliar with Python	High – requires knowledge of TypeScript and Angular-style patterns	Low – very approachable for beginners familiar with JavaScript
Scalability	High – built in tools and structure make it well suited for large applications	High – modular architecture allows structured growth	Moderate – scalable, but requires more manual setup and structure
Flexibility	Moderate– opinionated and tightly integrated.	High – structured but allows flexibility via modules	Very High – opinionated and highly customizable

3.6.2.1 Django

Django is a Python-based web framework that follows the Model-View-Template pattern. It is known for its built-in security features, scalability, and rapid development capabilities. Django is widely used in data-driven applications due to its built-in support for database management, authentication, and API development.

However, Django is more monolithic compared to lightweight backend frameworks like Express.js and Nest.js. While Django provides excellent data handling capabilities, it is less flexible for integrating with Node.js-based frontend frameworks like React. Given our team's preference for a JavaScript-based full-stack approach, we opted for backend frameworks that are better suited for handling real-time data transmission and RESTful API development.

3.6.2.2 Nest

Nest.js is a TypeScript-based backend framework built on top of Express.js. It provides structured architecture, dependency injection, and built-in support for microservices. Nest.js was considered for this project due to its scalability and TypeScript support, making it ideal for large applications that require strict type safety.

While Nest.js offers better code organization and maintainability, it adds additional complexity compared to Express.js. Express is lighter and simpler. Given our need for a lightweight backend to handle API requests efficiently, we opted for Express.js instead of Nest.js.

3.6.2.3 Express

Express.js is a minimalist and flexible backend framework for Node.js. It is widely used for building RESTful APIs and handling real-time communication between the frontend and database. Express.js allows rapid development, easy integration with MongoDB, and scalability, making it an excellent choice for our project.

Our decision to use Express.js over Django and Nest.js was based on its lightweight architecture that allows for quick request handling. And its Seamless integration with React which maintains a full JavaScript stack for both frontend and backend. Given these advantages, Express.js was chosen as our primary backend framework.

3.6.2.4 AWS Lambda

AWS Lambda is a serverless compute service that lets you run code without provisioning or managing servers. Instead of having a continuously running backend, we can define specific Lambda functions that respond to events — such as a user login, a data query, or a real-time update.

Each function can be triggered by a request coming from API Gateway. This setup allows us to lower Costs, as you only pay for the compute time used. It also allows for improved scalability with the functions, as they scale automatically depending on demand and it simplifies deployment, each function can be deployed independently, speeding up development and testing.

3.6.3 Database

The database plays a crucial role in storing both structured AIS data and semi-structured radar detections. Choosing the right database solution requires balancing flexibility, speed, and ease of use. For our project, we considered how intuitive the database technology is, how easily it supports our data model, how actively the technology is maintained by its developer community, and how pricing would impact our ability to scale the solution over time.

Table 3.21 *- Database Technology Comparison*

	Redis	**PostgreSQL**	**MongoDB**
Ease of Use	Moderate - simple commands, but limited to key-value data	Moderate - requires understanding of SQL and schemas	High - flexible schema, easy with Mongoose for JavaScript developers
Documentation	High - concise, well organized, especially for caching use cases	Very High - mature and well documented	Very High-large community, great docs, and Mongoose guides

Data Flexibility	Low - stores only key-value pairs	Low - strict schemas, relational only	High - schema-less, supports structured and unstructured data
Real-Time Capability	Very High - in-memory and built for real-time data/pub-sub operations	Moderate - strong with triggers and pub/sub extensions	High - works well with real-time updates when paired with Web Sockets.

3.6.3.1 Redis

Redis (Remote Dictionary Server) is an in-memory data structure store often used for caching, real-time messaging, and session management. It offers high-speed data access compared to traditional databases and is ideal for applications requiring fast data retrieval.

For our project, Redis was considered as a caching mechanism to store frequently accessed radar and AIS data, reducing the load on the primary database. Additionally, Redis supports publish/subscribe (Pub/Sub) messaging, which can be useful for handling real-time alerts and notifications when a dark vessel is detected.

While Redis is not a primary database, it plays a crucial role in improving response time and system efficiency in maritime tracking applications.

3.6.3.2 PostgreSQL

PostgreSQL is a powerful relational database management system that supports SQL standards and offers advanced features like JSON support, spatial indexing, and ACID compliance. PostgreSQL remains a valuable option in cases that demand strict data consistency, complex queries, or long-term archival. Its schema-based structure makes it ideal for managing user accounts, authentication records, system logs, and historical vessel tracking. PostgreSQL excels at relational integrity, making it well-suited for executing JOIN operations, aggregations, and capabilities that could support future features such as proximity alerts and zone violations. However, compared to MongoDB, PostgreSQL is less flexible with dynamic or semi-structured data and is more challenging to scale horizontally. Despite these trade-offs, PostgreSQL could provide a robust and reliable option for expanding the system’s capabilities where relational consistency is critical.

3.6.3.3 MongoDB

MongoDB is a document-oriented database that stores data in a flexible, JSON-like format known as BSON (Binary JSON). Unlike relational databases that use structured tables, MongoDB’s schema-less nature allows it to efficiently store unstructured and semi-structured maritime data, such as radar detections, AIS vessel tracking, and environmental conditions. This flexibility is particularly useful for our system, as vessel and radar data can vary significantly in structure and size.

To streamline data modeling and interaction with MongoDB, we looked at Mongoose, a JavaScript Object Data Modeling (ODM) library. Mongoose provides a schema enforcement allowing us to define structured data models. A built-in query functions, simplifying data retrieval and manipulation. And middleware support, enabling validation, pre-processing, and event-driven operations within our application.

Given that radar and AIS data accumulate rapidly, MongoDB's scalability and high-performance indexing ensure that our system can efficiently handle large volumes of incoming data. Additionally, its ability to support real-time queries aligns with the requirements of our web dashboard, which must continuously update vessel and radar detection information. By integrating Mongoose with our Express.js backend, we can maintain a structured yet flexible approach to database management, ensuring efficient data storage, retrieval, and real-time processing for maritime domain awareness.

3.6.4 Server

The web server is the deployment environment for both frontend and backend components. It must efficiently handle requests, scale as needed and be easy to configure. In our comparison, we looked at ease of deployment, community support, compatibility with our stack, general reliability, and particularly as our system may scale or require additional performance resources over time.

***Table 3.22** - Server Technology Comparison*

	AWS	**DigitalOcean**	**Azure**	**Google Cloud**
Ease of Deployment	Moderate - Easier with Lightsail; AWS overall is complex	Very Easy - simple UI, fast setup	Moderate-user -friendly but better for .NET developers	Easy - modern dashboard, fast launch process
Stack Compatibility	High - strong Node.js and MongoDB support	High - supports modern stacks (Node.js, MongoDB)	Medium -optimized for Microsoft stack	High - good for JavaScript based integration
Scalability	Very High - EC2, Lambda, ECS support flexible scaling	High -manual vertical/horizontal scaling with Droplets	Very High - full enterprise-grade auto-scaling	High -built-in scaling via App Engine and GKE
Documentation	Very High -extensive resources and community	High -developer-focused guides and active support	High – especially for Microsoft developers	Moderate – clear docs but smaller ecosystem

3.6.4.1 AWS

Amazon Web Services is one of the most widely adopted and versatile cloud platforms, offering an extensive suite of infrastructure tools for deploying scalable web applications. While AWS EC2, Lambda, and ECS provide powerful options for production environments, our team is currently evaluating AWS Lightsail for its simplicity and affordability during the prototyping phase.

Lightsail offers a more beginner-friendly interface than the traditional AWS console and storage. It allows developers to quickly spin up pre-configured environments for Node.js and MongoDB without the overhead of configuring VPCs, IAM roles, or complex load balancing setups. This makes it ideal for early testing and small-scale deployments.

3.6.4.2 DigitalOcean

DigitalOcean is a cloud infrastructure provider designed for developers who want a simpler and more affordable alternative to enterprise-level cloud services like AWS and Azure. It offers virtual private servers called “Droplets” that can be easily deployed to host web applications, databases, and backend APIs. For this project, DigitalOcean represents a strong middle-ground option for deploying the Maritime Web Application due to its balance of control, performance, and cost-effectiveness.

DigitalOcean also supports scalable infrastructure, meaning the project can start with a single low-cost Droplet and scale vertically or horizontally as needed. It provides built-in monitoring, networking tools, firewalls, and storage solutions that make it suitable for small-to-medium production systems.

3.6.4.3 Azure

Microsoft Azure offers a robust cloud infrastructure and is especially effective for enterprises already using Microsoft tools. Azure’s App Services and Azure Functions provide similar capabilities to AWS Lambda and EC2. It supports a wide range of backend technologies and has strong integration with Visual Studio and GitHub. However, its ecosystem can be more suited to .NET-based applications, and Azure’s pricing and performance for small teams or startups may not be as competitive as AWS for our specific use case.

3.6.4.4 Google Cloud

Google Cloud Platform is a modern cloud provider known for its performance and tools related to big data and machine learning. It features services such as App Engine and Cloud Functions, which is similar to AWS and Azure. GCP also excels in simplicity of setup. However, GCP’s community and third-party tool integration are less mature compared to AWS. GCP could serve as a viable alternative.

3.6.4.5 Network-Attached Storage

During project discussions, our sponsor highlighted the importance of a Network-Attached Storage (NAS) system as a potential solution for managing large-scale

maritime data efficiently. Given the high volume of radar detections, AIS overlays, and vessel tracking logs, the need for a centralized and scalable storage solution became apparent. Our sponsor emphasized that NAS could provide better data accessibility, reliability, and security, ensuring that both onboard systems and shore-based operators could retrieve and analyze data without disruptions.

NAS is a dedicated file storage system that allows multiple devices to access and manage data over a network. Unlike traditional hard drives that are attached to a single machine, NAS functions as an independent storage unit with its own processing power and network-sharing capabilities. In this project, NAS would serve as a central repository for maritime tracking data, ensuring that data from onboard Raspberry Pi and Intel NUC devices can be securely transferred and accessed by the shore-based control station. Since NAS can be accessed remotely, it enhances ship-to-shore communication by allowing real-time data synchronization, ensuring that maritime surveillance and monitoring remain continuous and efficient.

Scalability is another major consideration. Since NAS supports high capacity storage and allows for storage expansion, it can accommodate growing data needs over time. This is particularly valuable for long-term historical storage, where vessel tracking patterns and radar detection logs may need to be stored for extended periods for analysis.

Additionally, NAS systems support Redundant Array of Independent Disks (RAID), which helps protect against data loss due to hardware failure. This ensures that even if one drive fails, data remains intact and accessible. NAS Also provides high speed read/write operations, which is essential for real-time data access and processing. This is especially beneficial for applications such as ours.

3.6.5 Internal API's

Our internal APIs handle communication between the frontend dashboard, backend services, and real-time data sources. REST and WebSockets provide different methods for retrieving and updating data efficiently.

3.6.5.1 REST API

Representational State Transfer is the primary method for standardized data retrieval and CRUD operations. It provides structured endpoints that allow the frontend to fetch, update, or delete vessel and radar data. For example, a GET request to /api/vessels might retrieve all known vessels from AIS data, while a POST request to /api/radar-data might upload new radar detections. REST is widely used due to its scalability, predictability, and ability to integrate with caching mechanisms, making it an ideal choice for handling the structured communication between the frontend and backend.

3.6.5.2 WebSockets

WebSockets provide a persistent, two-way communication channel between the client and server. Making them ideal for real-time data updates. Unlike REST APIs, which require the client to make repeated requests to receive new data, WebSockets allow the

server to push new information to the frontend as soon as it becomes available. This is useful in the context of our maritime dashboard, where radar detections or dark vessel alerts may need to be displayed instantly without waiting for manual refreshes.

3.6.5.3 AWS API Gateway

Amazon API Gateway is a fully managed service that enables developers to create, publish, maintain, monitor, and secure APIs at any scale. It acts as a front door for our backend services, allowing HTTP requests from the frontend to be routed efficiently to their appropriate destinations.

In our case, the API Gateway would serve as the connector between the React frontend and backend Lambda functions. It will manage endpoints for user authentication, data submission, and retrieval operations. Benefits include Centralized Routing: Easy control over request handling and endpoint management. Rate Limiting and Throttling: Protection against API misuse or DDoS attacks. CORS Configuration: Seamless cross-origin communication between frontend and backend.

3.6.6 External API's

Our project integrates third-party APIs to provide mapping functionalities and radar data extraction.

3.6.6.1 Leaflet.js

Leaflet.js is a lightweight JavaScript library that allows us to display interactive maps within the web dashboard. It supports real-time plotting of vessels and radar detections, enabling maritime operators to visualize ship movements efficiently. By integrating geospatial overlays and customizable markers, Leaflet enhances our ability to identify dark vessels within a given area. The library's flexibility and ease of use make it a suitable choice for this application.

3.6.6.2 Raymarine SDK

The Raymarine SDK provides a bridge between our software and the Raymarine radar system, allowing for real-time radar data extraction and processing. By interfacing with this SDK, we can analyze radar signatures, classify vessels, and trigger alerts for unidentified objects. Integration with our Node.js backend ensures that radar data is efficiently stored and transmitted to the dashboard.

3.6.7 Authentication Methods for Maritime Web App

Authentication is a critical aspect of any secure system, ensuring that only authorized users can access sensitive data and functionalities. Various authentication methods can be implemented based on security requirements, ease of use, and system architecture. This section explores four primary authentication techniques, including password hashing, JSON Web Tokens (JWT), OAuth 2.0, and Multi-Factor Authentication (MFA), and evaluates their advantages and trade-offs.

***Table 3.23** - Authentication Technology Comparison*

	AWS Cognito	Firebase	Auth0
Ease of Integration	Moderate (best with AWS stack)	Very easy (especially for mobile/web apps)	Easy, with wide SDK support and simple config
Security Features	Built-in MFA, token rotation, OAuth	Built-in MFA, OAuth	Built-in MFA, role management, extensive protocol support
Flexibility	Moderate-AWS-centric	High - backed by Google infra	Very high; platform-agnostic with customizable rules and flows

3.6.7.1 Password Hashing

Password hashing is a traditional yet effective authentication method in which user passwords are processed through a cryptographic algorithm before being stored in a database. Instead of saving plain-text passwords, the system applies a hashing function which converts the password into a fixed-length cryptographic string. Additionally, a unique value is added to each password before hashing to prevent attacks like rainbow table exploits.

The primary advantage of password hashing is that it is self-hosted, meaning it does not require reliance on external authentication services. It is also relatively simple to implement, making it an accessible option for systems that need basic authentication security. However, password hashing alone does not prevent brute-force attacks if an attacker gains access to the database. If an account recovery system is not implemented, users may also struggle with forgotten passwords, leading to potential access issues.

3.6.7.2 JSON Web Token

JWT authentication is a stateless authentication method that allows users to log in once and receive a cryptographic token, which is then sent with each subsequent request. When a user logs in, the server generates a JWT token, typically containing a user ID, expiration time, and permissions, and sends it back to the client. The token is then stored in the user's browser (either in local storage or HTTP-only cookies) and included in each request to authenticate the user without requiring database lookups.

One of the biggest benefits of JWT authentication is its scalability. Since authentication data is stored in the token itself, the backend server does not need to maintain a session state, reducing server load. JWTs are especially useful in API based systems and distributed architectures where multiple services need to verify a user’s identity. However, one security concern with JWTs is token storage. If stored in local storage, an attacker could steal the token using Cross Site Scripting attacks.

3.6.7.3 OAuth 2.0

OAuth 2.0 is an authentication framework that allows users to log in using third-party accounts such as Google, Facebook, or GitHub, eliminating the need for the system to manage passwords. When a user chooses to log in through a third-party provider, they are redirected to the provider's authentication system. If authentication is successful, the provider issues a secure access token, which is sent back to the web application to verify the user's identity.OAuth provides a fast and secure login experience. It is particularly beneficial in systems where user account creation is frequent, as it eliminates the need for handling password security.

However, OAuth authentication does introduce external dependencies. If a third-party provider experiences downtime or restricts access, users may be locked out of the system. Additionally, privacy concerns arise as some third-party services track user logins across multiple applications.

For systems requiring seamless authentication with minimal password management, OAuth 2.0 is an excellent choice, especially when integrated with providers like Google. However, for security-sensitive applications, it may be preferable to use a self-hosted authentication system that does not rely on external identity providers.

3.6.7.3 Multi-Factor Authentication

Multi-Factor Authentication (MFA) improves account security by requiring users to verify their identity using more than one form of authentication. Rather than relying solely on a password, MFA typically combines two or more factors: something the user knows (like a password), something the user has (such as a one-time code sent via SMS, email, or an authenticator app), and something the user is (such as a fingerprint or facial recognition). When MFA is enabled, users first enter their standard credentials and are then prompted to provide a second factor, such as a time-based one-time password (TOTP) generated by an app like Google Authenticator or a code sent via email or SMS. This additional verification layer significantly reduces the risk of unauthorized access, even if a user's password is compromised through phishing or data breaches. While MFA greatly strengthens security, it can introduce usability concerns if users lose access to their second factor, such as losing their phone or changing numbers. For higher security applications, TOTP-based methods are generally preferred over SMS-based verification, as the latter can be vulnerable to SIM-swapping attacks.

3.6.7.4 AWS Cognito

AWS Cognito is a cloud-based identity management service that provides user sign-up, sign-in, and access control. It helps developers manage authentication and authorization without manually handling password storage, hashing, or session tracking. Cognito supports standard authentication protocols like OAuth 2.0 and OpenID Connect, and it integrates seamlessly with other AWS services such as API Gateway and Lambda.

Cognito uses user pools to manage user accounts and credentials, and identity pools to assign temporary AWS credentials for accessing services. It also supports multi-factor

authentication (MFA), account recovery, and password policies out of the box, improving both security and user experience.

3.6.7.4.1 Comparison to Traditional Methods

Compared to the traditional authentication methods described earlier (password hashing, JWT, OAuth 2.0, and MFA) AWS Cognito presents a comprehensive and managed alternative that consolidates many of their features. Unlike password hashing, where developers are responsible for implementing secure algorithms and managing password storage, Cognito automatically hashes and stores passwords securely, removing the risk of insecure handling. Cognito also leverages JWTs internally, issuing tokens upon successful authentication to allow for stateless and scalable access control, similar to custom JWT implementations but without the overhead of manual token management.

When compared to OAuth 2.0, Cognito extends similar functionality by supporting social logins through providers like Google, Facebook, and Apple. This allows the system to offer third-party authentication while still centralizing user management within Cognito itself. Furthermore, Cognito includes built-in support for MFA, such as time-based one-time passwords or SMS-based verification, aligning with best practices for systems that require an extra layer of security.

Essentially, Cognito integrates these diverse authentication strategies into a single, cloud-managed solution. This reduces development complexity and improves security while maintaining compatibility with modern authentication standards and user experience expectations.

3.6.7.4.2 Advantages of Using Cognito

AWS Cognito offers several distinct advantages that make it well-suited for our application's authentication needs. Its primary benefit is enhanced security, as it abstracts away the need to directly manage user credentials. This reduces our system's vulnerability to attacks that target improperly stored passwords or misconfigured authentication logic. Furthermore, Cognito provides time-saving tools by handling user sign up flows, email and SMS verification, and password policies without requiring custom implementation. It is also highly scalable, capable of supporting thousands of users while integrating directly with other AWS services like API Gateway and Lambda, creating a unified cloud infrastructure.

By adopting AWS Cognito, we can focus more on core development tasks while offloading user authentication to a secure and reliable system. It provides a best of both worlds approach by combining self-hosted flexibility with cloud-managed convenience.

3.6.7.5 Firebase

Firebase supports a wide variety of authentication methods, including email/password, phone number, and federated identity providers like Google, Facebook, Twitter, GitHub, and Apple. It simplifies the process of implementing secure login systems by providing SDKs for platforms such as web, Android, and iOS, and handling complexities like token

management, account linking, and password recovery. While AWS Cognito is our current primary choice for user authentication, the team also considered Firebase Authentication, a powerful alternative provided by Google. Firebase Authentication offers backend services to help developers authenticate users securely and easily, using only client-side code.

Firebase Authentication integrates seamlessly with other Firebase services like Firestore and Realtime Database but can also be used with custom backends, including those built with AWS services. Its use of JWTs for access tokens aligns well with modern authentication standards, allowing easy verification and session persistence on both the client and server.

Compared to AWS Cognito, Firebase Authentication is generally considered more user-friendly for rapid development and prototyping, particularly in mobile-heavy or real-time applications. It features built-in UI components, client SDKs, and easy integration with frontend frameworks like React. However, Firebase has limitations in advanced user pool customization and integration with IAM roles, which Cognito handles more robustly within the AWS ecosystem.

Firebase Authentication also supports multi-factor authentication for enhanced security, and login flows can be further extended using Firebase Cloud Functions. For teams already integrated into the Google Cloud Platform or requiring fast frontend integration, Firebase can offer a smoother experience.

While we are currently proceeding with AWS Cognito due to its tight integration with the rest of our AWS infrastructure, Firebase Authentication remains a viable alternative depending on system goals and team familiarity.

3.6.7.6 Auth0

Auth0 is a third-party identity-as-a-service platform that provides authentication and authorization services for modern applications. It is designed to handle complex identity needs for web, mobile, and API-based systems. Auth0 supports a wide range of authentication protocols, including OAuth 2.0, OpenID Connect, and SAML, allowing developers to implement secure login and authorization flows with minimal effort.

Auth0 stands out for its flexibility and developer experience. It offers an intuitive dashboard for managing user roles, permissions, and identity providers, along with customizable login pages and built-in support for multi-factor authentication. Social login through providers like Google, Facebook, and GitHub is easily enabled with minimal configuration. Auth0 also provides APIs and SDKs for popular frameworks like React, Angular, and Node.js, making integration seamless across tech stacks.

Unlike Firebase or Cognito, which are more tightly coupled with their respective cloud ecosystems, Auth0 is a platform agnostic solution. It can be integrated into any cloud-hosted or on-premise application and is often used by enterprises that require advanced identity governance features. Its generous free tier supports up to 7,000 active

users per month, making it an accessible option for startups and scalable enough for large-scale production systems.

Despite its flexibility and extensive features, Auth0 can become expensive for larger teams or high user volumes. Additionally, while it is powerful, it introduces a dependency on a third-party vendor for a core piece of application infrastructure. For applications where fine-grained control and customization are critical, or where integration into an existing identity system is required, Auth0 presents a compelling option.

3.7 Communication Protocols

3.7.1 NMEA 2000 and Marine Communication Protocols

A communication protocol, in technology, is a way for information and data to flow between different devices. When talking about communication protocols we would be using for our project, it is crucial to focus on standards built for and relevant to current marine technology. Considering the different elements of our maritime domain awareness system, we are looking for a standardized way to be able to successfully make communication smooth and flawless. These days, vessels increasingly rely on interconnected systems for navigation, monitoring, and situational awareness, which makes it essential to ensure seamless data exchange.

To develop our maritime domain awareness system, multiple electronic subsystems must communicate efficiently for uninterrupted collection, processing, and transmission of radar data. One of the primary components involved in this project is the Raymarine Cyclone radar system, which generates real-time surface vessel detections, and a GPS receiver to provide precise geolocation data to correlate with radar outputs. The system will also connect with Starlink to transmit processed information to a remote monitoring station to provide nearly constant real-time situational awareness. Each of these components operates using specific communication protocols: the radar and GPS systems are compatible with marine-specific standards such as NMEA 2000, while satellite communication relies on network protocols for long-range data transmission.We are looking to construct a unified framework of these different data streams to allow for efficient data exchange between the onboard processor, sensors, and external networks. Given that NMEA 2000 is the industry standard for modern marine electronics, we must understand its architecture and capabilities to successfully implement into our system.

To go into more detail about how the information is transferred, we will need to explain how multiple electronic devices communicate over a single, unified network. The National Marine Electronics Association (NMEA) has developed multiple protocols over the years to address these needs, with NMEA 2000 emerging as the modern industry standard for connecting marine sensors, displays, and control units. NMEA 2000 is designed to allow different marine electronics to share data using a standardized messaging framework. That way, there is no need for multiple independent wiring systems and overall will improve the efficiency and interoperability of marine networks.

NMEA 2000 is the upgraded version of NMEA 0183, an older protocol that operated on a point-to-point serial communication model. Now, the communication model is built on

the Controller Area Network (CAN) protocol, which allows multiple devices to communicate over a shared data bus instead of requiring a direct connection between each individual device. This means that all devices on an NMEA 2000 network can "talk" and "listen" to each other simultaneously, making it much more flexible and scalable than the previous version. The CAN protocol is used a lot in automotive, industrial, and aerospace applications so we know its a well-supported foundation for modern marine electronics.

The current NMEA 2000 version offers several major improvements like increased data speed and reliability. Currently, the protocol operates at a data rate of 250 kbps which is a big jump from NMEA 0183's 4,800 bps standard rate. This higher bandwidth supports real-time data exchange which should prove to be useful for our monitoring systems. Another important feature of NMEA 2000 is its network topology, which follows a bus structure with a central backbone cable. Devices connect to this backbone using drop cables and T-connectors and makes installation much more simple because it reduces the amount of cabling needed. Each end of the network has terminating resistors to prevent signal reflections and maintain a stable connection.

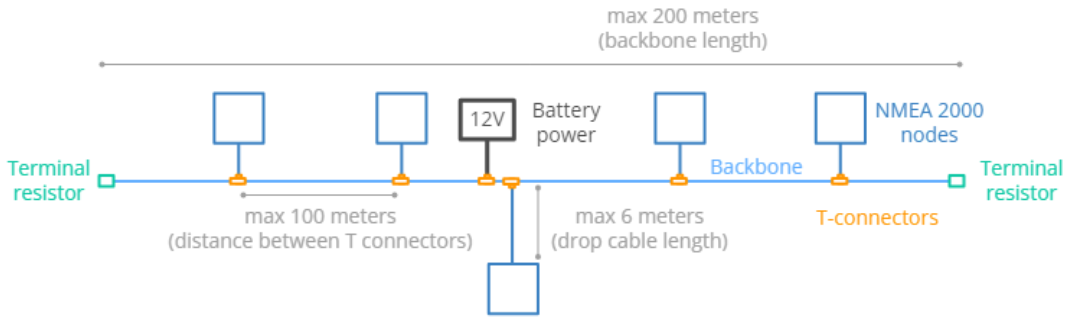


Figure 3.4 *- NMEA 2000 Network Topology*

To fully understand how NMEA 2000 communication works with marine electronics, it is important to examine how data is structured and transmitted across the network. Each NMEA 2000 message follows a standardized binary format based on the CAN protocol. In the past, NMEA 0183 used human-readable ASCII sentences, but now NMEA 2000's binary format helps make it significantly faster and more robust for real-time data transmission.

A single NMEA 2000 message consists of multiple key fields:

Table 3.24 *- Key Fields of NMEA 2000 Message*

Key Field	Size (bits)	Description
Priority	3	Determines urgency of message
Parameter Group Number (PGN)	18	Identifies the type of data being transmitted (e.g., vessel speed, GPS position, or engine temperature)
Source Address	8	Specifies the sending device
Data Fields	0 - 64	Contains the actual sensor or control data being transmitted.

Figure 3.5 *- NMEA 2000 message (PGN and Data Fields)*

The previous diagram explains the general structure of a message, but actual raw NMEA 2000 data appears as hexadecimal CAN bus frames when logged. The following figure is an example of real AIS (Automatic Identification System) messages captured on an NMEA 2000 network:

Timestamp (Epoch)	CAN ID	PGN (HEX)	PGN (DEC)	DataBytes
1620692704.079450	11F80F00	1F80F	129039	001B127CEAD5123D
1620692704.080000	11F80F00	1F80F	129039	0131F3D0ACF2231A
1620692704.080600	11F80F00	1F80F	129039	0203FFFF00000000
1620692704.081200	11F80F00	1F80F	129039	0320FFFF0070FEFF
...				
1620692705.079500	11F80F00	1F80F	129039	201B127CEAD5123D
1620692705.080050	11F80F00	1F80F	129039	2131F3D0ACF2231A
1620692705.080650	11F80F00	1F80F	129039	2207FFFF00000000
1620692705.081200	11F80F00	1F80F	129039	2320FFFF0070FEFF
1620692704.079450	11F80F00	1F80F	129039	127CEAD5123D ... 20FFFF0070FEFF
1620692705.079500	11F80F00	1F80F	129039	127CEAD5123D ... 20FFFF0070FEFF

legend
NMEA 2000 PGN (0x1F80F = AIS Class B Position Report)
#data bytes (0x1B=27)
sequence counter
frame counter
payload data (27 bytes)

Figure 3.6 *- Real-world NMEA 2000 Data (Fast Packet Trace example)*

In this example data log:

- The PGN value (0x1F80F) represents an AIS Class B Position Report.
- The CAN ID and PGN (HEX & DEC) fields help identify the type of message.
- The Data Bytes field contains encoded information related to vessel position, heading, speed, and other parameters.

Since NMEA 2000 messages use a binary CAN bus format, they require specialized software to decode into readable values. This is particularly important for our project because we will need to extract radar and GPS data from NMEA 2000 messages before processing it for vessel tracking.

For our project, we will hone in on integrating NMEA 2000 into our radar system and GPS receiver to allow the data to be efficiently transmitted and processed by the onboard Raspberry Pi. It is also important to take into account the harsh marine environments, so we can find the best way to implement and optimize real-time data sharing. Now we should have a reliable foundation for collecting and transmitting radar data before it is processed and sent through Starlink for remote monitoring.

3.7.2 Starlink Communication

One of the core features of this project is to be able to send information that our systems gather from anywhere on the planet to a remote monitoring station. The best COTS product designed for this exact task is Starlink, a low Earth orbit (LEO) satellite communication system that enables reliable and high-speed internet access from virtually anywhere. Since our system is designed for maritime surveillance, we need quick and low-latency communication that maintains this level of performance under any circumstances, especially when transmitting real-time data from remote locations. Other solutions that provide marine internet connection are usually slow, high in cost, and have coverage gaps. However, Starlink is a game-changing alternative that uses a dense network of satellites to provide reliable, high-bandwidth connectivity even in the middle of the ocean.

For our implementation, we are using a Starlink Mini terminal setup with unlimited Roam Access to provide in-motion, ocean use. This compact model works well in our plug-and-play system, while maintaining high-speed performance. Incorporating Starlink into our system allows us to efficiently transmit real-time radar and AIS data, video streams from the gimbal-mounted camera, and other vessel information to a remote monitoring station. Without a system like this, it would be nearly impossible or very expensive to have constant, high-speed communication from every corner of the ocean.

Outside of just providing an internet connection, Starlink also enables consistent data flow between our sensor network, the onboard processing unit, and the command center. This connectivity will let us relay alerts, transmit video feeds, and start or stop the process when needed. The Starlink Mini relatively consumes low power and is easy to set up right out of the box making it ideal for deployment on a self-harvesting power system.

Traditional satellite communications rely on high-orbit geostationary satellites, but Starlink actually orbits at around 550 km. This helps achieve much lower latency (20-40 ms) compared to conventional satellite internet that can be over 600 ms.

- Download Speed: Up to 100 Mbps, more than enough to handle radar data, AIS logs, and even video streams.
- Upload Speed: Around 10 Mbps, sufficient for transmitting alerts and tracking data.
- Latency: A low 20-40 ms, which keeps data transfer quick and responsive.
- Power Consumption: 20-40W, a key factor when designing our system's power supply.
- Connectivity: Supports Ethernet and Wi-Fi, allowing us to interface directly with our onboard Raspberry Pi.

This combination of low latency, fast speeds, and global coverage is exactly why we selected Starlink Mini over other satellite communication systems. Whether we're detecting dark ships in the middle of the ocean or transmitting alerts to a command center, Starlink gets data where it needs to go, instantly and securely.

There are other satellite internet options for maritime applications, but Starlink was the clear choice for our project. Here's how it stacks up:

Table 3.25 *- Satellite Comparison*

Feature	**Starlink Mini**	**Traditional Satellite Internet**
Latency	20-40 ms	600+ ms
Download Speed	Up to 100 Mbps	1-10 Mbps
Upload Speed	10 Mbps	<1 Mbps
Power Usage	20-40W	50W+
Coverage	Global	Limited (depends on provider)

Older satellite internet systems simply aren't fast enough for the real-time requirements of our project. The low-latency, high-speed, and low-power nature of Starlink makes it the best choice for transmitting our maritime surveillance data.

Now that we know that Starlink gives us a major edge in remote vessel tracking, it is also important to take into account some technical hurdles when implementing it into our system.

Because Starlink operates using a mesh of constantly moving satellites, our connection will occasionally switch between satellites overhead. This transition can cause brief latency spikes or packet loss, which could temporarily disrupt data transfer. To

workaround this, we need to optimize our transmission protocol. In case there are any critical alerts, they will always be prioritized and retransmitted if necessary.

LEO satellite internet is generally more resilient to bad weather than traditional satellite services, but it's not completely immune. Heavy rain, thick cloud cover, or extreme storms can cause signal degradation. While this is unlikely to cause a total connection loss, it's something we need to keep in mind when transmitting large data packets, like full-resolution video from the gimbal camera.

The 20-40W power draw of the Starlink Mini means we need to plan for efficient power management. Our system will likely rely on a combination of wind energy and battery storage, so we must ensure Starlink doesn't drain power when not in use. One way to handle this is dynamic power scheduling—Starlink could turn on only when necessary, such as when a radar contact is detected and needs to be reported.

By integrating Starlink Mini into our system, we ensure fast, secure, and reliable communication for our radar, AIS, and gimbal tracking data. Without it, monitoring dark ships in real time would be impossible. While weather, latency spikes, and power management present some challenges, they can be mitigated with smart software optimizations.

3.7.3 Wireless Communication

Wireless communication complements the wired interfaces in MDAS by enabling remote access and system flexibility, particularly during diagnostics, deployment, and software updates. The Raspberry Pi 4 includes integrated 2.4/5 GHz Wi-Fi and Bluetooth, allowing the system to wirelessly interface with external devices, especially during the prototyping or maintenance phase.

3.7.3.1 Local Network Remote Access

During system development and post-deployment monitoring, remote access to the Raspberry Pi can be established through several wireless methods over a local area network (LAN). These include:

- **SSH (Secure Shell)** for secure terminal access, allowing command-line operations such as log inspection, service restarts, and debugging.
- **VNC (Virtual Network Computing)** for accessing the Raspberry Pi's GUI over the network.
- **Raspberry Pi Connect**, a newer utility that facilitates encrypted desktop sharing and remote file access without needing the Pi's IP address.

These tools allow users to control the unit from nearby vessels or base stations without needing a direct USB or Ethernet connection. This will be useful for software updates or debugging while the unit is mounted on a ship or buoy.

3.7.3.2 Starlink Uplink for Cloud-Based Commands

Beyond local wireless access, the embedded system relies on the Starlink Mini terminal to send and receive data wirelessly over the internet. While the physical connection to Starlink is wired for reliability, the data is transmitted wirelessly via the Starlink satellite mesh back to ground stations and then routed to the MDAS web server. This enables:

- Remote radar activation/deactivation
- Real-time data uploads and alerts
- Remote image capture commands for the gimbal-mounted optical system

In the field, this capability allows ground operators to monitor units deployed hundreds of miles offshore without physical access to the system.

3.7.3.3 Gimbal Camera and Bluetooth Control

The gimbal-mounted camera system can also be configured to receive control signals over wireless protocols such as Bluetooth or Wi-Fi. Although primary control is wired for reliability, Bluetooth Low Energy may be used as a fallback or auxiliary method during setup or system calibration. This reduces the need to access the unit's interior and allows us to test camera positioning and lens adjustment remotely from mobile devices.

3.7.3.4 Environmental Considerations

Wireless signals are known to degrade in the presence of water vapor, metal enclosures, and electromagnetic interference, which are all common in marine environments. To counteract this, all antennas (for Wi-Fi or Bluetooth) are mounted externally on the enclosure with waterproof seals and directional orientation to maximize range and signal strength. For critical communications, wireless methods are never used in isolation and are always backed up by wired channels.

Table 3.26 *- Environmental Considerations*

Technology	Purpose	Connected Devices / Use Case
Wi-Fi	Remote diagnostics and updates over LAN	SSH, VNC, Pi Connect from local laptop or phone
Bluetooth / BLE	Local wireless control and setup	Optional gimbal calibration, mobile setup tools
Starlink Satellite	Global remote access, real-time data transmission	Base station command center via satellite internet

3.8.3 Wired Communication

Wired communication plays a crucial role in the onboard operation of the Maritime Domain Awareness System (MDAS), where multiple hardware components must

exchange data reliably under harsh marine conditions. Within the embedded system, the Raspberry Pi 4 Model B acts as the central hub, interfacing with various devices using its built-in USB ports, Ethernet jack, and 40-pin GPIO header. Selecting the correct physical communication interfaces for each component ensures low-latency, noise-resistant connections critical for real-time maritime surveillance.

3.8.3.1 Radar System Integration

The primary data source for vessel detection in MDAS is the Raymarine Cyclone Radar, which communicates using Raymarine's proprietary SeaTalkng/NMEA 2000 standard. Physically, this communication is routed through a radar cable that is connected to an Ethernet-compatible radar interface box (such as a RayNet to RJ45 bridge), allowing Ethernet-based communication with the Raspberry Pi. Using this wired Ethernet connection ensures the high bandwidth and consistent connection necessary to transfer radar data to the onboard processor. Since NMEA 2000 utilizes CAN (Controller Area Network) as its physical layer, it also benefits from error detection and high reliability in electrically noisy environments.

3.8.3.2 Starlink Mini Connectivity

Data from the Raspberry Pi must be transmitted to the remote base station over a satellite link. This is achieved by interfacing with the Starlink Mini terminal, which includes both Wi-Fi and an optional Ethernet port via a breakout adapter. For reliability, the system utilizes a wired Ethernet connection to the Starlink power supply, ensuring a stable uplink from the embedded system to the satellite network. This connection is critical for sending real-time alerts and radar overlays, especially when operating in regions with poor weather or electromagnetic interference.

3.8.3.3 Serial Peripheral Interfaces

Serial communication plays a foundational role in embedded systems and microcontroller-based designs, enabling various hardware components to exchange information using minimal wiring. Unlike parallel communication, where multiple data lines transmit several bits simultaneously, serial communication sends data one bit at a time over one or more channels. This method reduces the complexity of physical wiring and minimizes signal interference, especially beneficial in a tight space or noisy environments like maritime systems.

Given the modular nature of the Maritime Domain Awareness System (MDAS), selecting the right communication protocol is essential for connecting microcontrollers, sensors, and peripheral modules reliably and efficiently. The three most commonly used serial protocols in embedded systems are I2C (Inter-Integrated Circuit), UART (Universal Asynchronous Receiver-Transmitter), and SPI (Serial Peripheral Interface). Each has distinct advantages, trade-offs, and ideal use cases. Choosing among them requires evaluating parameters such as data transfer rate, wiring complexity, scalability, error-checking mechanisms, and power consumption

3.8.3.4 I2C

I2C was designed for communication over short distances within a single device or between components on a circuit board. I2C uses only two bidirectional lines which include SDA (Serial Data) and SCL (Serial Clock); they are shared across all devices on the bus. Each device on an I2C bus is identified by a unique 7-bit or 10-bit address allowing communication between a single master to multiple slave devices. The master initiates the communication, which sends out a start condition followed by the target device address and the desired read/write command. Data is transmitted in 8-bit packets with an acknowledgment (ACK) signal sent after each byte, ensuring basic error detection. I2C supports data rates up to 100 kbps in normal use, 400 kbps in Fast-mode and up to 3.4 Mbps in High-Speed mode.

In the context of the MDAS, using I2C can be beneficial when managing multiple onboard sensors and monitoring components. Since it allows multiple devices to share the same two data lines, I2C simplifies wiring and reduces the number of required connections. This helps in minimizing the amount of space that the electronic enclosure would need. It also has a low power draw which aligns with the MDAS goal of energy efficiency and minimal maintenance in remote environments.

However, despite its benefits, I2C has many limitations. Its relatively slow data transfer rate and susceptibility to signal degradation over long distances make it ill-suited for high-speed or long-range communication tasks. Additionally, because all devices on the bus share the same data and clock lines, a fault in one device can prevent communication across the entire bus.

3.8.3.5 UART

UART, or Universal Asynchronous Receiver-Transmitter, is a widely used asynchronous serial communication protocol designed for point-to-point data transmission. Unlike synchronous communication methods, UART does not rely on a shared clock line between devices. Instead, it uses two dedicated lines which include TX (Transmit) and RX (Receive); to send and receive data independently. Communication begins with a start bit, followed by a sequence of data bits, an optional parity bit, and one or more stop bits. The absence of a shared clock makes UART relatively simple to implement, requiring that both devices agree in advance on a shared baud rate to properly interpret data timing. Common UART baud rates include 9600, 19200, 38400, and 115200 bps, though higher speeds are possible with modern hardware.

In the context of the MDAS, UART is especially useful for establishing reliable communication between essential system modules where full-duplex data flow is beneficial. UART's direct, point-to-point design means there's no contention on the communication lines, reducing the risk of data collision or interference from other peripherals. This makes UART ideal for linking communication modules such as GPS receivers, long-range wireless transceivers, or onboard controllers responsible for logging and diagnostics. Its simplicity and low overhead make it a practical choice for transmitting short bursts of data or maintaining a continuous low-bandwidth link with peripheral devices, all while requiring minimal processing power.

However, UART does come with certain constraints. It supports only two devices per channel, one transmitter and one receiver, limiting its scalability in systems requiring communication with multiple peripherals unless multiple UART interfaces are available. The lack of a built-in error detection to identify if certain bits of data are correctly transmitted is a disadvantage. UART relies on framing and parity bits for data integrity and any other data handling must be implemented through software. In the MDAS, where data accuracy and communication reliability is crucial for vessel detection and system control, clear and effective software error handling must be implemented to ensure successful operation.

3.8.3.6 SPI

SPI is a high-speed, synchronous communication protocol used for short-distance data exchange between a master device and one or more slave devices. SPI is capable of full-duplex communication, meaning data can be sent and received simultaneously, which increases its overall efficiency. Because it does not require addressing or acknowledgment bits, SPI has very low protocol overhead. This allows it to achieve much higher data transfer rates up to or more than 10 Mbps, making it ideal for high-throughput applications.

In the context of the MDAS, SPI is especially useful when communicating with components that require rapid and frequent data transfer. For example, the radar system may output raw signal data at a high sampling rate, and storing or analyzing this data in real time requires a communication interface that can keep up without introducing delays or data loss. Using SPI's speed and timing make it a great choice for this system.

The SPI's limitations include the lack of scalability due to its requirement of separate SS lines from the master, so if we needed more SPI devices, it requires the use of a lot of the limited GPIO pins. It also has the same lack of built-in checking as the UART protocol and requires software to handle errors.

3.8.4 Serial Communication Usage

Due to the limited number of hardware serial interfaces on the Raspberry Pi, we are using logic level converters and port expanders where necessary to maintain isolation and avoid pin conflicts. Proper shielding and waterproofing of these wired connections is provided by a weatherproof case to ensure consistent operation in corrosive saltwater environments.

Table 3.27 *- Serial Communication Usage*

Interface	Protocol / Standard	Purpose	Connected Component(s)
Ethernet	TCP/IP, NMEA 2000 (via RayNet)	High-speed data transfer from radar system	Raymarine Cyclone Radar to Raspberry Pi

USB	USB 2.0 / 3.0	Connection to Starlink Mini and external modules	Starlink Mini Ethernet Adapter, USB storage
GPIO	Custom / Digital I/O	General-purpose control and sensor monitoring	Power relays, LED indicators, camera triggers
UART	Serial (RS232/TTL)	Low-latency serial communication	Gimbal controller, GPS module
I2C	I2C	Communication with low-speed sensors	Battery monitor, temperature sensor
SPI	SPI	High-speed sensor or peripheral communication	ADCs, custom hardware modules

3.9 Analysis of the Power System

The maritime device that is being built has many parts and systems that need to be powered in order to function. The stipulation for the project, however, indicates that the device needs to be self-sufficient. To be more specific, the device should not use power from the boat or structure that it is placed on, and as a result, the problem of a power source is critical to the functionality of the maritime radar.

To solve the issues of self-sufficiency pertaining to power, many options were considered, the primary being gas, solar, and wind power. Some sort of generator was the first option that was proposed. The pros of a gas or diesel generator is the sheer amount of power that would be produced and there would be no concern for the maritime device being underpowered. A generator would be a great way to power the radar system, but it does lack the total self-sufficiency that is required for the project. When installing the maritime device on a vessel, the goal is to have a hands-off system for the crew of the ship. As a result, if the crew had to refuel a fossil fuel generator on behalf of the maritime system, the total self-sufficiency of the power system would be lost.

With self-sufficiency being the goal of the power system, non-renewable sources like fossil fuels were easily scraped, leading to solar and wind being considered to power the maritime radar system. When considering renewable resources, one thing that must be acknowledged is the intermittent nature of such power sources. Solar can only be an option when the sun is out, and wind is determined by the weather. These renewable resources need to be analyzed to see if their reliability could potentially cause issues with the maritime radar system as well as if the power systems are adequate to even power the components of the system sufficiently.

To continuously power the radar system, the implementation of both solar and wind energy was considered with the implementation of a solar array along with a maritime wind turbine. This solution was aimed at using 2 power sources to cover the weaknesses of both forms of power. To have continuous operation of the radar system, one renewable

resource is not sufficient due to its variable reliability throughout the day. The combination of wind and solar allows for the creation of a more robust solution that would be more reliable, as the likelihood of both wind and solar energy collection being not sufficient to power the system is unlikely, and if that did come to pass, the inclusion of a backup battery would sustain the system until energy collection increased.

There are advantages to both solar and wind power but also a few negative things to consider. Having both sources of power complicates the system, requiring additional wiring, charge controllers, and brackets to mount things. The increased complexity makes the initial price higher and provides more opportunities for things to malfunction, so maintenance becomes even more critical. Solar panels require room to function properly with nothing in the way to shade them, which is difficult on commercial vessels where room is at a premium and cargo is shifting back and forth to cast shadows. Additionally, salt accumulation and rough sea conditions tend to impair the performance of the solar panels to a large degree, so they would need to be washed and serviced regularly, which really defeats the purpose of having a fully self-sufficient system.

Due to these constraints, another solution is to utilize wind power exclusively. Wind power does not rely on the weather and the hours of the day like solar power does. Vessels travel, so they generate their own wind. Even when the natural breeze is not strong, the motion of the ship generates airflow that is usable to produce power. This enables wind power to be utilized as a successful source of power without the use of direct sunlight.

Making the power system rely solely on wind power simplifies the power system and enables the maritime system to function independently. As a result, solar panels are not needed on the system. Wind turbines are placed in the most optimal spots, unbothered by shade or confined areas. Vertical-axis wind turbines (VAWTs) are particularly useful since they function optimally in heavy and fluctuating winds that prevail at the sea. VAWTs also contain fewer parts that move in comparison to the horizontal-axis turbines that are commonly employed. With fewer parts that move, the turbines require less maintenance and tend to be more durable.

The most effective aspect of the wind-only system is that it will utilize the movement of the ship to generate power at all times. Commercial vessels tend to travel either on their own engines or on ocean current, so wind power will likely always be available to utilize. This provides the constant power supply to the radar system that does not rely on the weather or time of day. Not utilizing solar panels also prevents issues of salt buildup on the panels and wear and tear on the panels due to maritime conditions, which would inhibit the radar system from receiving adequate power.

The largest concern with solely utilizing wind is if the ship is at rest for extended periods of time. At this point, the power system would not produce power for a while. This is easily rectified with the use of battery storage so it will continue to power even when no wind is present. The maritime radar system itself could also be designed to consume less power while not in use to conserve power when not producing much power. With commercial ships operating most of the time while moving, the appropriate amount of wind power with ample battery storage would be beneficial in patrolling the seas.

3.9.1 Implementation of a Wind Powered System

With wind power being the primary energy source, the power system must be designed to ensure stable and continuous operation. The system consists of a wind turbine, a charge controller, a battery storage unit, and a power regulation unit which work together to provide power to the maritime radar system and its supporting components such as the optical system. Each part of the system plays a critical role in maintaining self-sufficiency; precaution must be taken to ensure that power delivery remains uninterrupted, even in variable wind conditions such as when the vessel the system is on is docked or anchored at sea.

The wind turbine is the most important part of the power system, as it is the sole source of energy generation. Since the maritime device will be mounted on a moving vessel, the turbine will always have access to some level of airflow, whether from natural wind or the movement of the ship itself. This allows for a more consistent power source compared to land-based applications, but placement still needs to be optimized to avoid turbulent airflows caused by the vessel's superstructure or from a lack of airflow due to structures on the mast or helm of the ship. Ensuring the turbine is placed in an area where wind flow is as unobstructed as possible will maximize efficiency and reduce strain on the system which is imperative when wind is the sole power source being utilized. Additionally, since the turbine will be constantly exposed to salt spray and harsh marine conditions, it must be resistant to corrosion and built to withstand high wind speeds without suffering from excessive wear and tear.

Since wind speeds fluctuate, the power output from the turbine will not always be consistent. To manage these fluctuations and prevent damage to the system battery, a charge controller is necessary. The charge controller regulates the flow of energy from the wind turbine to the battery, ensuring that power is delivered safely and efficiently without the risk of damaging the battery. Without this, voltage spikes from strong gusts of wind could overload the battery or damage connected components, while low wind speeds could result in inefficient charging. The charge controller also prevents overcharging, which is critical for maintaining the longevity of the battery. When integrating the charge controller, care must be taken to place it in a protected area, as excessive exposure to moisture or salt can lead to failure of the power system.

Since wind power is intermittent in nature, there needs to be a battery storage system to ensure stable operation when production is low. The battery acts as a buffer, storing energy during times of high wind and delivering power when the turbine is not generating enough by itself. The capacity of the battery must be sufficient to maintain the system in periods of prolonged low wind, particularly if the boat is moving or is positioned where airflow is non-uniform. But batteries can also be a potential point of failure if they are not serviced, as overcharging and deep discharge can drastically reduce their lifespan. A properly integrated battery management system will be monitoring charge levels and compensating for energy usage to prevent these issues. Battery placement must also receive serious attention to avoid subjecting it to the high temperatures, vibration, and corrosive chemicals, all of which have negative impacts on performance.

The final component in the power system is the power regulation unit, which delivers

stable and proper power to the radar, communications equipment, and processing unit. System components demand different voltages and currents; thus, regulation must be carried out to prevent causing harm and allowing maximum operation effectiveness. Lacking this regulation, power delivery fluctuation can result in system failure or inefficiency, something that is not acceptable to a completely autonomous system. The power regulation unit must also possess protective mechanisms such as overcurrent protection and circuit isolation so that electrical faults are not allowed to propagate to the entire system. Insulation and good wiring also matter, as marine conditions pose certain challenges such as corrosion from salt, constant movement, and exposure to water.

All these features placed together allow the wind power system to provide constant and autonomous power to the maritime radar system. Employing only wind as a power source, the system avoids the maintenance and reliability issues of solar panels and uses the natural motion of the ship to generate power. Through meticulous design and integration, the system can maintain the radar in operation without external power supply, ideal for maritime surveillance applications.

Table 3.28 *- Renewable Energy Comparison*

Specification	**Solar Power**	**Wind Power**
Energy Source	Sunlight	Wind (natural or vessel-generated)
Availability	Daytime only, weather-dependent	Continuous with vessel motion, less weather-dependent
Power Consistency	Intermittent (sun-dependent)	More consistent (motion-enhanced)
Space Requirement	Large, unobstructed area for panels	Compact, flexible turbine placement
Space Requirement	Compact (smaller footprint)	Larger (wider blade span, taller structure)
Installation Complexity	High (wiring, brackets, panels)	Moderate (turbine mounting)
Maintenance Needs	High (cleaning, servicing for salt/wear)	Low (fewer moving parts in VAWTs)

Environmental Impact	Salt buildup, wear from sea conditions	Minimal (no shading or salt issues)
Cost Considerations	Higher (complexity, maintenance)	Moderate (simpler system)
Reliability in Maritime	Lower (shadows, degradation)	Higher (vessel motion ensures airflow)
Backup Needs	Essential (night/low sun periods)	Essential (stationary periods)

3.9.2 Comparison of Vertical and Horizontal Wind Turbines

Selecting the right wind turbine type for the maritime radar system requires understanding the core differences between vertical-axis wind turbines (VAWTs) and horizontal-axis wind turbines (HAWTs). Each design brings unique strengths and challenges to achieving a self-sufficient power solution in the harsh marine environment, shaping the foundation for subsequent component choices.

HAWTs, the conventional choice for wind energy, feature blades rotating around a horizontal axis, aligned perpendicular to the wind. This design excels in steady, unidirectional wind conditions, offering high efficiency and substantial power output—ideal for meeting significant energy demands. However, their reliance on wind direction necessitates a yaw mechanism to pivot the turbine, adding mechanical complexity. In a maritime setting, where salt corrosion and constant motion amplify wear, this increases maintenance needs and failure risks. HAWTs also require more space due to their larger blade spans and taller structures, posing challenges for installation on a vessel where room is limited and stability is paramount.

VAWTs, with blades rotating around a vertical axis, capture wind from any direction without adjustment. This omnidirectional nature suits the turbulent, variable winds at sea, enhanced by the vessel's motion-generated airflow. Their compact footprint fits constrained spaces, and a simpler design with fewer moving parts reduces maintenance—a key advantage for an autonomous system. Yet, VAWTs typically deliver lower power per unit due to reduced aerodynamic efficiency, potentially requiring multiple units or larger designs to meet needs. As a newer technology, their long-term performance in marine conditions is less proven, adding some uncertainty.

For maritime use, HAWTs offer raw power but demand oversight and space, conflicting with self-sufficiency goals. Their size may interfere with equipment like radar, and mechanical complexity heightens vulnerability. VAWTs prioritize adaptability and resilience, leveraging dynamic winds and minimizing upkeep, though their output must

align with system demands, likely paired with battery storage. This comparison guides the selection of specific turbine components, balancing efficiency, practicality, and environmental fit.

Table 3.29 *- Wind Turbine Mounting Comparison*

Specification	**Vertical-Axis Wind Turbine (VAWT)**	**Horizontal-Axis Wind Turbine (HAWT)**
Power Efficiency	Moderate (lower aerodynamic efficiency)	High (optimized for steady wind)
Wind Directionality	Omnidirectional (no alignment needed)	Directional (requires wind alignment)
Mechanical Complexity	Low (fewer moving parts)	Higher (yaw mechanism required)
Maintenance Needs	Low (simpler design)	Moderate to High (more components)
Space Requirement	Compact (smaller footprint)	Larger (wider blade span, taller structure)
Turbulence Tolerance	High (effective in variable winds)	Moderate (prefers steady flow)
Installation Complexity	Low (no yaw system needed)	Higher (requires stable, adjustable mount)
Typical Power Output	Lower per unit (scalable with multiples)	Higher per unit (single unit sufficient)
Marine Suitability	High (adapts to motion and turbulence)	Moderate (challenged by corrosion, space)
Technology Maturity	Emerging (less tested long-term)	Mature (extensively proven)

3.9.3 Wind Turbine Part Selection

The first wind turbine considered is the TESUP Magnum Horizontal Wind Turbine [15]. This wind turbine claims to generate 10 Kilowatts of power in a day inland which would be ample for the power of our system, and with the constant winds that are present on the open ocean, the TESUP turbine would likely generate more power than its advertised amount. In addition, “Rotation starts at a wind speed as low as 3 m/s, and from there, it continues spinning smoothly and effortlessly thanks to its large wind-catching surface

area [15].” The ability for the turbine to rotate towards the direction of wind is essential for a maritime environment where wind can be turbulent and omnidirectional.

An additional benefit with the TESUP system is the ability to pair the wind turbine with the TESUP solar panels. Although wind turbines were decided to be the sole power generator on the maritime radar system, discussion on the inclusion of solar panels is still on the table and having the option for a plug and play solar panel would be plus when considering the TESUP vertical wind turbine. Additionally, this manufacture provides the ability to only purchase the fins of the wind turbines separate from the alternator. This is a nice convenience, especially for maintenance because if the wind turbine were to get damaged, we could replace only the damaged fins rather than having to purchase an entire new generation unit.

Some potential complications with this wind turbine include the form factor of the horizontal axis and the complications with mounting it along with the 6-foot RayMarine Radar that needs to be powered by the wind turbine. The horizontal-axis nature of the TESUP Magnum's design creates significant issues relative to its compatibility with marine radar systems. Unlike the vertical-axis wind turbines, whose field of wind collection is uniform regardless of wind direction and which do not require ongoing adjustment, a horizontal-axis machine has to be positioned relative to wind direction. This is particularly burdensome on a moving platform, whose wind directions change often and erratically. The need to incorporate a yaw mechanism to slew the turbine introduces mechanical complexity and additional sources of potential failure, especially in the harsh marine environment where concerns over corrosion and abrasion are high. Additionally, the large swept area of the TESUP turbine could result in interference with radar space, either blocking its field of view or requiring a mounting location that is less than ideal for effective collection of energy.

Another consideration is the physical size and installation requirements. While highly efficient, the large size of the TESUP turbine requires a heavy, vibration-damping mount to maintain balance when mounted on a vessel. Such an addition creates a higher total weight and load on the structure, which could impact the balance and operational efficiency of the boat. Due to the need for a light, low-maintenance solution that is efficient over a wide range of wind conditions, other wind turbine designs, particularly vertical axis designs, could be a better solution to this particular situation.

When looking at the TESUP line-up of devices and power generators, a vertical axis turbine came up and potentially could be a better fit to the form factor that we are trying to achieve with the design of our maritime radar system.

The Atlas Vertical Wind Turbine Generator [16] is another style of generator that is available by TESUP and, unlike the Magnum Horizontal Wind Turbine, the Atlas has a much smaller footprint due to its horizontal design. Something that is a plus is that the vertical design also allows for multiple blade orientations. TESUP has 3 sets of blades, blades for high wind, moderate wind, and low wind. For our design, we expect high wind due to the maritime environment. Although this is what we expect, actual environmental tests may differ than our expected values. Having this in mind, the multiple fan blade

orientations allow us to be prepared for any possible unexpected discrepancies in environmental conditions.

Like the TESUP magnum, the atlas has a projected daily generation of 10kW which would be ample to charge the onboard battery and maintain power to the radar, star link, and the MCU. The manufacturer claims that the wind turbine is “well-protected from weather damage” however, this claim may only pertain to inland installments as conditions in a maritime environment may cause weather related issues to arise. Unlike the magnum, the atlas “wind turbine requires 4 m/s of wind speed for rotation to start.” This is 1 m/s more than the magnum. This extra wind speed required causes the vertical axis to be a less efficient design than the horizontal axis wind turbine. As mentioned before, this will likely be a non-issue due to the excess wind in maritime environments but if needed, “by adding the optional low-wind blade set, Atlas can start rotating at just 3 m/s wind speed,” similar to the wind speed of the magnum. The low wind blade set does increase the profile of the wind turbine, but we expect to use the medium or high wind blades, so this will likely not come to pass.

Another plus of the TESUP system is the integrated phone application that can be paired with the wind turbine. This phone application is able to link up with the alternator to measure and control the voltage and current that the turbine produces. This nifty feature is essential to the maritime radar system as it can ensure the battery is being efficiently charged without the potential for damage by over charging the battery with excess or high voltage.

Continuing research for vertical mounted wind turbines. The Tulip or flower shaped wind turbines mounted on a vertical axis seemed to align with our design specifications.

During our research on wind turbines, we reviewed the paper "Comparison of Rose, Aeroleaf, and Tulip Vertical Axis Wind Turbines (VAWTs) for Alternative Electricity Generation.[17]" It is a review of three VAWTs marked by flower-like designs—Rose, Aeroleaf, and Tulip—all having varying strengths and weaknesses. Their efficiency when in operation and performance factors need to be studied in order to establish the appropriate turbine suitable for various uses, especially marine environments.

The Rose turbine is a newly designed vertical axis wind turbine (VAWT) that is optimized specifically for low wind speeds and high stability. In experimental tests, it showed the highest power producing capacity, generating 2.34 watts at a 9 m/s wind speed, outperforming the Aeroleaf and Tulip turbine designs. One of its greatest advantages is its ability to reach stable rotational speeds very quickly, making it highly adaptable to settings where wind conditions fluctuate. In addition, the Rose turbine also has a shorter Energy Payback Period (EPP), which means that it recovers its energy input in manufacturing much more quickly than other models. However, its complex blade structure translates into a slightly bulkier design and could result in higher material costs. With its efficiency and reliability, it potentially could be a compelling design to utilize on our maritime radar installment. Something to be noted is that the maritime environment could be highly turbulent and as a result, the efficiency of the rose design could be disturbed due to this variability.

The Aeroleaf turbine has been particularly designed to be integrated into urban environments, often in the forms of wind trees where several small turbines mimic the leaf-like aesthetic. The turbine is moderately efficient, requiring a wind speed of at least 7 km/h (1.94 m/s) to start power generation. While its power generation is less than that of the Rose turbine, where it delivers 2.27 W when wind speed is 9 m/s, the Aeroleaf is highly compact and performs exceptionally well when subjected to turbulent airflow. The Aeroleaf turbine is highly suitable for low-scale energy harvesting and could be highly valuable in maritime environments where space is critical. However, its comparably low power generation, in addition to the high wind speed required to trigger operation, might be limiting in its potential application where large-scale offshore power generation is required. This design could possibly suit the needs of our maritime system as we expect both ample wind and turbulent conditions in the maritime environment. There is no real benefit to having excess power in the radar system, and therefore, maximizing efficiency is energy production is not necessary. The wind turbine needs only to be as efficient as is needed by the power demand of the system and if Aeroleaf can provide a formfactor that better suits the system requirements and functions better under turbulent conditions, then it could potentially be a viable choice for our power generator.

The Tulip turbine shows expertise in handling turbulent wind conditions that are most likely to occur at sea. However, in spite of its turbulence-handling ability, it has the lowest power output of the three models with a very small 1.76 W at 9 m/s wind speed. A big advantage of this turbine is that it produces minimal noise, making it suitable to be used in residential areas; however, it takes a longer time to reach a stable rotational speed compared to other models. The longevity and ability to withstand turbulence of the Tulip turbine might make it suitable for use in our project, especially since lowering the maintenance is very desirable in our system; however, its low efficiency makes it less desirable for open sea conditions or energy-demanding maritime activities.

For our maritime use, these flower based wind turbines provide a new perspective on finding a balance between efficiency as well as tolerability to turbulent conditions that will likely be the norm on the open ocean where we plan to implement our radar instalment. From the paper, the tulip and the rose style wind turbine seemed to potentially meet some design aspects that we are looking for in the wind turbine. The adaptability to turbulent conditions is a major concern that needs to be addressed in our design requirements and the utilization of a flower design as the blades for the wind turbine could allow for the creation of a more robust power generation system.

From flower turbines[18], the smaller tulip style turbine seemed to potentially be a viable turbine for our maritime radar system. Within the research on flower shaped wind turbines, the tulip design was described as an excellent option for turbulent locations but not necessarily being the most efficient design. The expected Watt production of the flower tulip turbine is 62.2 w/h or around 1500 Watts in a day. This expected generation is with a steady wind of 27 miles per hour. Although this wind speed is unlikely to occur inland, when placed one a maritime vessel, we expect to get at least this wind speed or possibly more, especially when the vessel is in transit. Something that is concerning is the height of the wind turbine as the blade comes in at 1.14 meters tall. This large blade could

pose issues in mounting as we plan to mount the wind turbine in an inverted manor beneath the Ray Marine 6-foot radar.

Table 3.30 *- Wind Turbine Comparison*

Specification	**Small Tulip Wind Turbine**	**TESUP Magnum Wind Turbine**	**TESUP Atlas Wind Turbine (High Wind Blades)**
Model Number	FT 1.15M Turbine	Magnum	Atlas
Type	2-Bladed Vertical Axis	Horizontal Axis	Vertical Axis
Rated Power	100 W at 14.5 m/s	10 kW	10 kW
Expected Production	≈62.2 Wh at 12 m/s, scalable with bouquet effect	Up to 240 kWh/day	Not specified
Generator	High Efficiency Permanent Magnet	Permanent Magnet	Permanent Magnet
Blade Material	Thermoplastic	Composite Materials	Aluminum
Turbine Diameter	0.55 m (1.79 ft)	2.35 m (7.71 ft)	0.46 m (1.50 ft)
Blade Length/Height	1.15 m (3.77 ft)	2.35 m (7.71 ft)	1.5 m (4.92 ft)
Total Height	1.4 m (4.58 ft)	Not specified	1.5 m (4.92 ft)
Total Weight	≈20 kg (44.1 lbs)	20 kg (44 lbs)	20 kg (44 lbs)
Cut-In Wind Speed	0.7 m/s (1.57 mph)	3 m/s	3 m/s
Cut-Out Wind Speed	Inverter controlled	Not specified	Not specified
Max Survival Wind Speed	54 m/s (120.8 mph)	50 m/s	50 m/s
Operating Temperature Range	-15°C to 50°C	-25°C to +40°C	-25°C to +40°C

Warranty Period	5 years (Flower Turbines parts)	2 years standard	2 years standard
Price	$1,214.06 (Off-Grid)	$899.00 (Generator only)	$899.00 (Generator only)

3.9.3 Charge Controller Selection

For this project we will be needing to power a RayMarine Radar, a Pi microcontroller, and a Star Link Mini along with any peripherals that we may add to the project. To power these items, a battery will be used to store the energy collected by the wind turbine. The process to store energy in this battery, however, needs to be regulated for the health of the system. This regulation can be implemented through the use of a charge controller which can prevent the overcharging of a battery, or a battery being damaged from too much current being transferred at one time. Having this controller is imperative, especially due to the variable power output that comes with wind power.

When finding a charge controller for a maritime radar system, there are some things that we must keep in mind. The waterproofing of the system is important as well as the size of the charge controller. We do not want something that is intolerant of water or a device that is too large as to not be practical to implement in our system.

The first charge controller that was investigated was a charge controller by TESUP[19]. Since we have investigated TESUP products before when looking into wind turbines, continuing to use TESUP products could potentially lead to a more seamless integration of parts into the power system of the maritime radar system. The TESUP charge controller is compatible with all TESUP turbines, however compatibility varies with other manufacturer turbines. The charge controller can manage up to 9000 Watts of power and can be used with 24V- 48V batteries which comply with the type of battery that we will be utilizing in our build. The voltage can be easily adjusted and there is connectivity through the TESUP app which can be highly beneficial when getting metrics while testing. Some things to note are the dimensions of the controller, with the controller taking up about .5 feet cubed. The space that the controller would take up is significant and housing such a component would need to be considered as a potential issue if we were to choose this charge controller.

The next charge controller that was considered was the ATO-WTCMPPT-500[20] which is a charge controller rated up to 500W with an output of either 12 or 24 volts. This charge controller is a more simple design compared to that of the TESUP as there is no LCD display or app connectivity. This lack of connectivity, however, is made up with the price of the component as it is cheaper than the TESUP charge controller. This may be a benefit as the charge controller is only in place to ensure the proper charging of the battery, and if it is known that the controller outputs the desired voltage, then no other information is needed. Another benefit is the profile of the controller. The dimensions of the controller are 100*80*15mm which is substantially smaller than the TESUP and therefore, the installation and housing of the controller is likely to be a non-issue when incorporated into our design.

The final controller that was looked at was the Pikasola Wind Turbine Charge Controller[21] which also has the capability to handle DC inputs of both 12 and 24 Volts. This charge controller has a rating of IP67 which entails its waterproofing. This is an essential characteristic because in a maritime environment, waterproofing is essential and with this rating, the controller may not have to be enclosed in a weatherproof box which could save on space and material cost. This charge controller can take up to a 600W wind turbine which would be sufficient to our maritime radar system design.

Table 3.31 *- Wind Turbine Comparison*

Specification	**TESUP Magnum Charge Controller**	**ATO-WTCMPPT-500 Charge Controller**	**Pikasola Wind Turbine Charge Controller**
Model Number	Not Specified	ATO-WTCMPPT-500	B07ZNK1C76
Voltage	24V/48V	12V/24V (Auto Recognition)	12V/24V (Auto Recognition)
Weather Rating (IP)	Not Specified	IP67	IP67
Power Rating	Up to 9 kW (9,000 W)	500W	400W/500W/600W
Dimensions (H x W x D)	32.7 cm x 20 cm x 8.5 cm	100 mm x 80 mm x 15 mm	3.1" x 2.5" x 1.2" (7.87 cm x 6.35 cm x 3.05 cm)
Weight	2 kg	0.4 kg	10 Ounces
Price	$199	$99.82	$35
Compatibility	All TESUP Wind Turbine models	Not Specified	Suitable for 400W, 500W, 600W Wind Turbines
IoT Connectivity	Yes	Not Specified	Not Specified
Manual Brake Function	Yes	Not Specified	Not Specified
Dump Load	Included	Not Specified	Not Specified
Adjustable Voltage	Yes, via front panel knob	Not Specified	Not Specified
Material	Aluminum casing	Not Specified	Aluminum alloy

Braking Voltage	Not Specified	Not Specified	DC15V-30V

3.9.4 Power Storage and Battery Selection

As mentioned before, the design of our maritime radar system will include a battery of sorts to store the power that is being produced by the wind turbine. Things that we must consider with the battery is the environment that it will be placed in as well as the nature of the charge it will receive. In a maritime environment, salty ocean spray will be a common occurrence and will induce a faster rate or corrosion on metal parts. In addition, the jostling of the waves created an unstable environment which could disrupt or even damage a battery which typically operates in a stable environment. Along with the environment, the battery will be charged intermittently, in proportion to the wind that the wind turbine receives. This type of charging can cause complications with certain types of batteries that require a steady charge and discharge of power. All of these factors must be considered when choosing a battery for our system because a battery failure would cripple the maritime radar system and require extensive cost when having to repair the damage to the system.

When searching for a battery we also came to find battery systems with integrated components such as charge controllers as well as regulators that could fit the components that we must power, those being the Star Link, the Ray Marine Radar, the PI MCU, and potentially a camera system. The Star Link mini runs 12 to 48 Volts and has an included power supply, so a battery that would have a plug to support this would be ideal. In addition the Ray Marine Radar supports 12 to 24 Volt power so that power connection would also be nice to have native on the battery. Finally the 5 Volt connection for the PI MCU will likely be included on the battery as a USB port.

Having an integrated charge controller could save space on the maritime radar system and also minimize the components which we will have to waterproof. If the battery contains the charge controller, a single waterproof box may be sufficient to house components that cannot get wet.

When researching these batteries, we made sure that they were primarily for maritime use and therefore, rated for the salty and turbulent nature of the ocean environment. One battery that is being considered is the Dakota Lithium 12V 100AH Deep Cycle Heated LIFEPO4 Battery[22]. One of the biggest advantages of this battery is its long cycle life compared to typical lead-acid batteries. With a lifespan of 3000–5000 cycles, it is a durable and reliable choice for long-term use in a marine environment. Its deep cycle capabilities allow it to handle frequent charging and discharging without significant performance loss, which is ideal when paired with an intermittent energy source like a wind turbine. The built-in heating feature is another major benefit, ensuring the battery performs well in cold conditions where lithium batteries typically struggle. Additionally, LiFePO4 batteries are significantly lighter than lead-acid alternatives, making them easier to install and transport. This battery also maintains a steady 12.8V output, providing consistent performance for sensitive electronics. From a safety perspective, LiFePO4 chemistry is thermally stable, reducing the risk of overheating or fire, which is crucial for maritime use. Another key advantage is that it can work with DC-DC converters,

allowing voltage adjustments to power devices requiring 5V or 48V.

However, there are some drawbacks to this setup. Since the battery operates at 12V, devices like the Raymarine radar and Starlink Mini, which typically require 24V or 48V, will need DC-DC step-up converters. This adds extra cost and potential inefficiencies to the system. Another challenge is that charging efficiency depends on wind conditions. If the wind is too weak, the battery may not charge sufficiently, and if it's too strong, a charge controller is needed to prevent overcharging. The Battery Management System (BMS) inside some LiFePO4 batteries may also have limitations on handling high surge currents, which could impact the performance of the radar system. Additionally, using a 12V battery instead of a 48V system means extra hardware like converters and charge controllers are necessary, increasing complexity and potential failure points. While LiFePO4 batteries are lightweight for their capacity, if a higher voltage is required, multiple batteries would need to be connected in series, increasing space requirements.

Another battery type that we considered was a battery meant more for a survivalist, where things such as power connections are available of the battery itself, whereas DC regulators may not be needed to power our system components.

The battery that could possibly meet these requirements is the PREDATOR 350-Watt Power Station, 294 Wh Capacity[23]. This battery truly embodies the plug and play system that we wish to implement in our design. The battery itself has outputs that would be able to power the majority of our components from. The battery includes two 120V-350W AC outlets. These ports would be able to power both the Star Link mini as well as the RayMarine Cyclone radar. The DC conversions would not be a problem since the radar comes with its own native voltage controller and the Star Link mini also has its own power supply which would regulate the power output from the battery. Additionally, the inclusion of a 5V-12W USB type A port would be perfect to power the PI MCU without the need to regulate the output from the battery. These plug and play supports drastically cut down on the components necessary to power the maritime radar system.

Some things of note for this battery system is the way that it is charged. The battery has the ability to be charged by solar which is a huge plus since solar may be used in variations of our radar system. The drawback, however, is that the battery only takes AC charge rather than a DC charge which is a dilemma for us since we will primarily be using wind turbines, a DC source, as power generation. To combat this, the use of an inverter will have to be considered to properly charge the predator battery and avoid damaging it whilst charging. The inverter would take the place of the charge controller because we would no longer need to regulate the voltage that the wind turbine outputs. This inverter would allow the battery to be charged by the wind turbine and there would no longer be an issue with using the battery as our main power source.

Another battery that we considered is the Amped Outdoors 24V 50Ah LiFePO4 battery[24] which presents a strong solution for our maritime power needs, particularly given its lightweight design (26.75 lbs) and high energy density. This makes it well-suited for powering the Starlink Mini, Raymarine Cyclone radar, and Raspberry Pi MCU while keeping the system portable and efficient. Its marine-grade construction ensures it can handle the harsh conditions of a boat environment, resisting vibrations, moisture, and

corrosion, which are all crucial factors for longevity and reliability. Additionally, its built-in Battery Management System (BMS) provides protection against overcharging, over-discharging, and short circuits, which helps maintain safety and extend the battery's operational lifespan. The added Bluetooth monitoring feature allows for real-time tracking of battery health and performance via a smartphone, which is a valuable feature when managing an off-grid power system.

However, integrating this battery with our wind turbine requires careful planning. Since wind energy generation fluctuates due to varying wind speeds, the battery must be charged through a compatible 24V wind charge controller to regulate and stabilize the power input. A wind turbine's output is highly variable, meaning the charge controller must efficiently handle power surges and prevent overvoltage damage to the battery. Additionally, wind turbines tend to generate wild AC power, which needs to be rectified into DC before charging. If the turbine's output doesn't match the optimal charge voltage and current for the Amped Outdoors battery, charging inefficiencies or even battery damage could occur.

Another consideration is whether 24V is the ideal voltage for all our components. While the Raymarine Cyclone radar and Starlink Mini can operate at 24V, we may need DC-DC converters to step the voltage down to 5V for the Raspberry Pi or up to 48V if any future components require it. While, unlike the predator battery, we would need a regulator for the Pi MCU, the added power and efficiency of the Amped Outdoors battery could possibly be a better fit for the power demand of our system. After evaluating multiple battery types and configurations, we are continuing to assess which power system offers the best balance between reliability, compatibility, efficiency, and ease of integration into our maritime environment.

Table 3.32 *- Battery Comparison*

Specification	**Predator 350W Power Station**	**Amped Outdoors 24V 50Ah $LiFePO_4$ Battery**	**Dakota Lithium 12V 100Ah $LiFePO_4$ Battery**
Nominal Voltage	12V DC	24V	12V
Capacity	294Wh	50Ah (1,200Wh)	100Ah (1,280Wh)
Continuous Power Output	350W	50A (1,200W at 24V)	100A (1,280W at 12V)
Peak Power Output	700W	80A (1,920W at 24V)	200A (2,560W at 12V)
Weight	8.8 lbs	26.75 lbs	31.9 lbs
Dimensions (L×W×H)	9.3" x 8.1" x 7.7"	10 1/8" x 6 1/2" x 8 1/4"	12.99" x 6.77" x 8.66"
Battery Type	Lithium-ion	$LiFePO_4$	$LiFePO_4$
Lifespan (Charge Cycles)	3500+	2000+ cycles	3000-5000 cycles

Operating Temperature Range	Not specified	Charge: 0°C to 45°C Discharge: -20°C to 60°C	Charge: 0°C to 45°C Discharge: -20°C to 60°C
Integrated BMS	Yes	Yes	Yes
Bluetooth Monitoring	No	Yes	No
Warranty	Not specified	10 years	11 years
Price	$199	$619 (battery only) $749.99 (with charger)	$799

3.9.5 Identification and Selection of Regulators

The use as well as the system Voltage that we will be running at. If the system were to run off of 12 Volts, then a voltage regulator may be needed to ramp up the voltage to 24 Volts for the Star Link mini as well as a regulator to step down the voltage of the system to 5 Volts for the Pi MCU. These regulators would make it possible for the maritime radar system to utilize batteries such as form Dakota Lithium(12V) or the amped outdoors 24V battery.

In the case of the Star Link mini, the standard input for the system would be AC such as from a wall socket. When pairing the Star Link with a battery that can only output at a single Voltage, some sort of conversion needs to take place to adapt the output to a usable form for the Star Link mini. A prominent solution that is available on the market is a Starlink Mini PoE DC Power Conversion Kit with 12v Accessory Output by MOBILEMUSTHAVE[25]. This device takes the place of a voltage regulator as instead of having to rely solely on the proprietary Star Link mini power supply, this device would take its place and allow for power to be consumed from a DC source such as the 12V or 24V battery options that we have selected. This conversion kit would function for both a 12V or 24V system as it is rated for between 9V and 30V as the input Voltage. This is a really nice feature because modeling our system around only one battery or one voltage can limit the possible expansion of the maritime radar system and with this converter, we gain flexibility to increase or decrease the battery voltage to possibly change to higher power batteries in the future if needed by our system requirements.

The converter is capable of delivering up to 100W at 48V to the Star Link mini which will ensure that the connection is secure and uninterrupted or slowed by poor power output by a cheaper converter. In addition to the output regulation to the Star Link, the converter also comes with two USB type A ports. One is rated for 12V while the other is a standard 5V port. This is a really important inclusion on this device as it means that a regulator would not be needed to power the Pi MCU since we can power the MCU directly from the regulator that manages the Star Link. We would need to be careful with which USB we plug the Pi into as the MCU could be fried if plugged into the 12V USB port instead of the 5V port.

Another regulator that is more budget friendly is the Starlink Mini Step Up Power Cable

12V/24V to 30V 3.5A Converter with DC Power Cable, 9V to 28V Input DC Step-Up Power Adapter Cord for Starlink Mini[26]. Unlike the previous regulator, this regulator solely applies to the Star Link mini and has no other outlets that could possibly power the Pi MCU or any other peripherals. With this in mind, the lower cost that this regulator offers does ease some financial burden. This DC regulator boosts up to a 30V output to power the Star Link mini. This voltage is less than the 48V that the regulator from MOBILEMUSTHAVE provides, but the loss of performance from the under voting of the Star Link mini may not affect the scope of transmissions that we expect to be sending over the Star Link network.

Using the cheaper Star Link regulator would also require the additional use of a DC regulator specifically for the Pi MCU. The PlusRoc Waterproof 12V/24V to 5V Converter DC-DC Step Down Module Power Adapter Compatible with Raspberry Pi [27] would be sufficient and with a price under ten dollars, would not infringe on the cost saving from avoiding the adapter from MOBILEMUSTHAVE. This adapter is capable of stepping down the voltage from a 12V or 24V battery to power a 5V device and comes with the USB type C connector which would power the Pi MCU that will be implemented in our design. The drawback to using these two regulators is the complexity of the system and the worry of waterproofing in our maritime environment. Having multiple regulators will increase the maritime radar system and could potentially incur cost when transporting or dealing with extra waterproofing that may be required. As we move forward, the decision between an all-in-one regulator and a modular, lower-cost approach will depend on our final system requirements, space constraints, and how much complexity we are willing to introduce into the power architecture.

Table 3.33 *- Regulator Comparison*

Specification	**PlusRoc Waterproof 12V/24V to 5V Converter**	**Starlink Gen 3 12V to 56V DC Step-Up Adapter**	**Starlink Mini PoE DC Power Conversion Kit**
Input Voltage Range	8V to 35V DC	9V to 35V DC	9V to 30V DC
Output Voltage	5V DC	56V DC	56V DC
Output Current	Up to 5A	Up to 2A	Up to 2A
Power Output	Up to 25W	Up to 112W	Up to 112W
Efficiency	Up to 96%	Approximately 90%	Approximately 90%
Waterproof Rating	IP68	Not specified	Not specified
Dimensions	1.85 x 1.38 x 0.67 inches	3.5 x 2.5 x 1.2 inches	3.5 x 2.5 x 1.2 inches

Weight	1.59 ounces	4 ounces	4 ounces
Price	$9.99	$29.98	$99.99

3.9.6 System Voltage Comparison

When designing the maritime radar system's power architecture, three primary voltage options must be considered: 12V DC, 24V DC, and AC. Each configuration presents distinct trade-offs in efficiency, wiring complexity, component compatibility, and system cost.

3.9.6.1 12V DC System

A 12V DC system is the simplest DC option, widely used in marine applications due to its compatibility with many off-the-shelf components. However, at lower voltage, higher current (amps) is required to deliver the same power ($P = V \times I$), necessitating thicker gauge wiring to minimize resistive losses and voltage drop over long cable runs. This increases weight and cost, particularly in a distributed system like a maritime radar setup. Additionally, high-current loads (such as the Starlink Mini and Raymarine Cyclone Radar) may require heavy-duty bus bars or power distribution modules to handle the amperage safely.

The Raymarine Cyclone Radar requires a VCM1000 voltage converter module to operate on either 12V or 24V, meaning the 12V system does not eliminate the need for additional power conditioning. Meanwhile, the Starlink Mini can be powered via a DC-DC step-up converter, but efficiency losses must be accounted for. A 12V $LiFePO_4$ battery system (e.g., Dakota Lithium) would pair with a 12V wind charge controller, avoiding the need for an inverter but still requiring careful current management.

3.9.6.2 24V DC System

A 24V DC system offers significant advantages over 12V by halving the current for the same power output, allowing for thinner, lighter wiring and reduced resistive losses. This is particularly beneficial in maritime applications where long cable runs are common. The Raymarine Cyclone Radar still requires the VCM1000, but the higher input voltage may improve efficiency in power conversion. The Starlink Mini would still need a DC-DC regulator, but a 24V input could reduce step-up conversion losses compared to a 12V system.

Battery-wise, 24V $LiFePO_4$ banks are readily available and can be constructed by wiring two 12V batteries in series. A 24V charge controller would be required for the wind turbine, but this setup avoids the inefficiencies of an inverter. The primary drawback is that some low-power devices (e.g., the Raspberry Pi) may need additional buck converters to step down to 5V, adding minor complexity.

3.9.6.3 AC System

An AC power system simplifies compatibility with commercial devices like the Starlink

Mini, which can use its standard AC power supply without modification. However, this convenience comes at a cost:

- Requires an inverter, introducing conversion losses (10-15%) and adding a single point of failure.
- The Raymarine Cyclone Radar still needs the VCM1000 even in an AC system, negating some of the plug-and-play advantage.
- AC battery systems (e.g., Predator) are less energy-dense than DC $LiFePO_4$ banks, leading to shorter runtime under high loads.
- More complex wiring due to AC safety standards (grounding, isolation, etc.).

While an AC system allows for "out-of-the-box" power supplies, the added inefficiency, cost, and reliability risks outweigh the benefits for a self-contained maritime radar system.

3.9.6.4 System Voltage Conclusion

After carefully evaluating the trade-offs between 12V DC, 24V DC, and AC power systems, it is clear that a DC-based architecture is the superior choice for the maritime radar system. The inherent efficiency of DC power—avoiding the conversion losses and reliability risks associated with AC inverters—makes it the most practical solution for a self-contained, power-sensitive application. Additionally, modern $LiFePO_4$ battery technology performs optimally in DC configurations, offering deeper discharge cycles and longer lifespan compared to AC-coupled alternatives. While a 12V DC system benefits from widespread compatibility and simpler component integration, its high current demands necessitate thicker wiring and introduce greater resistive losses over long cable runs. In contrast, a 24V DC system reduces current requirements by half, enabling lighter, more cost-effective cabling while maintaining efficiency—a critical advantage in maritime environments where power distribution spans significant distances. However, the final decision between 12V and 24V will require empirical testing to assess real-world performance factors, including voltage drop under load, power conversion efficiency for critical components like the Starlink and radar, and overall system cost. Preliminary analysis suggests that 24V is the more scalable and efficient option for larger installations, while 12V may remain viable for compact deployments. Further testing will validate these assumptions and determine the optimal voltage for balancing performance, reliability, and cost.

Table 3.34 *- System Voltage Comparison*

Feature	12V DC System	24V DC System	AC System
Voltage	12V DC	24V DC	120V AC
Current Requirements	High (thicker wires)	Half of 12V (thinner wires)	Standard

Wiring Complexity	Moderate (heavy gauge)	Low (standard gauge)	High (AC safety standards)
Efficiency	High (no conversion)	Highest (no conversion)	Low (inverter losses 10–15%)
Component Compatibility	Requires DC-DC converters	Requires fewer converters	Plug-and-play for AC devices
Raymarine Radar Power	Needs VCM1000	Needs VCM1000	Needs VCM1000
Starlink Power	Requires DC step-up	Requires DC step-up	Standard AC adapter
Battery System	12V $LiFePO_4$	24V $LiFePO_4$ (2x 12V in series)	AC battery (e.g., Predator)
Charge Controller	12V required	24V required	Not needed (uses inverter)
Cost	Low–medium	Medium	High (inverter + AC battery)
Reliability	High	Highest	Medium (inverter = point of failure)
Best For	Small installations	Large/long-distance setups	Quick prototyping

3.9.7 Expected System Strain

The maritime radar system's power architecture must be carefully designed to support all operational components while maintaining efficiency and reliability. Our core devices—the Raymarine Cyclone radar, Starlink Mini satellite terminal, gimbal-mounted camera, and Raspberry Pi control unit—each present unique power demands that collectively determine system requirements. The Raymarine radar operates at a nominal 48W (4A at 12V), but requires special consideration for its startup surges and scanning peaks that can briefly reach 60W (5A). This performance characteristic directly impacts our power supply selection, particularly since the radar's specifications mandate a 20A fused connection despite its lower continuous draw.

The Starlink Mini represents the second largest power consumer in our system, with an average draw of 50W (4.2A at 12V) that can spike to 75W during intensive data transfers. When combined with the radar's requirements, these two devices alone can push a 12V system to its operational limits, especially when considering additional components. Our gimbal-mounted camera, while relatively low-power at 15W nominal (20W peaks during active stabilization), contributes to cumulative current demands that must be factored into wiring and connector specifications. The Raspberry Pi's modest

3-12W range might seem insignificant, but becomes relevant when calculating total system draw during extended operations.

These power characteristics create several operational scenarios that test our system's capabilities. During standard operation with the radar, Starlink, and Pi active, we face approximately 110W (9.2A at 12V) of continuous demand. More concerning are peak load situations where all components operate simultaneously at maximum capacity, potentially drawing 167W (14A at 12V). These figures approach the limits of our tested 12V Weston Bank power supply (rated for 10A continuous, 15A peak), suggesting that while functional, the 12V architecture operates with minimal safety margin.

A 24V DC configuration would fundamentally improve this situation by halving current requirements—the same 167W peak load would demand only 7A at 24V. This reduction not only alleviates strain on wiring and connectors but also improves overall system efficiency by minimizing resistive losses. The transition to 24V would particularly benefit long cable runs common in marine installations, while maintaining compatibility with existing power conditioning equipment like the VCM1000 converter and Starlink's DC step-up adapter.

Table 3.35 *- Maritime Radar System Power Requirements Analysis*

Component	Voltage Input	Nominal Power	Peak Power	Current Draw (12V)	Current Draw (24V)	Efficiency	Special Requirements
Raymarine Cyclone Radar	12–24V via VCM1000	48W	60W	4A (nom), 5A (peak)	2A (nom), 2.5A (peak)	92% (VCM1000)	20A fuse required
Starlink Mini	56V via PoE	50W	75W	4.2A (nom), 6.25A (peak)	2.1A (nom), 3.13A (peak)	90% (step-up)	Needs DC-DC conversion
Gimbal Camera System	12V DC	15W	20W	1.25A (nom), 1.67A (peak)	0.63A (nom), 0.83A (peak)	95%	Stabilization peaks
Raspberry Pi 5	5V USB	3W	12W	0.25A (nom), 1A (peak)	0.13A (nom), 0.5A (peak)	85% (buck converter)	5V regulation needed

Table 3.36 *- System Power Totals*

Mode	Voltage	Power Draw	Current Draw (12V)	Current Draw (24V)	Notes
Idle Operation	12V/24 V	18W	1.5A	0.75A	Pi + Camera only
Standard Operation	12V/24 V	110W	9.2A	4.6A	Radar + Starlink + Pi
Peak Load	12V/24 V	167W	14A	7A	All components max
Emergency Mode	12V/24 V	48W	4A	2A	Radar only

3.10 Machine Learning Techniques

This section discusses different machine learning techniques/models that can be used on the AIS data sets collected by VHF transponders or Space-based transponders. These techniques aid in the classification of maritime vessels and the detection of anomalous behavior. Currently, maritime law enforcement often rely on significant manpower in order to cover the vast area that maritime vessels can traverse. Using computer-assisted tools monitoring radio frequencies and utilizing radar systems can cover a larger area in a much shorter amount of time. Machine Learning can be used to create a prediction based on the data parsed from the radio frequencies transmitted and received by radars and AIS transponders. Models used in classification include Gradient Boosting, Random Forests, Convolutional Neural Networks, and Density-Based Spatial Clustering of Applications with Noise (DBSCAN).

3.10.1 Gradient Boosting

Gradient Boosting is an ensemble method used for classification and regression tasks. Oftentimes, they use a series of decision trees and combine them to improve performance. Usually gradient boosting models start with a shallow decision tree and make initial predictions, calculate the errors based on the target values, and refine the model using the residuals. The process is repeated to attain the best results. Popular models of Gradient Boosting include XGBoost, LightGBM, and CatBoost.

XGBoost utilizes parallelization during tree building, alpha and lambda regularization to mitigate overfitting, handling of missing values internally, tree pruning using “max-depth” and built-in early stopping also to avoid overfitting. XGBoost performs better on smaller data sets. The algorithm’s main drawbacks are its memory usage, which is larger than models such as LightGBM, and it requires additional encoding as it is not native categorical. XGBoost is featured as one of the classification techniques discussed in section 3.11.1 in the classification algorithm Multi-Feature Ensemble Learning Classification Method (MFELCM) [99].

LightGBM, similarly, utilizes decision trees, however with deeper splits. It is known for efficiently handling memory and being able to perform quickly on large datasets with high-dimensional data. LightGBM is also native categorical, so no additional encoding is required. Its main drawback is that it can overfit on smaller data sets, so preprocessing and regularization is required. LightGBM is discussed in more depth and is the primary focus in processing AIS data in section 3.11.2 [100].

CatBoost is known for its native categorical support. It auto-handles missing values and categorical classifiers internally with its own encoding, so it is easy to work with as it can utilize default parameters. CatBoost also uses Ordered-Boosting instead of plain gradient boosting. Ordered Boosting only utilizes past data in a shuffled order to compute predictions. This allows the model to simulate real life situations where we cannot look into the future to determine an outcome. CatBoost works best when classifying data with lots of categorical features. CatBoost was tested in lieu of the LightGBM algorithm in the paper discussed in section 3.11.2, however, it performed worse than LightGBM, so it is not discussed further.

3.10.2 Density-Based Spatial Clustering of Applications with Noise (DBSCAN)

A Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm is used to group similar data points into clusters based on their density. Dense clusters of data are identified based on points that have a minimum number of neighbors, defined by an minimum points parameter, called Core Points, utilizing a parameter, epsilon, that defines the radius around a point to look for a neighbor. DBSCAN also identifies Border Points, which are points that are not Core Points, but contained within the radius of a Core Point, and Noise Points, which are not contained within the radius of any Core Points. By selecting a random starting point, DBSCAN can determine if any point is a Core Point and recursively check all of the density-reachable (or points contained within the radius of a Core Point) and repeat for all unvisited points.

Drawbacks to DBSCAN include its computational expense and its sensitivity to parameter choices. DBSCAN, during its training and testing, must iterate through every data sample in order to evaluate and predict the necessary clusters. On large datasets, this can be computationally expensive. The minimum points and radius parameters also need to be carefully chosen. Increasing the radius and/or decreasing the minimum neighbors for Core Points can lead to clusters that are larger than desired , underfitting the mode and decreasing the radius and/or increasing the minimum neighbors for Core Point the clusters will not be large enough to form any meaningful distinction from all possible routes.

DBSCAN is an effective algorithm for classifying clusters of data, while also filtering noisy data sets. With AIS transmission and its data often being inconsistent, thus creating lots of noise, DBSCAN provides an effective method for grouping dynamic data, such as a vessel's trajectory, while being able to filter noise. DBSCAN is one of the primary algorithms in the Traffic Route Extraction and Anomaly Detection (TREAD) [96] methodology discussed in section 3.12.2. DBSCAN is used in TREAD to cluster and determine the normal trajectories of many classes of vessels over long periods of time

and also simulate normal paths for vessels in the future. This methodology is beneficial for our MDAS as the radar data we collect on a detected object will not have any historical context. The ability to classify the vessel and map the detected object to a normal and future trajectory is valuable. However, as our system intends to be lightweight at sea, the base station performing the classification needs to have significant computational power to simulate the paths of radar detected objects.

3.10.3 Random Forest

A Random Forest algorithm utilizes an ensemble of Decision Trees to make predictions. It is useful in classification. A Decision Tree uses the entropy of the features within the sample data and calculates the largest Information Gain to decide which nodes to split the tree. The node with the highest Information Gain resides at the root of the tree, where each node below it contains the entropy of the information as it corresponds to its parent node. The Decision Tree then uses the leaf (or terminal) nodes to make decisions on the classification of a given sample.

Decision Trees are extremely useful and are often one of the best methods for classification. Its drawbacks come from the nature of the model, many of which can be addressed using a Random Forest. Decision Trees tend to overfit, the more layers (or splits) the tree contains. As entropy increases during training, the model tends to learn the intricacies of the training set, which clouds its judgement in its performance on the test set leading to high training accuracy, but lower test results. Shallower trees tend to not contain enough entropy at the leaf nodes to make any meaningful decisions and perform poorly on both the training and test sets.

To resolve these issues, we can split the training set and test sets into smaller sections, using a technique called K-Fold Cross Validation, and parallelize multiple Decision Trees to uniquely predict each split section of the data set. This Random Forest can then be used to create multiple predictions and use the highest probability for each class to classify the samples. Furthermore, the samples can be weighted based on their feature importance. By training the Random Forest on the sample set, we can classify the training set, calculate the feature importance based on its Information Gain, and repeat inversely using feature importance as weights. Since it is shown in many different AIS-based classification models [99][100] that classification predictions heavily rely on the static data, such as size of the vessel, to determine its class, weighting each feature allows AIS models to use the dynamic data, such as speed over ground, longitude, and latitude, to contribute to the classification of the vessel.

The previously described features of the Random Forest machine learning model make it one of the best and most applicable models for classifying vessels using AIS and radar information. As our MDAS is in its initial and prototyping phases, the data samples will be small. Therefore, its versatility allows us to scale the Random Forest and its Decision Trees to best fit smaller data sets.

3.11 Classification Algorithms

This section discusses different machine learning techniques/models that can be used on the AIS data sets collected by VHF transponders or Space-based transponders. These techniques aid in the classification of maritime vessels and the detection of anomalous behavior. Currently, maritime law enforcement often rely on significant manpower in order to cover the vast area that maritime vessels can traverse. Using computer-assisted tools monitoring radio frequencies and utilizing radar systems can cover a larger area in a much shorter amount of time. Machine Learning can be used to create a prediction based on the data parsed from the radio frequencies transmitted and received by radars and AIS transponders. Models used in classification include Gradient Boosting, Random Forests, Convolutional Neural Networks, and Density-Based Spatial Clustering of Applications with Noise (DBSCAN).

3.11.1 Multi-Feature Ensemble Learning Classification Method (MFELCM)

This section discusses the implementation of a Multi-Feature Ensemble Learning Classification Method (MFELCM) which attempts to leverage AIS data through space-based systems (i.e. satellites) to classify maritime vessels [99]. MFELCM utilizes an ensemble of machine learning techniques including two random forests, extreme gradient boosting (XGBoost), a bidirectional gated recurrent unit (Bi-GRU), and a one-dimensional convolutional neural network to perform classification. The ensemble of machine learning techniques provides a robust way of classifying maritime vessels due to their dynamic nature of travel, speeds, and course. It utilizes the different models to predict the classification on specific types of data utilizing the appropriate model to do so. To guarantee the validity of the data set, MFELCM uses each vessel's unique MMSI to ensure that unique vessels do not appear in the training, validation, and testing data sets. As the ground truth is known on static data for vessels such as its width, breadth, area, etc., a random forest is one of the better models for classification for that set of data.

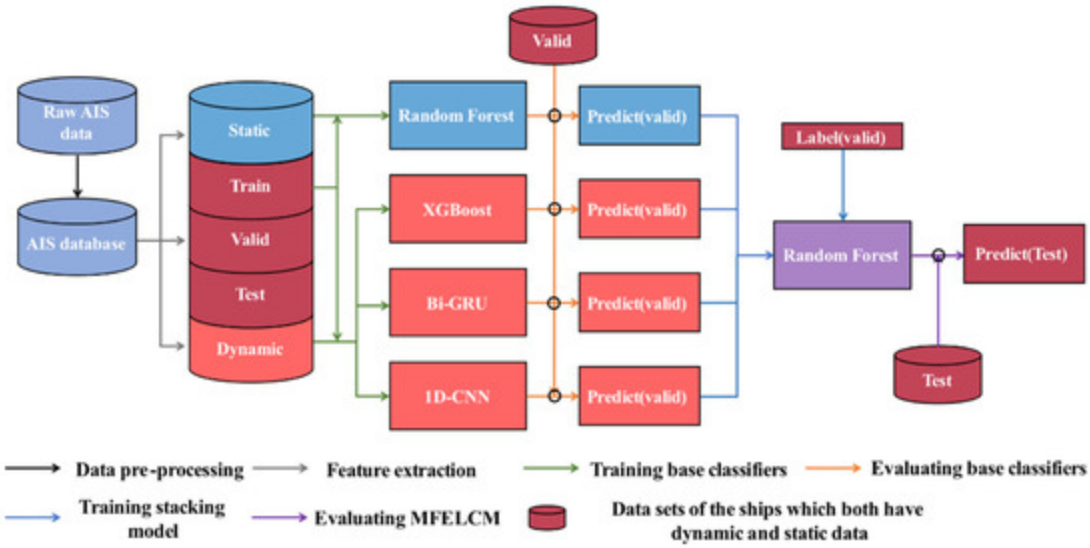


Figure 3.7 *- MFELCM Machine Learning Flowchart.*

The dynamic features of maritime vessels, however, require much more processing in order to classify a vessel against its noisy data. Dynamic data includes the AIS message for each vessel at a given time interval, T^{i}_{m}, which includes the Dynamic Feature Data about the vessel, DFD^{i}_{m}. Dynamic features include the current time (depending on the implementation this could be either message sent or received time), longitude and latitude, course over ground, speed over ground, rate of turn, acceleration, etc. Given the nature of AIS transmission and some of its faults, each set of dynamic features corresponding to a given vessel is given its own time-series feature set, TS^{i}_{m}. This allows each vessel to contain its own data as it traverses along its course and allows each feature to be smoothed accordingly. Noise in the data will include incorrectly predicted longitude/latitude by the onboard GPS, incorrectly calculated speed/course over ground, or simply missing AIS messages over a significant period of time. MFELCM uses three machine learning models to perform the smoothing and classification of the dynamic data, a 1-Dimensional Convolutional Neural Network (1D-CNN), a Bidirectional Recurrent Gated Unit (Bi-GRU), and Extreme Gradient Boosting. Figure 3.7 displays the Dynamic Feature Data and its Time-Series equivalent over a given interval, T.

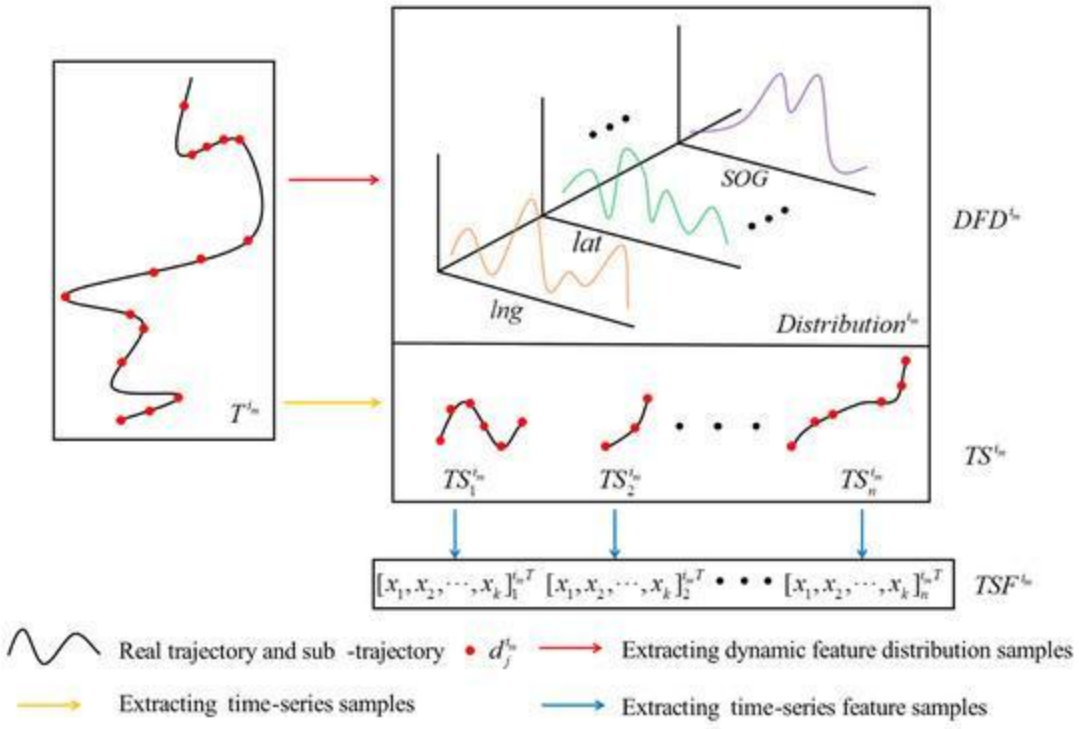


Figure 3.8 *- Process of dynamic features extraction.*

This MFELCM demonstrates high accuracy on the sample data. Its F1-score, the standard measure for predictive performance of machine learning algorithms, is approximately 93.5% on all data samples and 90% when evaluated on each ship and the entirety of its data samples. Figure 3.8 displays the accuracy and f1-score for each model on each class of ship, as well as on the entire set of data samples where class 0 to class 3 represent passenger ships, tankers, fishing boats, and cargo ships, respectively.

Items		Evaluated on Samples					Evaluated on Ships				
		Random Forest	1D-CNN	Bi-GRU	XGBoost	*MFELCM*	Random Forest	1D-CNN	Bi-GRU	XGBoost	*MFELCM*
Accuracy	class0	0.9931	0.9217	0.7191	0.8548	***0.9985***	0.8889	0.6667	0.6667	0.7778	***0.8889***
	class1	0.9320	0.7861	0.6052	0.5467	***0.9344***	0.9237	0.6271	0.6864	0.6102	***0.9322***
	class2	0.9340	0.8879	0.7690	0.8117	***0.9332***	0.9032	0.7903	0.7419	0.8065	***0.9355***
	class3	0.8244	0.6173	0.5972	0.6839	***0.8797***	0.8287	0.7315	0.6296	0.7546	***0.8750***
Total Accuracy		0.9205	0.7914	0.6566	0.7008	***0.9362***	0.8696	0.7077	0.6643	0.7222	***0.9010***
F1 score		0.9196	0.7873	0.6629	0.6991	***0.9359***	0.8719	0.7127	0.6681	0.7239	***0.9019***

Figure 3.9 *- Performance evaluation of base classifiers and MFELCM.*

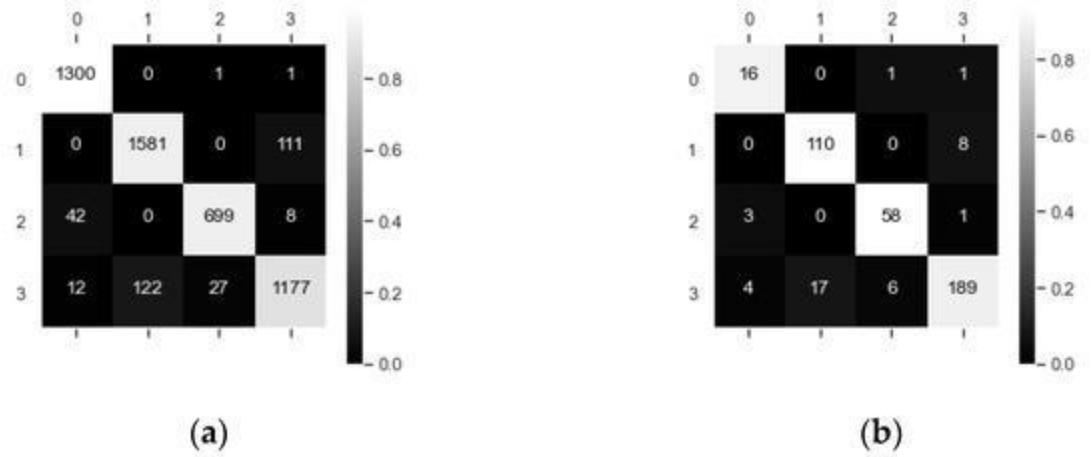


Figure 3.10 *- **(a)** Confusion matrix of samples (MFELCM); **(b)** Confusion matrix of ships (MFELCM).*

MFELCM performs well to classify different maritime vessels. It provides a robust method for parsing, calculating, and smoothing the dynamic data sets. The biggest challenge in classifying maritime vessels using AIS is the noise in the data. As previously mentioned, any errors in the on-board GPS unit, vessels moving in and out of range of the designated receiver or discrepancies in the time of the message received can skew a predicted path for the vessel. MFELCM's method for converting the dynamic data set in a vessel specific Time-Series, separation of static and dynamic data sets, and utilization of multiple machine learning algorithms specific to the appropriate data set allows this model to be accurate in its predictions and useful for classifying AIS messages.

3.11.2 LightGBM

LightGBM is a gradient boosting algorithm that has shown to be performant while requiring reduced compute capabilities when compared to other gradient boosting algorithms [100]. Developed by Microsoft, the LightGBM framework supports many different Gradient Boosting algorithms as well as Random Forests. This model utilizes Gradient Boosting and a Decision-Tree Classifier where the parameters are weighted proportionally to their feature importance during training. Figure 3.10.a shows the flowchart diagram for the training and testing process. In this model training and test sets were created by removing samples that did not contain one of the specified classes or empty classifiers and randomly selecting data sets for each vessel uniquely identified by its MMSI number in a 70-30% training to test set split. The ML Dataset consists of Voyage Reports and Position Reports based on the AIS message received. The Voyage Reports for each vessel are then aggregated by Time-series into a set called Trajectories, which is also used for sampling, as shown in figure. Figure 3.12 shows the testing and training process utilizing the LightGBM algorithm to perform classification.

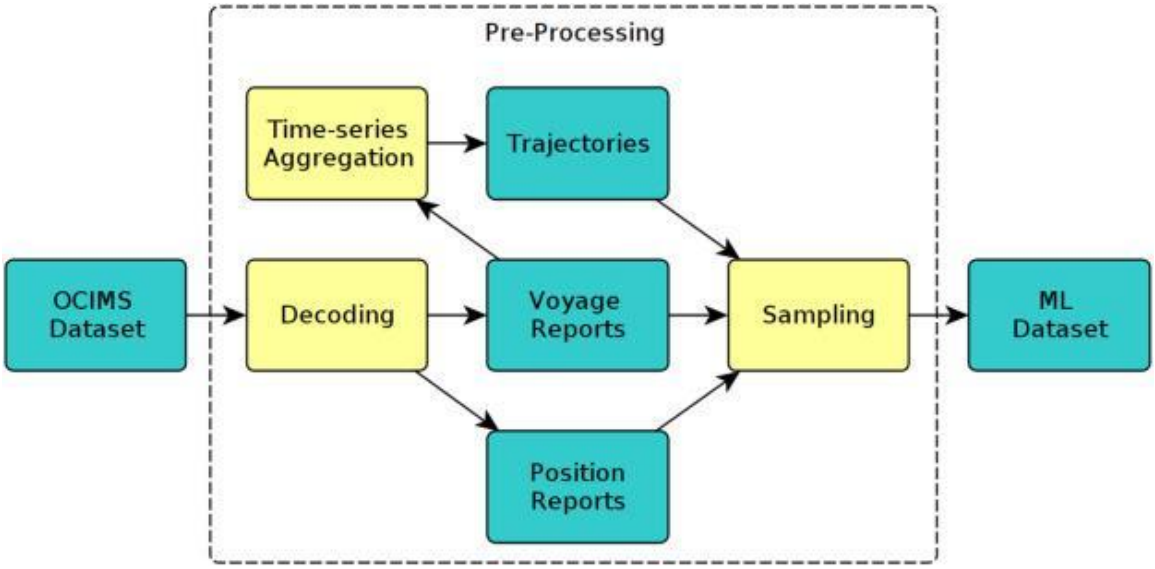


Figure 3.11 *- Dataset Preprocessing*

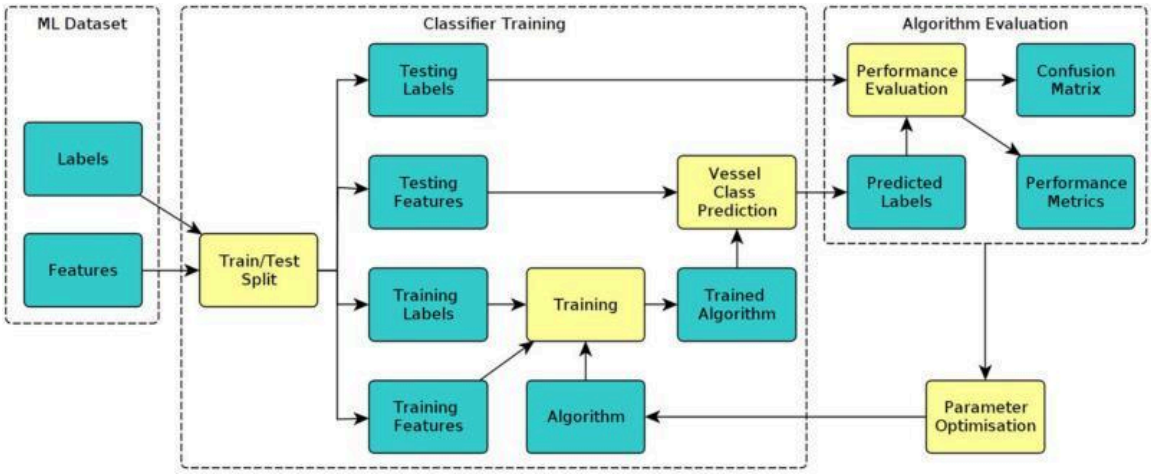


Figure 3.12 - *LightGBM Machine Learning Flowchart*

This model classifies vessels into seven categories, Cargo Ships, Tankers, Fishing Boats, Port Vessels, Passenger Vessels, and Recreational Vessels. However, to compare with the paper on MFELCM, the model was also rerun with the reduced classification set utilizing only the Cargo, Tanker, Fishing, and Passenger AIS classes. As shown in Figure 3.13, the quantity of the samples in each sample set are higher for Cargo and Tanker classes with relatively few samples for Recreational Vessels. These disproportionate sample sizes may contribute heavily to the results of classification.

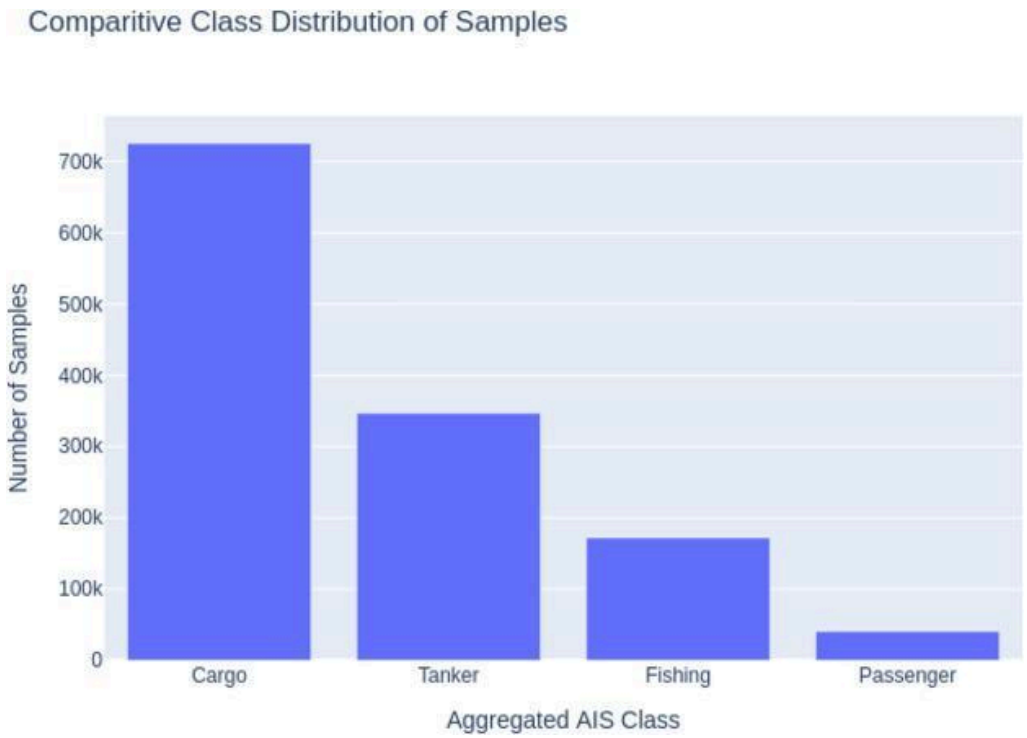


Figure 3.13 *- LightGBM Classification Results Bar Chart*

As displayed in Figure 3.15, LightGBM's ability to classify results is less accurate than that of MFELCM. Both utilize an ensemble of machine learning techniques and attempt to parse the dynamic data in a similar manner. LightGBM preprocessing the Voyage Reports into Trajectory reports allows the dynamic data, such as the vessel's current latitude, longitude, speed over course, course over ground, etc., be aggregated into a time series and smoothed in order to reduce the noise in the data. This model elects to smooth the dynamic data by dividing the Time-Series aggregations into 1-hour segments. This would ideally help smooth a vessel's trajectory (i.e. its path calculated based on a series of messages received over a period of time) by expanding the length of time to create an average of the vessel's position, in turn minimizing the minor errors of the dynamic data, such as slightly miscalculated speed over course or location by the GPS, and providing the ability to discard noise such as long periods between AIS message transmission.

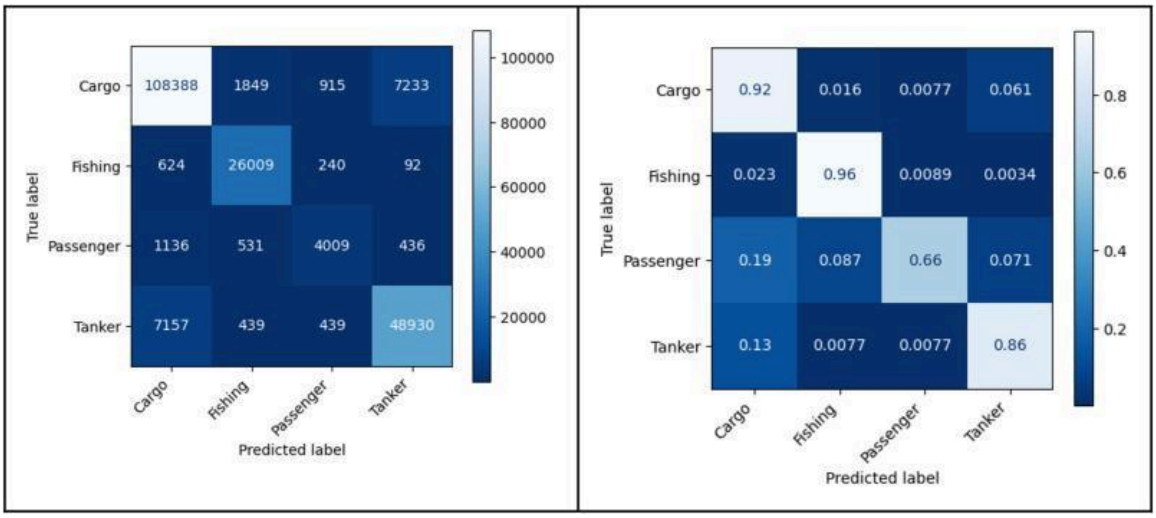


Figure 3.14 *- LightGBM Classification Confusion Matrices*

	Literature Results		Comparative Results	
Class	**F1 Score**	**Unique Ships**	**F1 Score**	**Unique Ships**
Passenger	0.8889	189	0.66	149
Tanker	0.9322	738	0.93	1539
Fishing	0.9355	674	0.86	575
Cargo	0.875	2161	0.92	3283
Weighted Average F1 Score	0.9		0.89	

Figure 3.15 *- Comparison of MFELCM (Literature) and LightGBM (Comparative) Performance Metrics*

As seen in Figure 3.15, LightGBM's predictive accuracy and f1-scores do remain high for Cargo Ships, Tankers, and Fishing vessels. Compared to MFELCM, the LightGBM model's predictive capabilities are high for Cargo Ships. The LightGBM model fails to provide quality predictive accuracy in Passenger vessels with an F1-score of 0.66.

In both models, the largest feature importance are the features that correspond to the vessel's size. It may be expected that the predictive error between passenger vessels and fishing vessels, given that they are more similar in size, would be correlated. However, the LightGBM model's confusion matrix, Figure 3.14, shows that Passenger and Fishing vessels are more likely to be falsely classified as Cargo vessels. Contrasted with the large sample size of Cargo Ships and relatively small sample size of Passenger and Fishing vessels. The LightGBM model's results are skewed proportionally to the size of each class's sample set. MFELCM's dynamic sample data is more evenly distributed among each class. Therefore, it is likely that the small sample size of AIS messages from Passenger ships is lowering the models accuracy on that particular class.

3.12 Anomaly Detection Algorithms

Classification algorithms using AIS data are useful in categorizing the behaviors of a Maritime Vessel and can provide a base for determining what is normal behavior. However, our MDAS system detects vessels that do and not transmit AIS using a radar detected system. Therefore, we need to be able to perform anomaly detection on the detected vessels. The dynamic data received either from AIS transmission or by the mounted radar system can be used to categorize a vessel using different classification techniques, as discussed in the previous section. Anomaly detection can utilize the categorization of a vessel in order to detect unusual activities such as speed deviations, unexpected stops, or unusual routes. While not all unusual activity correlates to illicit activity, many cases of unusual behavior can be the result of the ever-fluctuating weather and ocean-surface conditions, we can utilize anomaly detection to create a flag for law enforcement or authorities to investigate physically, reducing the amount of physical manpower normally required by augmenting it with a computer-assisted system. We discuss a methodology for AIS anomaly detection: Traffic Route Extraction and Anomaly Detection (TREAD).

3.12.1 Traffic Route Extraction and Anomaly Detection (TREAD)

The Traffic Route Extraction and Anomaly Detection (TREAD) methodology "aims to convert the large amount of AIS data into decision support elements, independently of the number of receivers, their performance, the platform of origin and the scale of the area of interest." [96] Maritime vessel traffic can be contextualized to provide useful awareness of regular maritime routes and seasonal conditions as well as high probabilities of risk in a given area. This information can be used to both detect anomalous behavior in daily traffic a priori and simulate regular traffic patterns by different classifications of vessels. TREAD performs this anomaly detection by utilizing machine learning techniques such as Density-Based Spatial Clustering of Applications with Noise (DBSCAN), Kernel Density Estimation, and similar data aggregation of vessel data by separating each vessel, defined by its unique MMSI, into a Time-Series trajectory comprising of its dynamic AIS information.

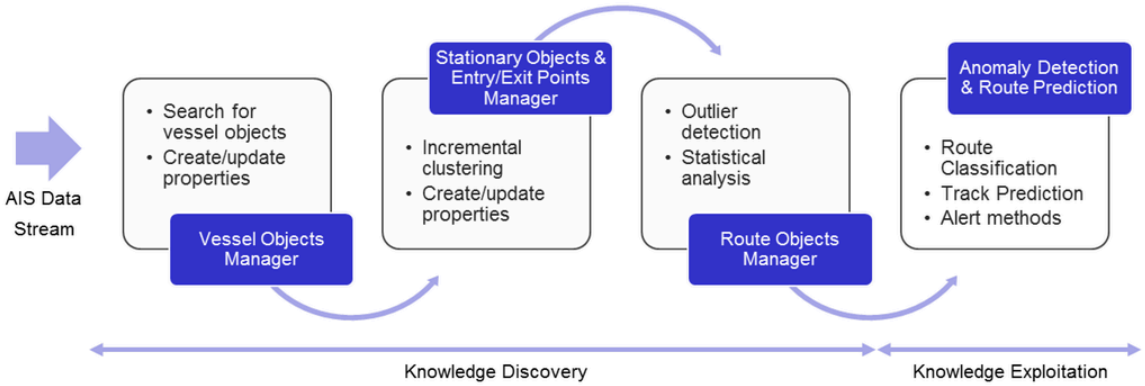


Figure 3.16 - *TREAD Process Flowchart*

TREAD calculates a "normal" path by using a "vectorial" representation of traffic. Each vessel's path is a straight path connected by a series of "waypoints". Intermediary points,

turning-points, are also introduced in locations where the changes in vessels' Course over Ground (COG) is consistent. Figure 3.17 (a) displays color-coded "normal" routes. Distinguishing between normal routes for different classifications of vessels using a vectorized approach requires significant algorithmic complexity. For this application, to perform the clustering of routes, DBSCAN is used. As explained in section 3.10.4 DBSCAN identifies and groups neighboring clusters and marks each point as connected or noise. By grouping the related dynamic data of vessels and filtering the noise in the data, TREAD can determine a regular path for a specific vessel type. Since AIS messages receival time frequently varies, often for legitimate reasons, TREAD utilizes Dynamic Time Warping, which shifts trajectories arbitrarily in time. Doing so allows the TREAD model to manage the infrequency of AIS transmission.

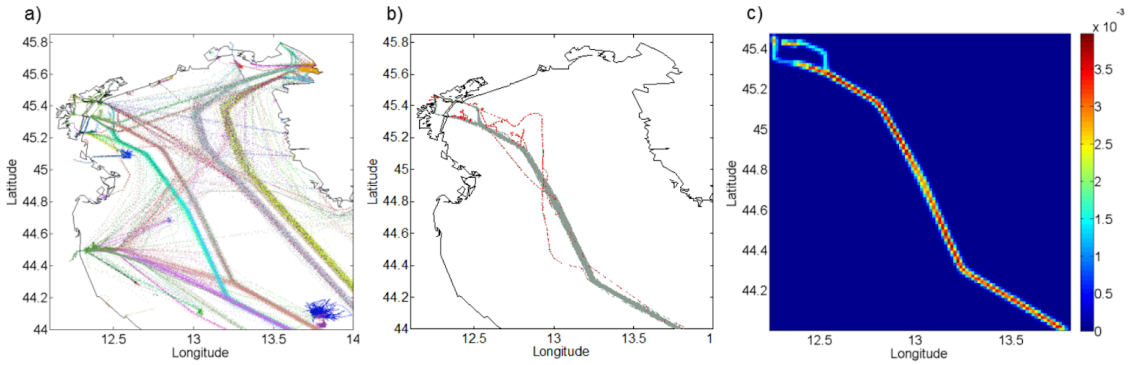


Figure 3.17 - *(a) Color-coded routes (b) showing potential outliers (red) (c) Kernel Density Estimation*

TREAD also utilizes an unsupervised, on-line learning model. Since data does not need to be a priori, as AIS transmissions are received the TREAD model can simulate the path of a vessel to predict anomalous behavior. As well stated in the paper, "the traffic pattern knowledge discovery process can therefore be linked to the notion of entropy, which measures the disorder in a system." If the "normal" path is defined, anomalous behavior can be detected by detecting trajectories outside the norm. Figure 3.17 (b) shows, in red, potential paths for vessels that would be marked as outliers. Should a vessel be detected on that route near the normal path, in green and shown in Figure 3.16 (c), it could be flagged as potential anomalous behavior. Similarly, a vessel following its normal path could make a series of turns, as demonstrated in Figure 3.18, inside the density estimation of its path and be flagged as anomalous behavior. TREAD characterizes anomalous behavior by creating a probability estimation that a vessel performs certain vectorized maneuvers based on vessels historically performing similar maneuvers. By defining a probability threshold, anomalous behavior can be flagged. In Figure 3.18, the red dots identify the anomalous behavior returning to blue as more probable maneuvers are detected.

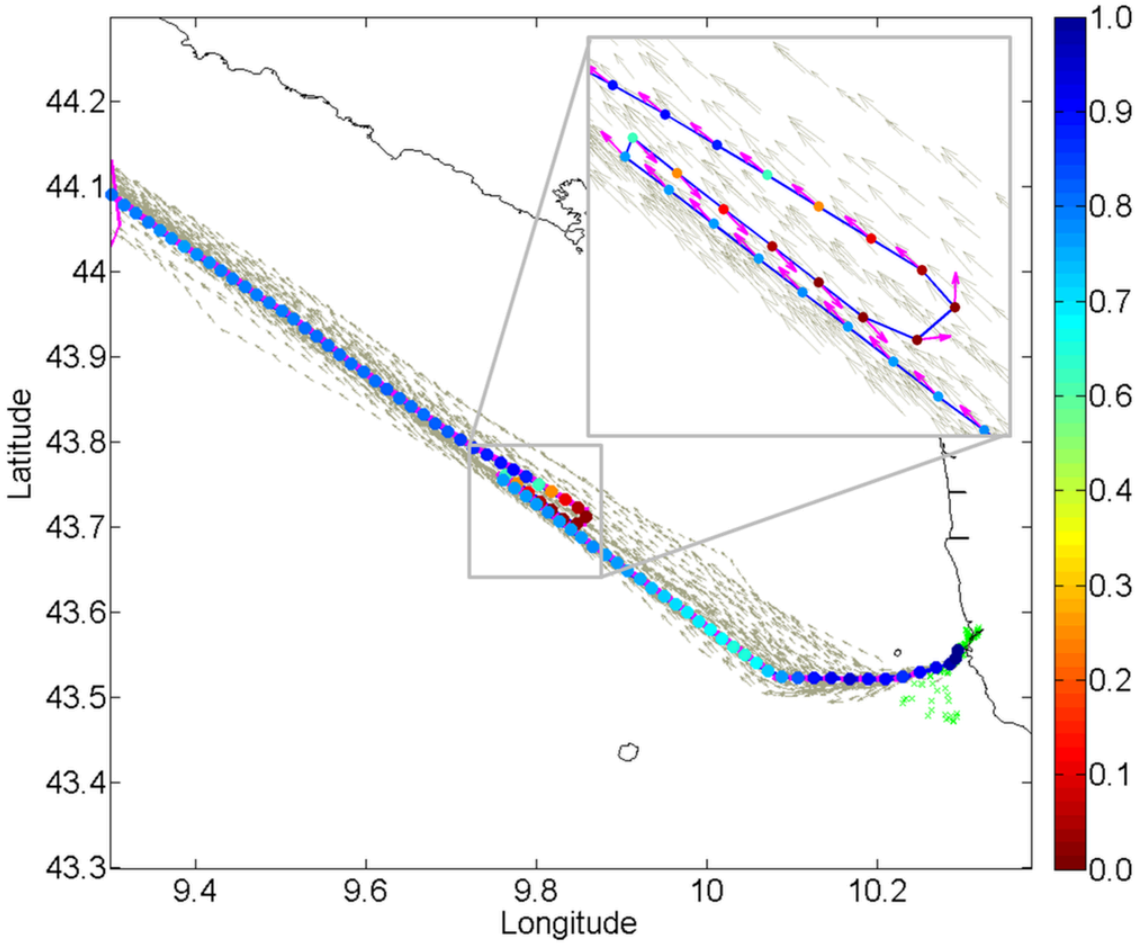


Figure 3.18 - *TREAD Probability of Anomalous Behavior*

The TREAD methodology provides a robust system utilizing an entropy-based decision making system, an unsupervised system that can be adapted to data dynamically with undefined ground-truth, and the ability to determine and locate anomalous behavior using historically normal paths and future path simulation methods of maritime vessels.

Chapter 4: Standards and Design Constraints

Engineering systems operating in real-world environments have to perform extremely well but do so safely, reliably, and in compliance with industry standards. For maritime systems such as the MDAS, this responsibility is no different. The MDAS must be capable of operating in rapidly changing environmental conditions, minimizing any impact on host vessel operations, maintaining secure and reliable communications, and complying with a broad range of regulatory domains. These include radio frequency emissions, electromagnetic radiation safety, equipment interaction, and long-term environmental durability.

Standards are required as they establish the exact rules and detailed specifications that are significant in ensuring compatibility among various types of equipment, safe usage, and adherence to legal requirements. Design constraints, however, are the physical limitations within which the engineering team has to operate successfully. These include a variety of factors, but are not limited to limited costs, weight, size requirements, and power, as well as environmental concerns and safety issues.

Within the MDAS program, there is the need to ensure compliance with relevant standards and recognize corresponding limitations in order to ensure that the system performs well, meets the maritime installation specifications, supports potential scalability, and finds acceptance by the regulators and stakeholders such as NOAA and government agencies.

4.1 Standards

Standards form the pillars that govern the development of complex systems such as the MDAS by outlining the specifications, criteria, and optimum practices that ensure the safety, synchronization, and lawfulness of systems in varied environments and applications. Standardized components enable the linking of innovation and compatibility by describing the performance standards that the equipment must meet, the transmission protocols for the transmission of data that manage the communication between systems, and the interfaces between components that cross multiple platforms.

Several bodies, such as the IEEE, FCC, IEC, IMO, and ITU, define them as the functional standards that govern the implementation of technologies like radar, wireless communications, and satellite transmission in different industries and global borders. The standards ensure that devices like MDAS function well on ships without interfering with the operation of installed navigation systems, communication equipment, and safety equipment.

Besides meeting technical requirements, standards are essential to safeguard the well-being of users and the general public. These standards often include limits for power output, radiation exposure, resistance to environmental stresses, and stability of operation. Compliance with these regulations goes beyond legal conformity; it is essential to promote confidence in the system's ability to perform under adverse or uncertain conditions. For a system like the MDAS, which may operate independently in remote

locations or aboard commercial vessels, conformity to these standards is crucial to successful integration, acceptance, and continued use.

4.1.1 FCC 47 CFR Part 80 - Maritime Services and Radar Compliance

The Federal Communications Commission (FCC) is the U.S. government body responsible for overseeing the nation's electromagnetic spectrum, which encompasses radio and microwave frequency ranges used in commercial and governmental systems. Among its many responsibilities, the FCC regulates rules for maritime communication and navigation systems under Title 47 of the Code of Federal Regulations (CFR), Part 80. This section of the code establishes the legal requirements for marine communication services, such as those utilizing radar, AIS, VHF radios, and emergency position-indicating radio beacons (EPIRBs), all of which are widely utilized in the maritime sector for efficient and safe vessel operation.

The Maritime Domain Awareness System (MDAS) makes use of radar technology as a key element for the immediate identification and tracking of vessels and targets in the nearby marine traffic. As such, adherence to FCC 47 CFR Part 80 is not a procedural formality; rather, it forms the underlying basis for the lawful operation of the system, safety standards, and compatibility on U.S. vessels. The MDAS design entails the fitting of a Raymarine Cyclone radar unit operating on the X-band range of 9.3 GHz to 9.5 GHz, the specific band assigned for use in maritime radar in Part 80 requirements. The radar is pre-certified with FCC ID PJ5-953MPSSR and therefore meets technical specifications like frequency allocation, emission bandwidth, peak power, and pulse characteristics.

This standard not only prescribes the frequencies on which a marine radar system can operate, but also its permitted power output, the extent of its emission bandwidth, and how the radar must interact with other systems on a ship. These are not abstruse considerations, each one has influenced particular engineering choices in terms of antenna choice, spatial location relative to other communications modules, shielding mechanisms, power management, and signal filtering within the context of MDAS. For example, due to the demands laid out in Part 80, under which radar emissions must be confined to a particular spectral band, caution must be exercised during integration so that the radar either overlaps or interferes with adjacent modules. These include things like Starlink Mini terminals, GPS receivers, or Wi-Fi access points, all of which might be operating in nearby bands.

Equally important, FCC Part 80 sets certification standards for radar equipment used on vessels. Any radar product that lacks certification to this standard is not allowed to operate legally in the United States. waters, particularly on government research vessels like those operated by NOAA, who have had early interest in the MDAS platform. This consideration alone drove the choice to utilize pre-certified COTS hardware over custom radar development, which would have entailed a long and costly certification process.

Further, the FCC Part 80 standard specifies the guidelines for installation and operating procedures. For MDAS, this calls for the radar to be installed in a manner that will prevent interference with other sensitive electronic devices, along with having all RF cabling correctly routed to avoid coupling or leakage. The rule also requires that

procedures for startup and shutdown be secured so that signal flooding or transients emitted that could affect onboard navigation or communications systems are minimized. Moreover, the MDAS must also follow emission duty-cycle limits, which govern how frequently the radar would transmit and the duration of individual pulses—factors directly related to power utilization, thermal capability, and image quality in the radar.

It quickly became apparent through initial project briefs with the technical consultants and with NOAA stakeholders that compliance with purely FCC-sanctioned maritime standards was a requirement if prototypes were ever to appear on real maritime vehicles. This not only guarantees the lawful usability of MDAS but also provides room for follow-on evolutions to more sophisticated deployment setups, e.g., integration with port authorities, the Coast Guard, or even utilization of multi-sensor fusion systems on board autonomous surface vehicles (ASVs) or uncrewed surface vessels (USVs). All of these prospective areas of utilization fall under the same Part 80 standard, further indicating its significance.

In conclusion, FCC 47 CFR Part 80 is not merely a licensing requirement but is, in fact, a holistic technical, structural, and strategic framework that has had far-reaching influence on the MDAS project in its various phases, ranging from hardware selection through software control protocols, system-level electromagnetic compatibility, to future deployment planning. By aligning its design with the specifications of this regulation, the MDAS project guarantees that it is legally, safely, and functionally capable of providing vessel detection services in the actual environments it was designed for.

4.1.2 IEEE C95.1-2019 - Electromagnetic Radiation Safety

The Institute of Electrical and Electronics Engineers, commonly referred to as IEEE, is among the most well-known organizations in the world. The premier organization has a key role in the development and publication of essential standards that ensure safety, facilitate interoperability, and enhance performance across a wide variety of technology fields. One such standard that has seen the most extensive use and implementation throughout the industry goes by the designation IEEE C95.1-2019, with the official title being "Standard for Safety Levels with Respect to Human Exposure to Electric, Magnetic, and Electromagnetic Fields, 0 Hz to 300 GHz." The primary objective of this important standard is to provide comprehensive protections for the general public as well as for workers engaged in professions where exposure to such fields may occur, protecting them from the potential biological effects that may arise from exposure to electromagnetic fields. This includes those fields that are generated by high-power systems, more specifically radar systems, which are designed to emit high amounts of electromagnetic radiation. Since the Maritime Domain Awareness System (MDAS) is using a commercial radar that works in the X-band frequency range of 9.3 GHz to 9.5 GHz, it is clear that the relevance of this standard is not just timely but absolutely essential under the present circumstances.

The basic aim of IEEE C95.1-2019 is to set and specifically define the maximum permissible exposure (MPE) levels that are acceptable over a broad range of frequencies along with various use cases. These MPE values are given in terms of electric field strength in volts per meter (V/m), magnetic field strength in amperes per meter (A/m),

and power density in watts per square meter (W/m²). These are arrived at through a comprehensive study and analysis from scientific and medical research, and these are focused on major points such as tissue heating, nerve stimulation, and hazards from prolonged exposure to such fields.

Specifically, for X-band radar frequencies, the power density limit is set at 50 W/m² for occupational exposure and 10 W/m² for general public exposure, averaged over a six-minute time period. These are set to avoid having personnel or bystanders in the area of operating systems exposed to hazardous levels of non-ionizing radiation, either in open-air environments or in enclosed structures where reflections and accumulation can occur. In the MDAS application, this standard has a serious role in influencing not just the physical design, but also the procedures for installation and the overall control system architecture. One of the earliest challenges that was planned out and taken into account was the physical location of the radar, more specifically in terms of areas that are within human reach. Unlike commercial vessels that mount radar arrays tens of feet above deck level, MDAS is intended for mounting on small to medium-sized platforms, e.g., research vessels, patrol craft, and commercial vessels. In most of these cases, the radar can be mounted only 5 to 10 feet above deck level, putting the emission zone in potential proximity to individuals.

In order to successfully satisfy this particular requirement, the MDAS system incorporates a variety of features that have been directly derived from the best practices established in IEEE C95.1-2019 guidelines. To begin with, the system effectively employs a fan-beam radar antenna pattern, which is specifically designed to project energy mainly within the horizontal plane. This intentional design choice significantly minimizes any vertical dispersion of the energy, thereby greatly reducing the likelihood of any exposure at ground level. Along with the essential aspect of physical placement, it is interesting to note that the MDAS radar system also includes an innovative component known as the "Standby" mode. This particular mode is software-based but serves a very significant purpose: it effectively shuts down radio frequency emissions under particular circumstances such as maintenance, docking procedures, or when personnel are known to be present in very close proximity to the antenna itself. This essential function is addressed with accuracy, being managed both locally at the site and remotely via the system's microcontroller unit (MCU). By incorporating this functionality directly within the system firmware, the design as a whole greatly reduces the risk of emitting during docking or when maintenance is being done to other hardware on the pilothouse. Along with this, it should be mentioned that the startup sequence of the system is expressly designed to default to a non-radiating state, which works to ensure a safe default configuration right at power-up.

The other important aspect that can be understood from the standards set in IEEE C95.1-2019 is the duty cycle, along with the averaging time that is utilized in the radar system's emission profile. While pulsed radar systems, such as the Raymarine Cyclone, carry low average power levels since each pulse is brief and transient, their ability to transmit high peak power can constitute significant short-term hazards in instances where personnel unknowingly enter the path of the beam. The MDAS, or Marine Dynamic Antenna System, thus employs refined time-based emission control techniques and

sophisticated feedback systems to diligently track and adjust pulse rates in such a way that the time-averaged exposure of individuals is always maintained well beneath the safety threshold, even under more demanding edge case scenarios such as when low-speed scanning missions are being conducted or during diagnostic test procedures.

In short, the incorporation of IEEE C95.1-2019 in the design process of MDAS prioritizes electromagnetic radiation safety as a primary imperative instead of an add-on requirement. By including important features such as controlled emission profiles, non-radiating states, physical mounting design considerations the MDAS system clearly demonstrates adherence to current safety standards in the industry. The part selection choices made not only maintain the maximum allowable exposure levels for both occupational and general environments but also instill confidence in stakeholders and end-users on the operation of the system with a high level of safety and reliability. As the MDAS progresses to full implementation, ongoing adherence to IEEE C95.1-2019 will be significant in maintaining its rightful utilization in various maritime environments.

4.1.3 IEEE 1547-Inspired Considerations for Off-Grid Power Integration Using Wind Turbines

IEEE 1547, titled "Standard for Interconnection and Interoperability of Distributed Energy Resources with Associated Electric Power Systems Interfaces", provides a standard for the integration of distributed energy systems with the power grid. While the MDAS platform is not interconnected with any conventional utility grid, the general principles of this standard provide valuable guidance when developing autonomous, off-grid systems. They assist in providing safe operation, stable voltage regulation, reliable energy delivery, and limited electromagnetic interference.

Though IEEE 1547 addresses grid-connected systems explicitly, its best practices can be modified to guide the integration of wind energy systems into isolated, remote-use applications. For an ocean-based installation such as MDAS utilizing a vertical-axis wind turbine, the system needs to provide power quality, control logic, and safety protections over varying weather and load conditions.

As MDAS is built nearly entirely out of commercial off-the-shelf (COTS) hardware, much of the electrical infrastructure already meets expected performance and interoperability standards. By using pre-regulated DC-DC converters, marine-rated battery banks, and wind-oriented charge controllers, for example, both simplifies the power delivery architecture and reduces the need for excessive custom validation or compliance testing. However, a number of considerations still require judicious design and configuration to ensure safe and continuous operation in accordance with 1547-inspired standards. To meet these expectations, the system must support the following features:

- Voltage and load regulation: Utilizing multi-stage DC-DC regulators assists in supplying power to every component in a narrow voltage range. This guards against under-voltage and over-voltage conditions that will ruin the radar, camera systems, or onboard computer units. These regulators, being COTS modules, are

typically pre-qualified for marine or embedded applications, with little need for custom engineering.

- Safe power sequencing: Sensitive electronics like network modules and radar transceivers require controlled startup and shutdown procedures. One can use microcontroller-based logic that interrogates battery state-of-charge and system readiness before enabling high-power components. Most embedded microcontrollers and COTS-based charge controllers already have such features natively supported.

- Electromagnetic interference minimization: Communications and radar equipment near one another need to be shielded against noise coupling. This is normally accomplished with shielded enclosures, grounded busbars, and low-noise DC regulators. With the MDAS dependence on certified radar and structured wiring harnesses, the majority of this shielding is inherent in the system design.

- Operational isolation: The system must be galvanically isolated from the host vessel to prevent unintentional coupling with onboard power systems or corrosion due to stray currents. for a deployable unit that is meant to be independent. Employing wind turbines and batteries that are off-grid marine applications can make this requirement easily fulfilled with little modification.

Although the MDAS platform does not necessarily have to meet IEEE 1547 certification, the platform's design embodies the spirit of the standard. With the use of modular, proven COTS components and the implementation of standard safety and power integrity practices, the system meets the necessary reliability for continued remote operation. These adaptations also make it easy to scale over the long term, simple to deploy on many ships, and make regulatory authorities confident in the environment where predictable power quality and electrical safety are the top priorities.

4.1.4 Code of Federal Regulations - Automatic Identification System

The Automatic Identification System (AIS) is a ship tracking and anti-collision maritime safety system, officially defined in 33 CFR §164.46 as the technical specifications, operational parameters, and equipment for commercial ships operating in U.S. waters. Internationally cooperative efforts developed AIS, adhering to regulations specified by the International Maritime Organization (IMO) and the International Telecommunication Union (ITU), while in the United States, enforcement is carried out by the U.S. Coast Guard (USCG). It constitutes a building block that enables Maritime Domain Awareness (MDA) and is a vital constituent of the MDAS system's operational mission.

AIS requires most commercial vessels over 65 feet in length to transmit their position, identity, navigational status, and other voyage information on VHF radio channels. This data is transmitted continuously to shore-based receivers, as well as to other vessels and aircraft. Larger vessels and passenger ships must carry a USCG-approved type AIS Class

A device by the rule, while smaller commercial and fishing vessels may carry a Class B transceiver, with operational restrictions.

The standard defines what a “properly installed and operational” AIS unit is, using IMO circulars and NMEA installation standards to ensure installation quality and performance. AIS proper operation relies on valid and current MMSI numbers, correct navigation input sources (e.g., heading, GPS), and continuous transmission when underway or at anchor. Notably, ships are to maintain AIS transmission continuously except in situations of direct danger to vessel safety or security. The significance of this standard to the MDAS project is tremendous, since its implementation provides a two-tiered system of maritime visibility, compliant vessels are easily recognizable, and non-compliant or "dark" vessels can be filtered out systematically by algorithms. Due to their cooperative nature, AIS broadcasts enable filtering of radar data to eliminate known targets. This enables the MDAS software to concentrate its processing capability and alert systems on uncooperative vessels, whose identities or positions would otherwise remain unknown.

From a systems design point of view, this regulation impacts numerous architectural and software decisions. The AIS data stream, potentially derived from public databases or by direct signal interception, serves as a baseline reference for vessel detection through subtractive means. In addition, the hardware specifications delineated in the regulation dictate the compatibility requirements and establish expectations for any AIS-related hardware employed in the prototyping stages or complete implementation. Additionally, availability of AIS pilot plugs, adherence to portable AIS standards, and operational practices—such as logging idle time in the vessel log—are important factors for ships that are strong candidates to be MDAS hosts. Although MDAS does not emit AIS signals, its passive monitoring method relies heavily on this infrastructure to detect and classify cooperative maritime traffic in real-time.

In total, 33 CFR §164.46 regulates the conduct of the majority of commercial maritime traffic while, simultaneously, creating an inherent component of MDAS's core functionality: non-cooperative vessel detection and tracking through the lack of AIS transmissions. The regulation thereby serves as a critical reference point for the design, justification, and anticipated performance of the MDAS detection algorithm.

4.1.5 Waterproofing and Ingress Protection

The MDAS must remain operational in unpredictable conditions on the ocean where moisture, spray, and even temporary submersion are likely. We will cover general environmental factors like humidity and corrosion, but there is a quantifiable value called ingress protection (IP), which are requirements to keep onboard electronics safe.

There is a waterproofing standard for applications like this called IP67, defined by the International Electrotechnical Commission. An IP67-rated enclosure ensures complete protection against dust and allows immersion in up to 1 meter of water for up to 30 minutes. For subsystems such as the radar processor, camera control module, and embedded computing unit, this rating establishes a minimum requirement. The IP65 standard covers splash or spray while IP67 offers extra security and aligns with military practices for expeditionary hardware.

To meet IP67 standards your design must include gaskets for sealing enclosures, cable glands with compression seals, waterproof connectors, and conformal coating for PCBs. It can also be crucial to add vents to avoid internal condensation during temperature fluctuations.

Our system will face plenty of salt fog and water jets during its lifetime. For those reasons we plan to meet or exceed IP67 certification to prevent ingress and avoid electrical failure.

4.2 Constraints

The development of the Maritime Domain Awareness System (MDAS) is not only driven by technological advances and operational demands but also by a set of pragmatic limitations previously identified as design constraints. These are caused by various factors: economic limitations, environmental factors, availability of power, and available technologies.

In contrast to standards, which establish the regulatory or safety standards that a design must satisfy, constraints are project-specific problems or limitations arising from stakeholder requirements, resource constraints, and technical considerations. Constraints are not inherently bad but are actually critical parameters that compel the creation of a viable and workable system. The MDAS project, being charged with delivering an economically feasible, autonomous, and deployable marine detection and monitoring platform, is confronted with numerous such constraints. Three significant limitations are discussed in the sections that follow:

4.2.1 Environmental Factor Constraints

Environmental factors place a number of critical restrictions on the development and deployment of the MDAS unit. Working in marine environments is far more challenging than in inland or controlled environments. The system should be capable of performing even after being exposed to continuous salt spray, high humidity, intense solar radiation, harsh winds, and fluctuating temperatures which are conditions that pose issues for both the structural and operational longevity of the unit. These environmental conditions provide harsh constraints on the types of materials, mechanical designs, and electronic components that can be used in the system, because only certain arrangements are capable of withstanding prolonged exposure in harsh marine environments.

A significant challenge is exposure of the system to water. Constantly surrounded by water, the high humidity and salt filled air contribute to corrosion and degradation of structural components and electrical contacts over time. This places a critical restriction on the MDAS to incorporate waterproofing techniques and tightly seal all enclosures so that water cannot penetrate. Electrical failure due to corrosion and short-circuiting could endanger the functionality of important systems such as the radar or communication modules. Because of this, we have to be more prepared in identifying, managing and reducing the possibility of water leaks by properly enclosing the design.

Another challenge the system may face is from temperature fluctuations out at sea. This

can introduce stress on mechanical connections, housing parts, and internal hardware. Thermal cycling from day to night can cause the loosening of fasteners, weakening of adhesives or mechanical fatigue leading to a restraint on how parts are constructed together. Additionally, the system must withstand long term exposure to ultraviolet (UV) radiation, which causes materials like plastics and coatings to degrade and ultimately develop surface cracking, brittleness and discoloration. Thus another constraint is added further reducing to the range of materials appropriate for the MDAS configuration and introduces other needs to combat it. Leading to the discussion of protective finishes, UV resistant coatings and a robust mechanical design.

Wind exposure presents another major constraint, particularly since the MDAS features a vertical-axis wind turbine. While the idea of having as much wind as possible seems like a benefit for an off-grid power generating system, it also introduces aerodynamic loads for the structure to face. Harsh winds can impose torque on the wind turbine which in turn induces forces, vibrations and extra pressure upon the mounting implementation. All of these need consideration within the system engineering design because effectively mounting the equipment and reducing the factors that can negatively impact it greatly benefit the accuracy of data acquired and improve the life expectancy of moving components.

4.2.2 Weight and Form Factor Constraints

Another key constraint affecting the MDAS system is the limitation on its overall weight and physical dimensions. From the beginning, the system was introduced to be designed and installed without requiring heavy equipment or permanent structural modifications to the respective vessel it would reside on. This placed a constraint on weight and form factor as the system had to offer portability and be easy to manually set up. To achieve this goal, the weight and form factor was limited as it needed to be easily transportable, and mountable with a limited number of people with basic tools making the system suitable for deployment in a wide variety of maritime environments.

The weight constraint directly influences component selection and structural design. More dense materials, although generally much stronger, had to be avoided or used sparingly so as to not design a structure too difficult to transport. Therefore, each design section, made up of the outer open structure frame, radar, vertical wind turbine, power storage and electronics enclosure had to be designed with a focus on reducing weight while also trying to be relatively stable. Any extra weight would compromise the structural strength of the unit once mounted on surfaces like the boathouse rooftops or mass-like structures.

In addition to the weight constraint, we also experienced a size/form factor constraint as well. The size of our system was highly dictated by the size of radar we selected. Choosing a large radar meant that the system needed to fully support the physical length and rotational clearance of the radar array without compromising on mechanical stability. This decision not only affected the dimensions of the MDAS frame but also the size and placements for the other components like the turbine, electronics, enclosure and especially the mounting structure.

Furthermore, the form factor imposed more constraints on the systems deployment methods and locations. It required a flat open area with enough clearance for the radar and turbine to function without obstruction. Having a large radar can rule out many mounting options like on masts or railings which wouldn't be able to operate due to obstacles. Certain mounting options would need to accommodate it by evenly spreading out weight and ensuring solid anchoring to the surface. On the other hand, for smaller radars, many different mounting options are available and wouldn't require much anchoring strength due to a smaller form factor of the design. Masts and railing would be an available option.

As a result, the system had to be carefully designed to ensure no part imposed more restrictions on another. The weight and form factor constraints led to more limitations on the design than expected to retain the goals of the MDAS project in terms of portability and simplicity of deployment.

4.2.3 Raspberry Pi Constraints

Another technical constraint in our project is how the Raspberry Pi will manage communication with several onboard hardware components and sensors that make up our Maritime Domain Awareness system. The Raspberry Pi board we are using has four USB ports and a 40-pin GPIO header. These interfaces will allow us to handle both incoming and outgoing data between the embedded system and connected devices like the radar, camera/gimbal system, GPS module, and power management components.

One of the main challenges with the interfaces is being able to efficiently allocate the limited GPIO and serial ports that are available. The Raspberry Pi will be able to receive data from the radar through the USB interface, while simultaneously controlling the gimbal system, which will require PWM or UART signals. We also need to manage communication with onboard sensors and modules via I2C or SPI. Since there is a limited number of pins available for each communication protocol, there is a constraint on how many components can be connected directly.

Another constraint to consider is how the system will transmit data back to the command center. This will be done via Starlink through a USB-based connection which will occupy yet another one of the available USB ports. We also need to monitor power consumption on the Raspberry Pi, since the system must remain operational in a remote environment where power is limited and supplied through a renewable source.

4.2.4 Time Constraints

The MDAS project operates within a compressed academic schedule, divided into two formal phases: Senior Design I during the Spring semester and Senior Design II in the Summer semester. Under typical timelines, prototype development and system integration span both semesters. However, the timeline for this project is notably accelerated due to sponsor expectations. Our sponsor, SOUTHCOM, has communicated a desire for a near-finalized system by the end of Senior Design I, rather than at the conclusion of the full design cycle. This requirement places additional strain on the team, as it compresses development, procurement, testing, and validation activities into a

limited timeframe. One of the most significant challenges resulting from this elevated timeline is procurement delay. All component purchases must be routed through the sponsor and associated budget personnel, requiring formal approval prior to any order. This introduces unavoidable latency, particularly problematic when a critical component, such as the radar system, requires specialized adapters or cables not initially identified. These procurement bottlenecks can create cascading delays, impeding the ability to iterate or address hardware integration in a timely fashion. As a result, time management must account not only for technical development, but also for procedural lag within the funding and approval pipeline. Delivering a functional prototype within these conditions requires careful planning, frequent sponsor communication, and parallel development efforts wherever possible.

4.2.5 Budget Constraints

The MDAS project is conducted under a constrained but substantial budget, with an allocation of approximately $15,000 provided through sponsor support. While this figure is high in comparison to standard university projects, it must be emphasized that maritime radar systems, embedded computing modules, and off-grid power systems inherently carry significant cost. A substantial portion of the budget, roughly $8,500, was dedicated to the acquisition of the Raymarine Cyclone 6 ft open array radar, selected for its optimal tradeoff between performance, integration flexibility, and energy efficiency. This single purchase accounts for over half the project’s financial resources, leaving the remaining funds to cover power storage, wind generation components, enclosure materials, computing hardware, optical enhancements, and network transmission equipment. All purchases must be coordinated through the sponsor and processed via an internal approval system. This limits the team's ability to procure last-minute items and reinforces the need for highly accurate initial part selection. Any unplanned expenses or incorrect orders risk causing substantial schedule slips or compromising system functionality. Moreover, because the system is being designed to be field-deployable and potentially productized, each component selected must balance reliability and environmental tolerance with cost. Tradeoffs are often made between commercial-grade and industrial-grade components to ensure long-term resilience without exceeding budget constraints. Maintaining system quality while operating under both cost and approval constraints has proven to be a central challenge in the engineering process.

4.2.6 Power Generation and Energy Storage Constraints

The maritime radar system's power demands present a significant constraint due to the high energy consumption of its core components. The Starlink Mini and Raymarine Cyclone Radar alone account for nearly 90% of the daily power budget, requiring 148Ah of the total 168Ah (12V system) or 74Ah of 84Ah (24V system) daily consumption. This heavy load necessitates an oversized battery bank—at least 210Ah at 12V or 105Ah at 24V—to maintain the recommended 80% depth-of-discharge limit for $LiFePO_4$ batteries. Even with proper sizing, the system would only achieve 1-1.5 days of autonomy without recharging, making consistent energy harvesting absolutely critical. The radar's intermittent 5A current spikes during scanning operations further complicate power stability, potentially triggering voltage fluctuations that could affect sensitive electronics.

A second major constraint stems from the system's reliance on renewable energy sources in an unpredictable marine environment. While a 400W wind turbine paired with 400W of solar panels could theoretically generate the required 2,016Wh daily, real-world conditions often yield significantly lower output. During extended periods of calm winds or overcast skies, the system risks entering an energy deficit that could lead to premature battery depletion. This vulnerability is exacerbated by the 10-15% efficiency losses in voltage conversion stages (VCM1000 for radar, DC-DC step-up for Starlink). The 24V architecture partially mitigates these issues by reducing current-related losses, but cannot eliminate the fundamental challenge of inconsistent energy harvesting. Careful power management, including load prioritization and potential backup sources, becomes essential for mission-critical operations.

The physical implementation of the power system introduces a third set of constraints, particularly regarding wiring and thermal management. In 12V configurations, the high current demands (peaking at 14A) require thick 10AWG wiring runs to minimize voltage drop, adding weight and complicating installation in tight marine spaces. Even at 24V, where current is halved, the system still requires robust busbars and properly rated connectors to handle sustained 7A loads with intermittent spikes. Thermal considerations are equally critical—the VCM1000 converter and Starlink's step-up regulator can generate substantial heat during continuous operation, potentially derating their performance in enclosed compartments. These physical constraints demand careful system layout with adequate ventilation, proper wire routing to avoid electromagnetic interference with radar signals, and strategic placement of fuses/breakers to protect against faults while avoiding nuisance trips during legitimate current spikes.

4.2.7 Lightning and Surge Protection

One of the design challenges for this system is protecting the electronics from lightning strikes and power surges. Since the radar system, wind turbine, and antenna components are mounted externally and placed near the top of the vessel, they are more likely to be affected by lightning or electrical discharges during storms. Even without a direct lightning strike, electrical surges from nearby strikes can still travel through connected cables and damage internal circuitry.

This introduces a constraint in how we design the wiring and enclosures, especially for power and data connections that come from the radar, wind turbine, and external sensors. We need to make sure that these connections have some level of electrical isolation and surge mitigation to prevent the rest of the system from being affected. It’s also important to think about how energy from unexpected spikes, like sudden voltage jumps in the power supply, might affect sensitive electronics like the Raspberry Pi or the gimbal controller.

A big part of this is going to be making smart choices about what kinds of connectors, fuses, and internal protections are used at key points in the system. Without surge protection, the entire embedded system could be put at risk from a single event, which is something we want to avoid, especially since these units are meant to be deployed for extended periods without on-site maintenance.

4.2.8 Cyber Security Constraints

Another important consideration in the design of our Maritime Domain Awareness system is cybersecurity. Even though this system isn't designed to be accessed from the open internet, it does use networked components like the onboard Raspberry Pi, Wi-Fi modules, and the Starlink internet connection that could potentially be targeted if not configured properly.

One of the challenges here is making sure that the system stays secure while still being functional. Since we need to send data from the embedded system to the command center over Starlink, the information passing between the two needs to be protected. The Raspberry Pi will also likely have services like SSH or VNC enabled for remote access and updates, and that opens the door to outside access if not handled carefully.

Another constraint is that we may not have a full-time IT or security team monitoring the system, so it has to be designed to be as secure and self-sufficient as possible. That means things like using strong credentials, disabling unnecessary services, and making sure software is kept up to date. There's also the risk of someone having physical access to the unit when it's in port or during maintenance, so we have to consider ways to limit what they can do even if they plug into the system directly.

Overall, it's something that has to be built into the design early on, because adding security features after the system is already up and running is a lot more difficult and less effective. Since this system could be used in sensitive areas or for tracking vessels that don't want to be seen, protecting the data and access points is a critical part of the design process.

Chapter 5: Comparison of ChatGPT with other Similar Platforms

With the rapid development of Artificial Intelligence (AI) and Large Language Models (LLM) in recent years, we have sought to transform a machine's ability to make decisions and learn as humans do. Machine Learning models have often been implemented in applications that require their decision-making capabilities in various real-world environments in a naturalistic and human-like manner. Now, with the ability to interact with humans as a chatbot, these LLMs have gained much attention with the assistance they provide everyday to not only technology professionals, but also the general public. This rapid rise of LLMs has had a profound impact with improving productivity and creativity, but it has also raised new challenges around accuracy, ethics, and over-reliance.

5.1 LLM Comparison

Large Language Models (LLMs) are deep-learning algorithms capable of "recognize[ing], summariz[ing], translate[ing], predict[ing], and generate[ing] content using very large datasets" [28]. Essentially, you submit a prompt (input) to an LLM and it predicts an appropriate response (output) based on patterns learned from all this text data. Modern LLMs are typically built on transformer architectures, which helps these models understand context and relationships between words very effectively. In fact, the 'T' in ChatGPT stands for transformer ("Chat Generative Pre-trained Transformer"). This technology has unlocked AI's ability to have very human-like conversations with its users.

These AI chatbots can carry conversations with users, answer questions, and assist in many tasks using natural language. They have become widely accessible and increasingly popular in recent years. They will continue to grow as we see them available as mobile apps and integrated in our phones as AI assistants (e.g. Samsung's Galaxy AI and Apple's Apple Intelligence). Users can ask for explanations of various concepts, get help with coding, or prompt creative outputs like poems, stories and even drawings and videos. In all these cases, the LLM crafts a relevant response using its trained knowledge.

While more and more people find this technology incredibly useful and beneficial, it is important to understand the drawbacks and limitations that are present. There is a phenomenon associated with generative AI known as "hallucination," meaning the models can sometimes produce false or misleading information. The critical point to remember here is that these LLMs do not truly know facts, they simply predict the most likely text. It is always a good idea to exercise caution and verify critical information when dealing with these models, as these tools can occasionally give answers that sound reasonable but are actually incorrect. As these AI assistants integrate further into our workflows, it's vital to understand the strengths and weaknesses of the various LLM platforms. Different platforms might excel at certain tasks, where some might be ideal for creative writing, while others are designed to assist in coding or for business productivity.

5.1.1 Case Studies - ChatGPT

5.1.1.1 Embedded Systems Perspective

Throughout the development of the project, ChatGPT was used as a tool for researching technologies, organizing findings, and drafting documentation to support the design of the system. It quickly became a reliable way to gather sources to conduct research and get any technical clarification, especially in areas unfamiliar to the team. One of the most beneficial applications came during the research phase on marine communication protocols, where we learned about NMEA 2000 and its role in allowing the radar system to interface with other marine devices.

At first it was difficult to understand NMEA 2000 when going through typical search engines since it typically led to overly technical whitepapers or high-level summaries. ChatGPT was prompted to explain NMEA 2000 in context, including how it differed from NMEA 0183 and how it relates to the CAN protocol. The responses provided a solid conceptual foundation. This made it easier to dive deeper into details like baud rate, network topology, and physical layer connections. At this step, it was important to cross-verify using manufacturer documentation and trusted technical blogs considering AI ‘hallucinations’ previously discussed. In most cases, ChatGPT offered information that was either accurate or very close to correct. It would even provide sources at times which made it quicker to comprehend complex subjects.

ChatGPT also supported the research process when evaluating options for a single board computer and programming languages that could work for the embedded system. The most convenient use at this time was to prompt it to compare boards such as the Raspberry Pi, Arduino Uno, and BeagleBone Black and their key features including processing power, memory, and available interfaces. Similarly, it was useful when comparing several programming languages by asking ChatGPT to describe their advantages and limitations in embedded applications. These results were later confirmed using official datasheets and vendor websites. While certain specifications were double-checked, the ChatGPT helped quickly narrow down the best options for both hardware and software choices.

5.1.1.2 Optical System Perspective

In addition to using ChatGPT for assisting with research on embedded systems, we also utilized this LLM for research on the optical components of the project. Specifically, we used ChatGPT to compare different types of camera sensors and lens systems to help narrow down which ones would be the best for our project specifications. By using ChatGPT for research it helped quickly compile comparison data, while we could specifically choose parameters we were looking for. There are many factors that all come into play when comparing optical technologies, and ChatGPT helped to easily display the vast amount of options in an easy to read format.

One thing that we needed to be cautious about was the validity of the options that were provided in the outputs. For a majority of the output options, any provided links did not result in valid products or reliable articles. Some of the links were to web pages that previously existed, while others led to nowhere in particular. While it is incredibly helpful for all of the outputs to be instantly presented, it is important to check the credibility of each piece of information to prevent incorrect information from being the foundation of our project. When beginning to research which camera sensor to use for the project, it was a bit overwhelming looking into all of the different models of both CCD and CMOS sensors. By using ChatGPT, it made it a bit easier to process all of the different options to determine what to research further.

In addition to providing comparisons of the camera sensors, we also used ChatGPT to provide basic comparison data of the different possible lens systems for the magnification portion of the project. The best use for this was generating a table with comparison data, which also provided equations for the basic working principles. By utilizing this, we were quickly able to narrow down and select the proper lens system, making sure to double check stated specifications.

5.1.2 Case Studies - Gemini

5.1.2.1 Radar Perspective

To evaluate Google Gemini's ability to assist with technical research, we posed the following open-ended question to it: identify the most significant radar features for the detection of vessels in a maritime environment. This included understanding environmental conditions such as sea clutter and weather, specifying radar capabilities necessary for long-range surveillance, and suggesting radar technologies that can be embedded with systems. Additionally, we asked for comparisons among existing commercial radar models as well as a recommendation deemed most suitable for offshore vessel detection application.

Gemini demonstrated a strong capability in comprehending the operational environment. It rightly deduced that radar systems for maritime environments have to factor in considerations such as high humidity, corrosion from saltwater, intense rainfall, and fluctuating wave interference. The model identified the requirement for Doppler processing, pulse compression, and remote configuration capability for tracking small, non-cooperative vessels. It also acknowledged that energy efficiency and system ruggedness are paramount for off-grid platforms such as ours, which are powered by wind and need to be capable of independent operation. With that said, to uncover more specific comparisons and technical depth, we found it necessary to refine our original prompt several times. Follow-up queries were used to push for details on radar types, architectural differences, and integration challenges that were not included in the initial response.

Gemini's strongest point was its emphasis on software integration. It pointed out SDK availability as a tie-breaker for choosing a radar, a point that most product listings overlook. Gemini listed several commercial radar systems and rightfully excluded those

lacking integration tools. Its final recommendation supported the same radar we selected independently, reinforcing our decision with a sound technical rationale.

While Gemini was highly informative, it also showed some shortcomings. It lacked complete instruction on hardware interfacing and failed to include links to SDK documentation. Some suggestions also seemed to ignore pragmatic factors like component availability, physical constraints, or budgetary issues. Prompt refinement again played a role here, as we had to request clarification and additional context when its answers lacked practical grounding or skipped over implementation details. These lackings made additional manual exploration necessary in order to fully evaluate the possibility of integration.

In spite of these limitations, Gemini was a great resource to early radar investigations and validated major selection criteria. It was particularly useful for formulating technical trade-offs and identifying viable radar options for naval surveillance missions. In general, it acted as a solid foundation for subsequent analysis and assisted engineering decision-making.

5.1.2.2 Material Section Perspective

Google Gemini was prompted to generate a list of building materials capable of designing a maritime structure that is corrosion resistant and supports the radar system while also providing space for a wind turbine alongside other electronic components. Gemini responded with a list of requirements the materials would need, these being waterproof and corrosion resistant which were exactly what was a repeat of what it was prompted. It also gave very vague suggestions on their implementations. We were suggested to only use stainless steel for the mounting components, it didn’t provide any specific materials for the structure at all. Due to the lack of options, more specific prompts were given to it which consisted of describing the different structural components, asking for a list of possible materials best suited for that structural component's task and providing the divide and conquer document to clarify the requirements of the project.

After these more specific prompts, Gemini responded with a list of material selections for each specific part of the system, the types of grades of material it recommends alongside some pros and cons of each material. For the frame it listed possible options of using aluminium alloys, stainless steel 316, and fiberglass reinforced polymers. For the top plate where the radar pedestal is mounted, Gemini suggested similar materials as the frame but included carbon fiber composites as well. The bottom plate was also mainly aluminum and steel. Lastly, the electronics enclosure mentioned some chemical and water resistant materials like high-density polyethylene.

Given the list of possible materials, their pros and cons and types of implementations. We considered these options as brainstormed ideas and researched more into them. We found many published reports that discussed the use of stainless steel, aluminum and fiberglass reinforced polymers in maritime environments each with their own purpose. Through this we discovered the different types of possible corrosion and hazards in maritime environments alongside ways to combat them such as pitting, UV resistance, and coatings

to prevent material deterioration. Overall, using the Gemini LLM, it was a great entry to learning all the different types of materials that are possible to use and how they can be implemented into the project's design.

5.1.3 Case Studies - Deep Seek

5.1.3.1 Turbine Perspective

Our team employed DeepSeek to conduct thorough research on wind turbine components for the maritime radar system. The initial queries about vertical versus horizontal axis turbines provided only basic comparisons that lacked marine-specific details. We quickly realized these general responses didn't address critical factors like saltwater corrosion resistance, maintenance requirements at sea, or spatial constraints on vessels.

To improve the results, we refined our prompting strategy to focus on specific aspects. We asked targeted questions about marine durability, power output verification at different wind speeds, potential radar interference, and mounting requirements. This approach yielded much more actionable data. For the TESUP Magnum HAWT, DeepSeek helped uncover that its advertised 10kW daily output was based on ideal inland conditions rather than marine environments. It also identified potential radar interference issues from the rotating blades that weren't clearly stated in the product specifications.

The research on the TESUP Atlas VAWT proved particularly valuable. DeepSeek provided detailed information about its 4 m/s startup wind speed and the availability of optional blade sets for different wind conditions. However, we noticed the AI couldn't verify manufacturer claims about long-term marine durability without additional sources, highlighting the need for supplemental research.

DeepSeek's ability to generate comparison tables between turbine types significantly aided our evaluation process. These tables clearly outlined differences in power outputs, maintenance needs, and spatial requirements, serving as excellent reference points. However, we discovered some inconsistencies between sources that required direct follow-up with manufacturers to resolve.

A key limitation emerged in the AI's occasional reliance on unverified manufacturer claims. When examining the flower-style Tulip VAWT's turbulence tolerance, for example, DeepSeek could only reference the academic paper we provided rather than offer independent analysis. This reinforced the importance of complimenting AI research with hands-on testing and expert consultation.

Ultimately, DeepSeek proved invaluable for accelerating our initial research phase and formulating precise technical questions for vendors. While it dramatically improved our efficiency in gathering and organizing information, we maintained a rigorous verification process to ensure all data met our project's demanding marine environment requirements. The AI served as a powerful research assistant, but final decisions required careful cross-checking against multiple reliable sources.

5.1.4 Case Studies - Claude

5.1.4.1 Web App Perspective

During early frontend development of the maritime dashboard, the software team consulted Claude to better understand how to structure React components for a real-time data-driven web interface. The prompt provided to Claude was: *"How should I set up my React components for a dashboard application that displays live radar data and vessel overlays?"* . While Claude returned a component layout and code example, the response lacked clarity and contextual explanation. The code did not include comments or describe how the individual components were expected to interact within the broader application.

Additionally, the response relied on utility class names that were later identified as Tailwind CSS, but Claude never explicitly mentioned that Tailwind was being used. There were no instructions regarding installation, integration, or how styling would be handled, which caused confusion for the team, particularly since no CSS files were imported or referenced. Claude also used the native Canvas API for rendering data, but without acknowledging that it was Canvas. At first, this implementation appeared to be part of React-specific logic, but after further investigation, it became clear that the code was leveraging raw HTML5 Canvas methods without providing that important context.

This lack of explanation around technologies and dependencies proved to be a substantial barrier. The team had to independently research the Canvas API and clarify how it fit into a React-based architecture. Overall, while the code technically aligned with the intent of the prompt, Claude's response demonstrated a major limitation. Which was the assumption of prior knowledge. The AI failed to provide instructional scaffolding or commentary suitable for a team still learning frontend best practices. Claude's answer may have benefited more advanced developers, but it ultimately did not support the team's learning objectives effectively in this context.

5.2 Project Impact

After reviewing these LLM and AI platforms, what they offer and how effective they are, we are going to consider how these tools can or cannot be applied in our Senior Design project. Just in the last few years, AI tools and assistants have created both opportunities and problems for engineering students. On one hand, having an AI that can answer any question, generate code, or summarize research is very powerful and can accelerate certain stages of the project. On the other hand, there are concerns regarding fully understanding best practices when using these platforms to avoid over-reliance that might take away from our core learning objectives or introduce errors that go unnoticed. For these reasons, we carefully discussed what parts of our project using LLMs would be beneficial and where it might be risky.

In a broad scope, LLMs can have a large impact on our project in areas like research, design, brainstorming, coding, and documentation. As we discussed, AI tools are easily accessible and students have found them useful for most academic tasks. For the most part, an AI assistant is like an extremely knowledgeable team member who is always available for questions and answers. However, as a project team we wanted to set some

ground rules and best practices for any of us using AI. Our most important guideline is being extra positive that the information we get is accurate, especially since we hope to see our final product scaled and deployed in the real world where wrong information could be disastrous. We also have to consider the ethical guidelines of our course and school, since AI is meant to be used as a helper, not a source of plagiarized content and information.

In the context of Senior Design, time and resources are limited. An LLM can also serve as a force multiplier during the project's timeline. That being said, our team will decide when and how we want to integrate AI. Some teams might choose ChatGPT extensively for research or coding help. While we hope to take advantage of this ability, we also want to limit usage to a certain point to avoid dependency. In terms of this project, there are many benefits we could gain from these tools, as well as determinants that could impact the project's success. With this in mind, we are going to outline the positives and negatives and how they relate to our engineering project.

5.2.1 Benefits

As we mentioned, AI tools have become readily available and are extremely powerful at optimizing and impacting various tasks. In the context of our senior design project, using LLMs can offer several significant benefits, especially in the research and development phases.

One of the first challenges we faced was getting up to speed on unfamiliar topics since none of us have worked on maritime domain awareness systems before. LLMs like ChatGPT and Claude can help as quick research assistants. They are really good at explaining complex concepts, providing definitions, and summarizing materials in a way that is easy to understand. For example, in the research for this project, a team member can ask "How does Automatic Identification System (AIS) tracking work and what information is included in this broadcast?" With the help of AI, it will produce an in-depth explanation and might even compare different maritime safety technologies. As the user, you are able to construct your input and reform it to ask specific questions until you get the most helpful or understandable answer. In our experience, the LLMs have been very useful in the first stage of our project when we are trying to understand various new components and technologies. Instead of searching through dozens of websites and resources, it is much easier now getting a straightforward answer or overview right away, which also helps us know where to focus further reading.

As we know, our Senior Design project will also require a lot of documentation, mainly in the form of reports. LLMs can significantly speed up writing tasks. We see AI being most useful suggesting better wording or translating technical wording into layman's terms. For example, we might conduct an experiment and one of the team members could ask the AI to "summarize these experiments results in an academic but conversational tone" to include in the report. Then you can just edit the response until it fits into your report. Moreover, we believe AI tools will help us outline what needs to be included in certain diagrams or pseudo-code. By offloading some of this work to the AI (while still reviewing and correcting it), we believe it can help us spend more time on actual design and testing.

One of AI's biggest contributions to the space is its ability as a coding assistant. A tool like ChatGPT, can auto-complete code, suggest functions, and generate entire modules based on the user's prompt. In our project, which involves developing an AIS-subtraction algorithm and embedded systems architecture, we expect to leverage AI to generate portions of the codebase. AI assistants are also very useful when stuck with a bug or error message, where asking AI can quickly provide an explanation of the error and possible fixes. The benefit here is not just speed, but sometimes also learning better coding practices from suggestions. However, it is worth noting while AI can help speed up the coding process, it's still crucial that the team members fully understand the code.

Overall, the use of AI tools and LLMs will certainly make us more efficient. Tasks and processes that might usually take hours, can now be completed in minutes with AI's help. It serves as a valuable resource for information and brainstorming. The key is to use it as an assistant, not an easy way out, for gathering information and speeding up routine tasks while maintaining control over final decisions.

5.2.2 Detriments

While these AI tools have many benefits, we are also aware of many determinants that may impact the reliability of these tools. Relying on LLMs without being careful could introduce risks that could impact the project's success.

One of the biggest drawbacks of using an LLM in a project is the possibility of getting incorrect information. As discussed earlier, these models can produce answers that sound very convincing but are completely wrong. If we were to unknowingly use a false piece of information from ChatGPT in our design calculations or component selection, the project could very well face many setbacks or even fail. For instance, if the AI gave an incorrect formula for battery sizing and we designed with that, our system might not meet the requirements. That is why every critical piece of information from the AI needs to be reviewed and verified. To avoid this situation, we made it a rule that any data or fact in our design has to be cross-checked with a credible source like a textbook, datasheet, or reputable website. We understand that there will be times where time pressure might tempt a team member to copy-paste an AI answer, but know to resist that urge without validation. We figured that even with the convenience of saving time, misinformation leading to design failure could result in more lost time then we gained.

Another one of the leading disadvantages with using AI is actually a human factor risk. That is, the temptation to rely on it for every answer which skips learning crucial skills. Imagine a situation where a student uses an AI tool for every design idea or writes their entire report, they might get through the project and even get a good grade, but they will certainly miss the learning that senior design is meant to provide. We were very cautious about this issue. While AI could be helpful, we won't become dependent on it to the point that if something went wrong, we wouldn't understand our own system. There's a known drawback that "increasing reliance on AI risks neglecting human skills and interpersonal interactions" [30]. In senior design or any engineering project, this could mean that team members are not developing debugging skills or not fully understanding the concepts behind the project because the AI is always handing over the answers. To avoid this, our team is treating AI responses as teaching tools, not always solutions.

An interesting detriment we realized is that if AI generates a piece of code or design that we use, and later there's a bug or issue, it might be harder for us to fix if we didn't develop it ourselves. There's a certain familiarity that comes from writing your own code or making your own designs. It is possible that using a function that AI wrote and then it breaks, we might find ourselves spending more time understanding and fixing the issue then the time saved from using AI-generated code. This poses too big of a risk to our project. With this in mind, we will not use AI to generate code for the system rather it will be a teaching tool.

While using AI tools can be very helpful and save a lot of time, it is important to be aware of these detriments. That is why we will remain cautious by cross-checking information, limiting reliance on AI for actual decision-making, and actively learning from AI outputs rather than accepting them as fact.

5.3 Conclusion

After conducting individual case studies on various AI platforms like ChatGPT, Google Gemini, Anthropic Claude, and DeepSeek, we've come to better understand capabilities that can significantly benefit, and sometimes hinder, engineering projects. As each platform continues to grow and advance within our society, it is important to understand the impact they may have on our projects and academic work. These tools have increased efficiency, creativity, and learning by quickly researching unfamiliar topics and almost instantly generating entire codebases and documentations.

However, with the benefits come important limitations. All LLMs share drawbacks in the accuracy of their responses. As we discussed, they can straight up produce incorrect or biased outputs that could severely impact our project if not kept in check. A large issue specifically with students is over-reliance on AI. Without understanding its responses, students will not develop skills the same way they would in normal learning processes. As these tools continue to become more common in industry and school environments, engineers and students will have to understand how to effectively leverage AI while also verifying and validating the information it gives you.

For our senior design project, we adopted a balanced approach. We did not rely heavily on any AI tool for the core engineering work, especially where any risk could impact the success of our project. We found the best way to use LLMs is as assistants to gather information while always cross-checking responses. This approach allowed us to enjoy the productivity gains of AI without exposing ourselves or the project to any unnecessary risks. In cases like code generation, LLMs were occasionally used to produce templates or component layouts which led to gaps in documentation, unclear styling assumptions, or unverified logic. So, while we didn't avoid AI-generated code entirely, we learned that using it responsibly means reviewing every line, understanding the architecture, and not treating it as a drop-in solution.

Going forward, our recommendation for anyone using AI tools in their workflow to do so thoughtfully and transparently. These tools are amazing at fast computation and gathering information, but it is critical to always use your best judgement when interpreting and refining the responses. The most important thing to keep in mind is being up to date with

best practices, as the technology and guidelines evolve quite fast. When used properly, these tools can drive innovation and efficiency. But like any powerful tool, they must be handled with care, considering the drawbacks that we discussed in this section. In the context of real-world systems like our maritime radar platform, we anticipate that AI tools will continue to play a supporting role, not as a substitute for critical thinking or domain knowledge, but as an accelerator for research, prototyping, and documentation. With proper validation and integration into the engineering process. AI has the potential to help teams explore more ideas faster, reduce redundant tasks, and better prepare for evolving technical standards across disciplines.

Chapter 6: Hardware Design

6.1 Power System Design

The power system design for the maritime radar system requires meticulous planning to ensure reliable operation and seamless integration of all components. The fundamental decision between implementing an AC or DC power architecture carries significant implications for system efficiency, cost, and maintainability. Both approaches have been thoroughly evaluated, with each presenting unique advantages and challenges that must be carefully weighed against operational requirements and environmental constraints.

For an AC-based power system, the architecture would center around converting the wind turbine's variable DC output to stable AC power through an inverter stage. This conversion process fundamentally changes the power management requirements compared to a DC system, as it eliminates the need for a charge controller but introduces new considerations regarding waveform quality and synchronization. The AC power would then interface with a specialized battery system like the Predator battery, which is designed to both accept AC charging input and deliver clean AC output to mission-critical components including the Raymarine Cyclone radar and Starlink Mini terminal. The Predator's integrated power distribution system offers multiple high-wattage AC outlets that simplify component integration, while its built-in USB ports provide regulated 5V DC power for the Raspberry Pi MCU, effectively eliminating the need for external AC-to-DC conversion for low-power components. The primary advantages of this AC architecture include native compatibility with commercial off-the-shelf equipment and reduced wiring complexity, which significantly lowers the risk of installation errors and subsequent electrical faults. However, these benefits come with several substantial trade-offs. AC battery systems typically demonstrate reduced operational lifetimes when subjected to the high cyclic loads characteristic of radar and satellite communication systems, particularly during periods of low wind activity when recharge opportunities are limited. The inclusion of high-efficiency inverters, while mitigating some losses, still results in measurable energy dissipation during the DC-to-AC conversion process, typically ranging from 5% to 15% depending on load conditions. Furthermore, the inverter represents a critical single point of failure - its malfunction could potentially disable the entire system unless redundant units or bypass mechanisms are implemented, adding to both cost and system complexity.

In contrast, a DC-based power system would employ a fundamentally different architecture centered around direct energy storage and distribution. The system would utilize a high-quality charge controller to condition the wind turbine's variable DC output before delivering it to a storage bank such as the Dakota Lithium 12V 100Ah $LiFePO_4$ battery. This battery chemistry was selected for its exceptional energy density, deep discharge capability, and superior cycle life compared to traditional lead-acid alternatives, making it particularly well-suited for the demanding maritime environment. Power distribution in this architecture would be managed through a series of precision voltage regulators, with the Starlink Mini PoE DC Power Conversion Kit serving as the interface for the satellite terminal, while the Raspberry Pi could draw power directly from the battery's integrated USB ports. The DC approach offers several compelling advantages,

chief among them being significantly improved energy efficiency by eliminating the multiple power conversion stages required in an AC system. Operational costs are also reduced through the elimination of expensive inverter hardware, while the modular nature of DC power distribution allows for more flexible system expansion and troubleshooting. However, these benefits must be balanced against several technical challenges. The DC architecture requires meticulous voltage regulation design to ensure each component receives stable, clean power within its specified operating parameters. Proper grounding and noise suppression become critical design considerations, particularly for sensitive RF equipment like the radar system. While DC systems are inherently less susceptible to single-point failures compared to their AC counterparts, they remain vulnerable to extended periods of low wind activity, necessitating careful capacity planning for the battery bank.

The selection between AC and DC power architectures ultimately hinges on a careful evaluation of operational priorities and constraints. The AC system offers superior compatibility and ease of integration at the cost of reduced efficiency and higher long-term maintenance requirements. Conversely, the DC system provides greater energy efficiency and lower operational costs but demands more sophisticated design implementation and presents greater integration challenges. Additional factors including expected duty cycles, maintenance accessibility, and future expansion requirements must all be considered in the final design selection. Comprehensive testing under realistic operating conditions will be essential to validate the chosen architecture's performance and reliability, with particular attention paid to power quality metrics, system transient response, and fault recovery characteristics. The DC architecture currently presents a more compelling solution for deployments where energy efficiency and operational cost are primary concerns, while the AC system may be preferable for installations prioritizing rapid deployment and compatibility with existing infrastructure. Both approaches will require careful implementation of power monitoring and management systems to ensure optimal performance throughout the system's operational lifecycle.

Upon further investigation, we decided to pursue a DC-based power system for the MDAS. This decision was primarily driven by the power requirements of the Raymarine Cyclone radar, which necessitates a clean 12V or 24V DC input through its VCM1000 interface module. An AC system would have required unnecessary power conversion stages—first from the wind turbine’s DC output to AC via an inverter, and then back to DC within the radar’s onboard systems—resulting in greater energy losses and added system complexity.

In addition, DC power distribution offers significantly improved efficiency compared to AC architectures, which suffer from conversion inefficiencies and potential single points of failure in the inverter stage. With a DC system, power from the wind turbine can be stored and regulated directly without the intermediate conversion, preserving more of the harvested energy and improving overall system resilience.

The modularity of a DC architecture also simplifies expansion, allowing additional loads or upgraded components to be integrated without extensive redesign. Overall, given the Cyclone radar’s inherent DC requirements and the operational benefits of higher

efficiency, lower maintenance, and greater fault tolerance, a DC power architecture was determined to be the optimal solution for the MDAS deployment.

6.1.1 Understanding the Implementation of a DC System

This hybrid wind-powered system is designed for use in a maritime or remote coastal environment, with the goal of reliably powering a set of critical electronic components: a high-performance Raymarine Cyclone radar system, a Starlink Mini satellite internet terminal, a gimbal-stabilized camera system, and a Raspberry Pi 5 microcontroller. The system is engineered for autonomy, resilience, and energy efficiency in off-grid conditions, using a Pikasola wind turbine as the primary energy source and a lithium battery bank as the energy reservoir. It also incorporates an MPPT (Maximum Power Point Tracking) wind charge controller to ensure optimized charging performance regardless of wind variability.

The design prioritizes clean DC power distribution, minimizes conversion losses, and utilizes marine-rated wiring and components where necessary. It features integrated voltage regulation and smart power management via DC-DC converters and a fuse-protected distribution layout to support each device's specific input requirements. This system is modular and scalable, allowing for future component additions or battery capacity upgrades. Its low-maintenance nature and the elimination of fuel reliance make it ideal for long-term deployments in isolated or energy-scarce environments.

The schematic diagram illustrates the electrical and communication layout for the system. All components are labeled and numerically identified, with color-coded wiring representing power(red for positive and black for negative) and signal lines(purple). The system is built around a centralized lithium battery, which is charged by the wind turbine and then powers each connected component through direct or converted voltage paths. Below is a detailed explanation of each numbered component in the diagram:

Given the decision to adopt a DC architecture, special attention was devoted to designing the system for maximum efficiency and protection. The choice of a 12V $LiFePO_4$ battery as the central storage element reflects a balance between energy density, operational safety, and compatibility with the system’s major loads. In particular, the direct compatibility between the battery voltage and the radar’s requirements allows for simplified wiring and reduced energy losses compared to an AC-inverted setup. The use of MPPT charge control further optimizes energy capture from variable wind conditions, ensuring the battery bank maintains sufficient reserves to power the MDAS platform even during extended periods of low wind activity.

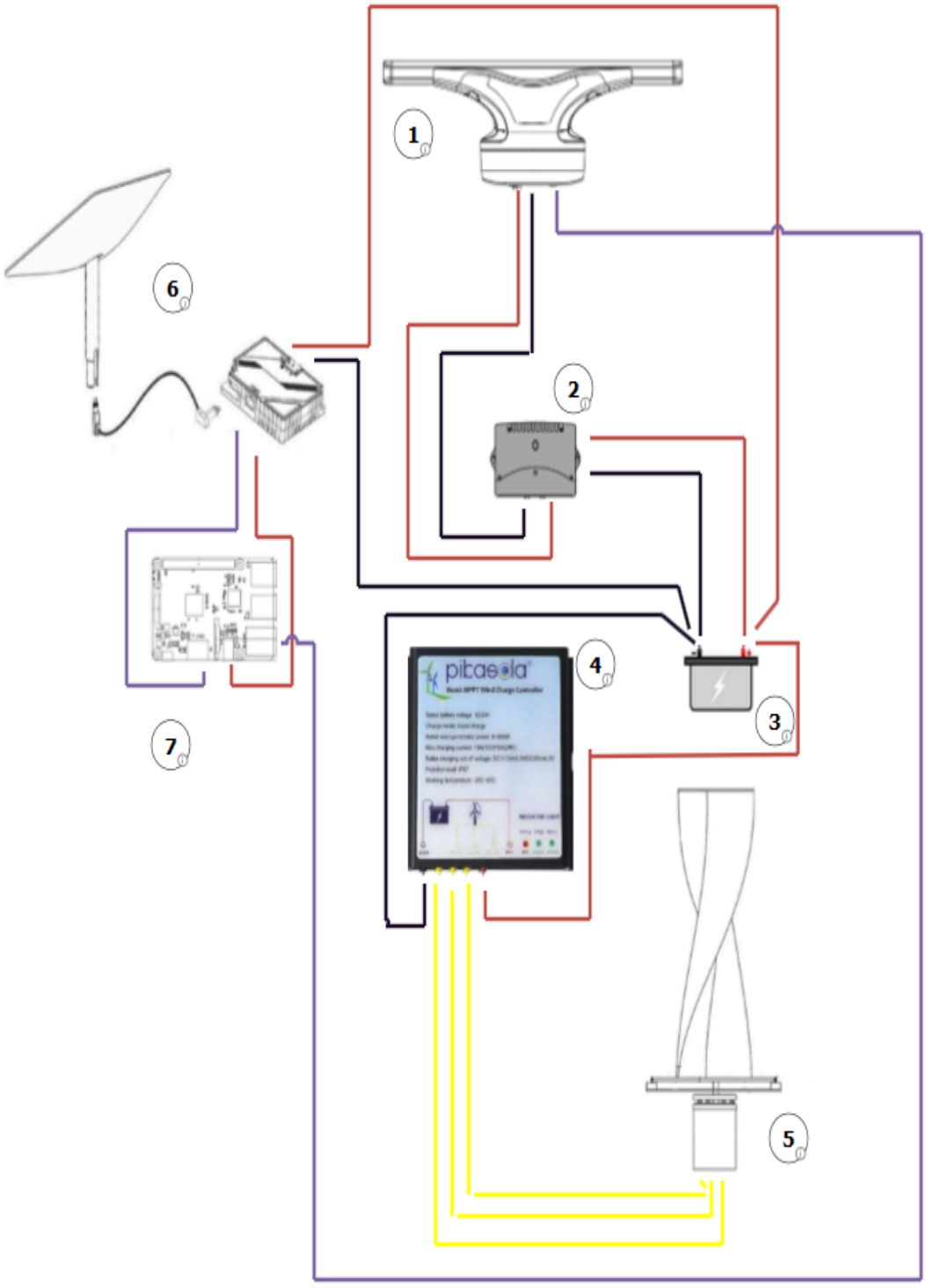


Figure 6.1 *- MDAS Radar System Diagram*

1. Raymarine Cyclone Radar

- Purpose: Marine radar system for navigation and object detection.
- Power Needs: 12–24V DC via Raymarine's VCM1000 interface module.
- System Role: One of the highest-draw components in standard and peak operation.
- Notes: Must be fused (20A) and mounted securely with minimal vibration. Requires clean, regulated input voltage and interference-free communication wiring.

2. VCM1000 (Radar Interface Module)

- Purpose: Converts power to required voltage levels and communicates between the radar and control system.
- Placement: Typically located near the radar or within a protected electronics bay.
- Connections: Takes in power from the battery system and feeds it to the radar dome. Also interfaces with the Raspberry Pi or controller for communication and diagnostics.

3. Lithium Battery (e.g., 12V 50Ah or 100Ah $LiFePO_4$)

- Purpose: Stores electrical energy from the wind turbine.
- Type: Marine-rated, deep-cycle $LiFePO_4$ battery with built-in BMS.
- Location: Central energy hub in the system.
- Considerations: Ensure proper venting, marine casing, and fuse protection. All system current loads pass through this battery either directly or via converters.

4. MPPT Wind Charge Controller (Pikasola Branded or Compatible)

- Purpose: Regulates and optimizes incoming energy from the wind turbine before passing it to the battery.
- Voltage Matching: Matches turbine voltage with battery voltage (typically 12V or 24V).
- MPPT Advantage: Ensures highest efficiency charging even during variable wind conditions.
- Indicator: Screen shows power status, turbine RPM, battery level, and error diagnostics.

5. Wind Turbine (AC Output, 400W Rated)

- Function: Converts wind energy into 3-phase AC electrical power.

- Wiring: Connects via 3 yellow AC lines to the MPPT controller.
- Mounting: Should be installed in a high-wind location, preferably pole-mounted with minimal vibration and marine-rated brackets.
- Cut-in Wind Speed: ~2.5 m/s; peak performance at ~12 m/s.
- Special Note: Ensure free airflow and perform periodic mechanical checks.

6. Starlink Mini System

- Purpose: Provides low-latency satellite internet access.
- Power Needs: 56V DC via PoE (Power over Ethernet).
- System Impact: One of the largest continuous draws in the system (50–75W).
- Conversion: Requires a step-up DC-DC converter from 12V or 24V to 56V.

7. Raspberry Pi 5 + Gimbal Camera System

- Purpose: Handles system logic, telemetry, and video processing.
- Voltage Needs: 5V USB-C for the Pi and 12V DC for the camera.
- Conversion: Step-down (buck) converters used to feed these devices from the 12V battery line.
- Communication: May connect with radar via Ethernet or serial, and with the internet via Starlink.

6.1.1.1 Generating Usable Power

Generating usable power is something that is vastly important in the power system design and is the primary challenge that needs to be tackled. Wind turbines generate power on a wild spectrum due to the variability of wind speeds that spin the alternator. This wild current that gets output needs to be conditioned so that the battery that the turbine gets connected to does not get damaged. Most wind turbines come with a native charge controller and regulator, so it is important to ensure the battery matches the voltage output of the charge controller.

Beyond simple voltage matching, the implementation of an MPPT-based charge controller allows the system to dynamically adjust the electrical load seen by the turbine, maximizing the amount of power harvested under varying wind conditions. Without MPPT, a conventional fixed-load controller could result in significant power loss, particularly during gusty or turbulent wind conditions common in coastal deployments. By properly sizing the charge controller to both the turbine's output characteristics and the battery's charging profile (e.g., bulk/absorption/float stages for $LiFePO_4$ chemistry), we ensure not only optimal energy capture but also extend the service life of the battery through proper management of charging currents and voltages.

6.1.1.2 Utilizing Regulators to Supply Power

From the battery, an array of regulators will need to be implemented to supply power to certain components on the Maritime Radar System. The main concern being devices that typically rely on AC power and only come equipped with AC to DC power supplies. The main culprit of this being the starlink.

The Starlink Mini comes equipped with an AC to DC converter to comply with on-grid infrastructure, however, since the system we will be employing is strictly off-grid, a DC to DC converter needs to be implemented. As previously explored, adapters of this nature exist and will need to be purchased in-order to adapt the power system being DC only. As seen in Figure 6.x, the starlink regulator is capable of stepping up the 12 volt input of the battery and is able to supply the Starlink with an adequate amount of power based on our system design. In addition, the step up device also includes a 5 volt USB port which the PI MCU could make use of as its power source.

The voltage regulation strategy is critical not only for delivering the correct voltage levels but also for ensuring clean, ripple-free power to sensitive devices such as the Starlink Mini and the Raspberry Pi 5. Low-quality voltage regulation could introduce noise and voltage instability, leading to communication errors, MCU resets, or even permanent device damage. Therefore, high-efficiency buck and boost converters with built-in filtering capacitors were selected to maintain power integrity. Each regulator is also protected by individual inline fuses sized appropriately for the load it serves, ensuring that faults are isolated and do not propagate through the entire power distribution network.

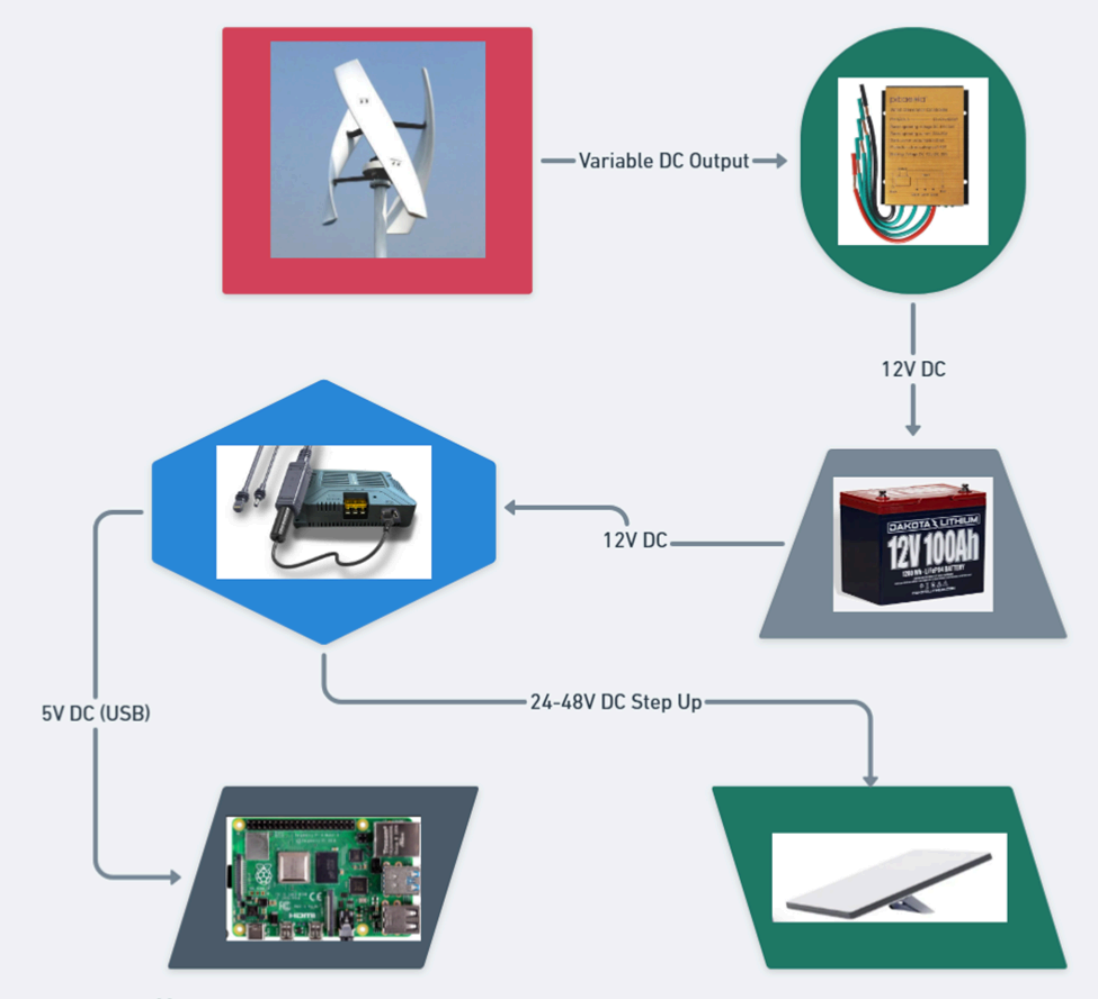


Figure 6.2 *- Power System to deliver DC Voltage to the Pi MCU and Starlink Mini Terminal*

The RayMarine Cyclone Radar requires its own specific voltage regulator that needs to be powered and operational in-order to power the radar itself. This regulator is the VCM100 and is a voltage converter module that is needed to take a DC input of either 12 or 24 volts and output the desired voltage and Amps that the RayMarine Cyclone draws.

In our power system, the system voltage will likely be 12 Volts and thus the wiring and fuzes implemented will change. Using 12 Volts, we will need a 20 Amp breaker between the distribution box (battery in our case) and the voltage converter module. In addition we must take in consideration that the VCM100 may draw up to less than 20 Amps and thus the wiring needs to be adequately robust to support this current. To account for these things in the prototype, 14 gauge wire was used which is rated for 15 Amps. Since 15 Amps is the maximum Amperage that our prototype power bank can produce, this is all that was needed. A 20 Amp fuse was also integrated for safety of the system. With this setup, we were able to confirm power being provided with the VCM100 and verify that the Cyclone Radar was in sleep mode, ready to take commands from the network cable.

During testing, special attention was paid to voltage drop across the wiring, especially under peak current draw conditions. In future full-scale deployments, thicker cabling (such as 12 AWG) may be specified to further reduce line losses and enhance resilience to thermal stress over longer cable runs. Thermal management considerations for the VCM100 housing itself were also evaluated to ensure that in enclosed installations, adequate ventilation or passive heat sinking would prevent overheating during extended radar operation sessions.

6.1.1.3 Wind Turbine Implementation

Implementing a wind turbine as part of the power system requires careful selection of a compatible charge controller to ensure safe and efficient battery charging. The charge controller serves as the critical interface between the turbine's variable output and the battery bank, regulating voltage and current to prevent overcharging, undercharging, or damage to the battery. Using an incorrect charge controller—such as one with mismatched voltage thresholds or insufficient current handling—can lead to catastrophic failure, including battery degradation, reduced capacity, or even thermal runaway in extreme cases. For example, if the controller's output voltage is lower than the battery's required charging voltage (e.g., 14.4V for a 12V $LiFePO_4$ battery), the battery may only partially charge, leaving the system vulnerable to power shortages during critical operations.

Most small-scale wind turbines employ a three-phase brushless alternator, which produces alternating current (AC) that must be rectified to DC for battery storage. These turbines typically have three output wires (one for each phase) and a ground wire for safety. The charge controller must accommodate this three-phase input, converting the turbine's variable AC output into a stable DC voltage that matches the battery bank's requirements (e.g., 12V or 24V). For our system, which will likely use a 12V $LiFePO_4$ battery, the charge controller must be specifically designed for 12V systems and capable of delivering the correct three-stage charging profile (bulk, absorption, and float) to maximize battery life and performance.

The wiring configuration between the turbine, charge controller, and battery is equally critical. The turbine's three-phase wires connect to the controller's input terminals, while the controller's output terminals deliver regulated DC power to the battery. Proper grounding is essential to protect against electrical surges, particularly in marine environments where corrosion and moisture are constant threats. Additionally, the charge controller should include features like maximum power point tracking (MPPT) for wind turbines, which optimizes energy harvest across varying wind speeds, and programmable voltage thresholds to match the $LiFePO_4$ battery's unique charging characteristics. Failure to adhere to these specifications could result in inefficient power conversion, reduced turbine output, or even damage to the battery or connected electronics.

To further harden the wind energy subsystem against marine-specific challenges, waterproof junction boxes, tinned marine-grade cabling, and UV-resistant cable sheaths will be used throughout the turbine wiring paths. Redundant surge suppression devices will also be implemented between the controller output and the battery input to guard against voltage spikes induced by transient wind conditions or lightning strikes nearby.

6.2 Sensing Hardware: Raymarine Radar

The Raymarine Cyclone radar serves as the primary sensing device for vessel detection in the MDAS system. It is physically connected to the onboard NUC PC through a network cable, which allows communication over a local area network. This connection facilitates both the transfer of radar data to the onboard processor and the transmission of control commands from the MDAS software back to the radar. This setup enables real-time scanning, remote configuration, and command-based control of radar functions.

6.3 Onboard Processing Unit: Intel NUC

A Raspberry Pi was originally planned as the main computer for the MDAS system, but an Intel NUC PC was provided instead for the prototype phase. The NUC offers faster processing and better stability for running the radar software which is great for developing on.

It connects to the Starlink Mini terminal via an Ethernet networking cable to access the internet for tasks like cloud syncing and remote control. It also connects to the radar using a second Ethernet port or network switch, acting as the link between the radar and remote systems. In the future, the NUC may be replaced with a smaller, lower-power computer like a Raspberry Pi 5 or Jetson Nano.

6.4 Satellite Communication Unit: Starlink Mini

The Starlink Mini is the MDAS system’s main communication link to the remote base station. It provides satellite internet access, allowing the system to transmit radar data, receive remote commands, and sync with cloud services even in remote ocean regions.

It connects to the NUC PC via Ethernet, enabling reliable data transfer and remote access. The unit is mounted in a position with clear sky visibility to ensure a strong satellite connection. Its compact size and ease of setup make it suitable for use on various vessel types.

6.5 System Monitoring Accessory: Smartphone Interface

Initial design concepts wanted to utilize a smartphone as an internet signal for the MDAS system. However, research determined that a smartphone alone could not replace the role of a Starlink terminal for satellite data transmission. As a result, the smartphone's role was shifted to act as an optional accessory for system monitoring.

When paired with the MDAS system, the smartphone allows the operator to track data transmission rates, monitor system status, and remotely toggle radar power on and off via the cloud-linked server interface. While it is not a required component for autonomous operation, it is still a piece of hardware that can be included into our design to provide flexibility in testing.

6.6 Wiring and Network Adapters

This section covers how we set up the power wiring and network connections for the MDAS system. Both are important to make sure everything runs smoothly and stays safe, especially in outdoor and marine environments. We explain the types of wires we used, how they're protected, and how the radar and other devices are connected to each other and to the internet.

6.6.1 Power Wiring Considerations

Proper power wiring is critical to ensuring the reliable operation and long-term durability of the MDAS system, especially given the harsh marine environments it is intended to operate in. Careful selection of wire types, sizing, protective measures, and routing strategies directly impacts both performance and system safety.

All power wiring in the MDAS will utilize marine-grade, tinned copper wire. Tinned copper provides superior resistance to corrosion compared to standard bare copper wire, which is essential in saltwater environments where moisture, humidity, and salt exposure are constant threats. In addition to corrosion resistance, marine-grade wire features more flexible, fine-strand construction that resists vibration fatigue better than typical household or automotive wiring. This flexibility is crucial on systems exposed to constant movement and vibration from wind, waves, and mechanical systems like the radar array.

Wire gauge selection is based on the maximum expected current draw of each subsystem, the acceptable voltage drop, and the total wire run length. As a general design rule, voltage drop across power wiring will be kept below 3% to ensure stable operation of sensitive electronics like the Cyclone radar and the Starlink Mini.

- For high-current devices like the Raymarine Cyclone radar and its associated VCM100 voltage converter, 14 AWG wire was used in the prototype, rated for up to 15A. In production systems, 12 AWG or thicker wire may be employed depending on final load measurements and run length.
- Lower current devices such as the Raspberry Pi and small sensors will utilize 16–18 AWG marine wire.
- Inline ATC-style fuses or breaker systems are installed as close to the battery bank as possible to protect against overload and short circuits/

All wiring is routed through UV-resistant, waterproof conduit wherever feasible to protect against mechanical abrasion and sunlight exposure. Entry points into enclosures are sealed with marine-grade cable glands to maintain weatherproof integrity. Wiring looms are secured with non-corrosive clamps spaced at regular intervals to prevent flexing and fatigue over time. In addition, special attention is paid to grounding practices to ensure consistent system reference voltages and to protect against lightning-induced surges or electrostatic discharge.

6.6.2 Network Connections and Adapters

Beyond power delivery, robust and reliable data communication is critical to the MDAS system's operation, enabling radar data processing, control signaling, and telemetry uploads.

The Raymarine Cyclone radar dome interfaces with the edge processing system (Intel NUC) through a RayNet to RJ45 Ethernet adapter. RayNet is Raymarine's proprietary Ethernet-based protocol designed for marine networking, and the adapter enables the NUC, which has a standard Ethernet port, to communicate with the radar.

During initial deployment, the Intel NUC will serve as the primary data collection and processing platform, interfacing directly with the radar through this RayNet-to-Ethernet link. This connection enables the NUC to receive raw radar data streams, control the radar scanning parameters, and perform preliminary object detection and telemetry preparation tasks.

As system development progresses, the goal is to migrate primary radar control and data handling responsibilities from the Intel NUC to the more power-efficient Raspberry Pi 5. When this transition occurs, the same RayNet-to-Ethernet adapter will be employed, connecting the Cyclone radar directly to the Raspberry Pi's onboard Ethernet port. Software drivers and protocols will be adapted accordingly to ensure seamless operation on the new platform.

This transition strategy enables system flexibility and modular upgrades without needing to modify existing hardware connections or rerun network cabling.

Both the NUC and Raspberry Pi require outbound internet access to upload processed radar data, system telemetry, and operational health reports to cloud servers. This is achieved through a wired Ethernet connection to the Starlink Mini terminal's integrated router.

A simple Ethernet switch may be deployed in the system to allow simultaneous connections from both the NUC and the Raspberry Pi to the Starlink network. This ensures that regardless of which device is handling primary processing duties, reliable, high-bandwidth uplink capabilities are maintained. Wired connections are preferred over wireless to maximize speed, reduce latency, and improve reliability, particularly in congested or high-interference environments.

6.7 Imaging System

For the optical portion of this project, there are three main components: the lens system, the gimbal, and the CMOS sensor. The lens system consists of an achromatic doublet lens encased within a lens tube that blocks the outside light. The lens focuses the light onto the CMOS sensor, creating the image. The gimbal is a 3-Axis rotational model that provides adequate movement in the horizontal and vertical directions. These systems create the ship imaging system and integrate into the MDAS for added security and awareness.

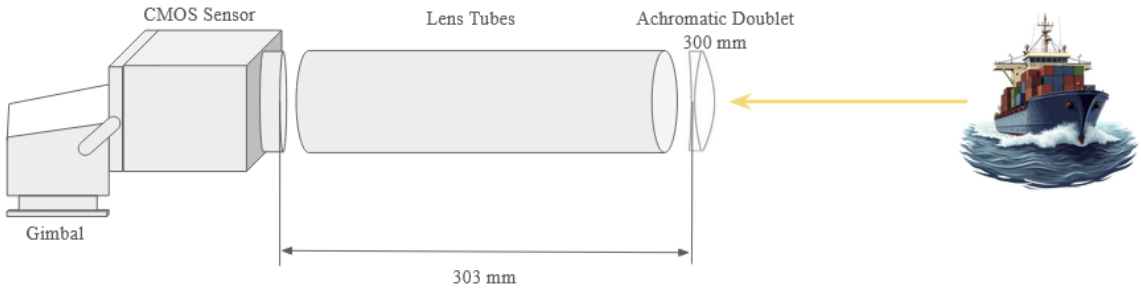


Figure 6.3 *- Optical System Layout*

6.7.1 Lens System

For the lens system, the original objectives of the imaging system were a 5-7.5x magnification with a 1.056 arcseconds angular resolution. After running simulations on Zemax, the magnification of the system came out to 4x, and the angular resolution 3.30 arcseconds. The Zemax diagram and measurements are:

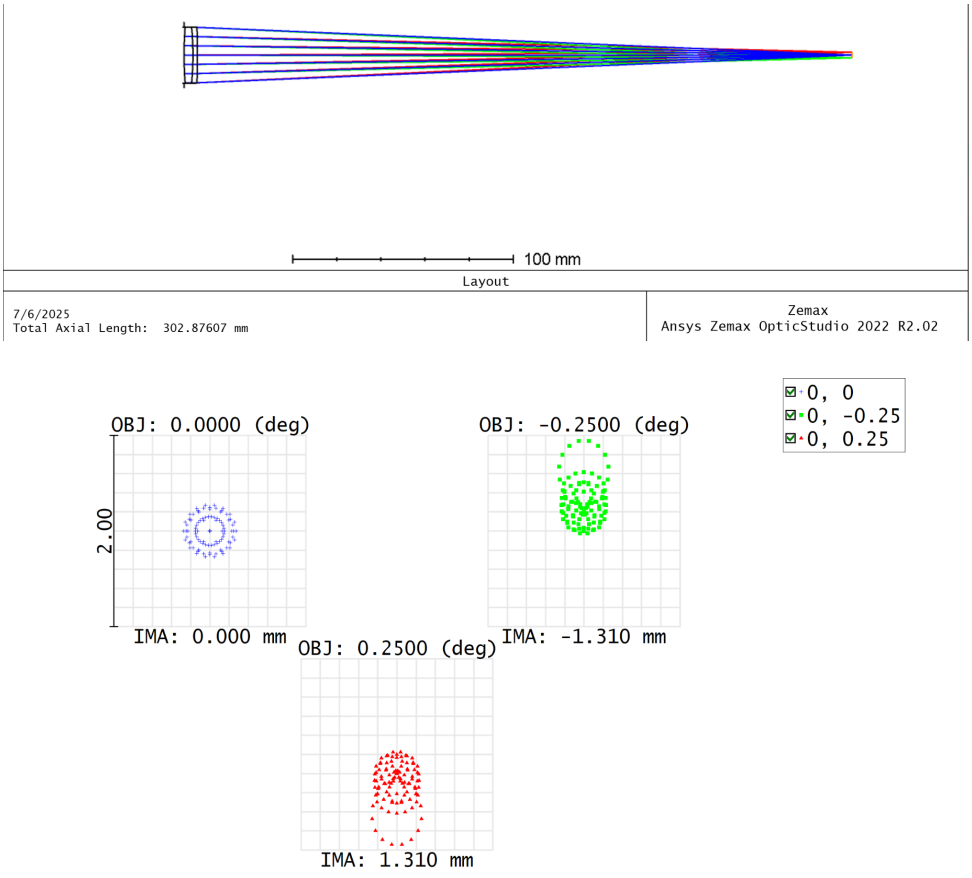


Figure 6.4 *- Zemax Data*

The lens in the system is a 300 mm achromatic doublet. The magnification of the system is $M = \frac{sensor\ size}{object\ size} = \frac{\#\ of\ pixels * 4.8\ um}{object\ size\ (mm)} \sim 1.5x$. While this is below our anticipated value of 3x, it is still good for the purpose of the system. For the FOV of the system, we need to look at the sensor dimensions as well as the objective lens. The sensor specifications can be found in Appendix C, [3]. The FOVs are as follows:

- HFOV = $\frac{sensor\ dimension}{f_{obj}} = \frac{3.9\ mm}{300\ mm} = 0.013\ rads = 0.745°$
- VFOV = $\frac{2.9\ mm}{300\ mm} = 0.009667\ rads = 0.554°$
- DFOV = $\frac{4.86\ mm}{300\ mm} = 0.0162\ rads = 0.928°$
- $\alpha = \frac{pixel\ size}{f_{obj}} = 1.6 \times 10^{-5}\ rads = 0.000917° = 3.30\ arcseconds$

The resulting imaging area is acceptable for imaging at a distance, which is the goal of the system. Through the testing of the system, we determined that the distance needed to be adjusted. For prototype purposes, we chose the demonstration distance to be 5 meters. We can convert the angular FOV to a physical coverage using our estimated distance of 5 m, with the horizontal coverage being $W = 2 \cdot 5 \cdot tan\frac{0.745°}{2} = 3.9\,m$ and the vertical coverage being $H = 2 \cdot 5 \cdot tan\frac{0.554°}{2} = 2.8\,m$. While the horizontal coverage is not the entire length of a standard cargo ship, it is definitely fitting for the purposes of the prototype demonstration and can still produce good pictures of identifiable features for ships. Looking at the resolution at 5 m, the angular resolution correlates to a spatial resolution of around 8 mm, meaning that text and other fine details will be able to be clearly resolved. This value also matches the Nyquist limit of the sensor, meaning there won't be any sampling issues.

6.7.2 Gimbal

For the gimbal, we will be using an SCORP-C 3-Axis Mirrorless Gimbal kit, which can be controlled through the handle or using bluetooth to the partner application. For stabilizing the camera sensor on the high seas, we determined that we needed the three-axis movement, as opposed to the pan-tilt model previously discussed. This model's range of motion is perfect for our project, as we need to utilize all movement spaces to get accurate images of passing ships or other targets.

Figure 6.5 - *Gimbal Apparatus*

The mounting platform of the gimbal is an adjustable quick-release mounting plate, which can definitely support the CMOS sensor. It also comes with a lens-support bracket, which was definitely useful since the 300 mm of lens tubes was somewhat heavy.

Chapter 7: Software Design

This project consists of two core software components that operate across different environments. The first is a command center application that runs the AIS Subtraction Algorithm. Its role is to process and compare radar data with AIS transponder information, removing known vessel targets and highlighting potential dark ships. This software will run on a dedicated system at the base station and provide a dashboard for analysts to monitor maritime activity. The second component is the embedded software developed for the Raspberry Pi onboard the vessel. This program is responsible for handling real-time sensor data, controlling hardware systems like the gimbal and camera, managing communication between systems, and ensuring that relevant data is sent back to the base station. As much of this data received by the radar can be sensitive to government/state entities and other organizations, we also include our design goals for implementing software security. Each of these software systems plays a critical role in enabling the Maritime Domain Awareness (MDA) capabilities of our system and are developed with unique goals and constraints made for their respective platforms.

7.1 Dark Vessel Classification Algorithm

The AIS Classification Algorithm is the primary function for identifying and determining if a detected vessel is performing illicit or normal activity. The classification algorithm relies upon data collected and transmitted from the radar through a satellite transmission device, such as Starlink or a satellite phone, to an offsite bay station. The bay station must also receive the publicly broadcast Automatic Identification System (AIS) data which includes data such as ship name, course and speed, classification, call sign, registration number, Maritime Mobile Service Identity (MMSI), along with other information. Utilizing this information, we can create an algorithm that can utilize the publicly broadcast AIS information alongside the information for detected vessels from the on-board radar system to classify the detected vessel and determine if it may be performing illicit activity. The classification process includes parsing and formatting the data received by the radar, so that the radar data is comparable to incoming AIS data and can be used to compare, retrieval of the publicly broadcast AIS information, comparing the publicly broadcast AIS data to the formatted radar data and “subtracting” the vessels that are determined to be “known” (i.e. vessels that are publicly broadcasting AIS) to isolate determined “dark vessels” for classification, and classifying the dark vessels based upon the radar data received about the detected vessel and its patterns. Data collected by the radar is mapped to an object, AIS known-vessel, dark vessel, or unknown object. This list can be expanded as more objects become identifiable. Each object allows us to set a flag based on its behavior. If an object detected by the radar is an AIS-known object, we can map it to its unique MMSI. If an object is classified as a vessel, but not mapped to a known AIS vessel, we can flag it as potential illicit activity. Other objects detected by the radar that cannot or should not be classified by the classification/anomaly detection algorithm should be flagged accordingly. Figure 7.1 demonstrates each state of the process to be described in the following sections.

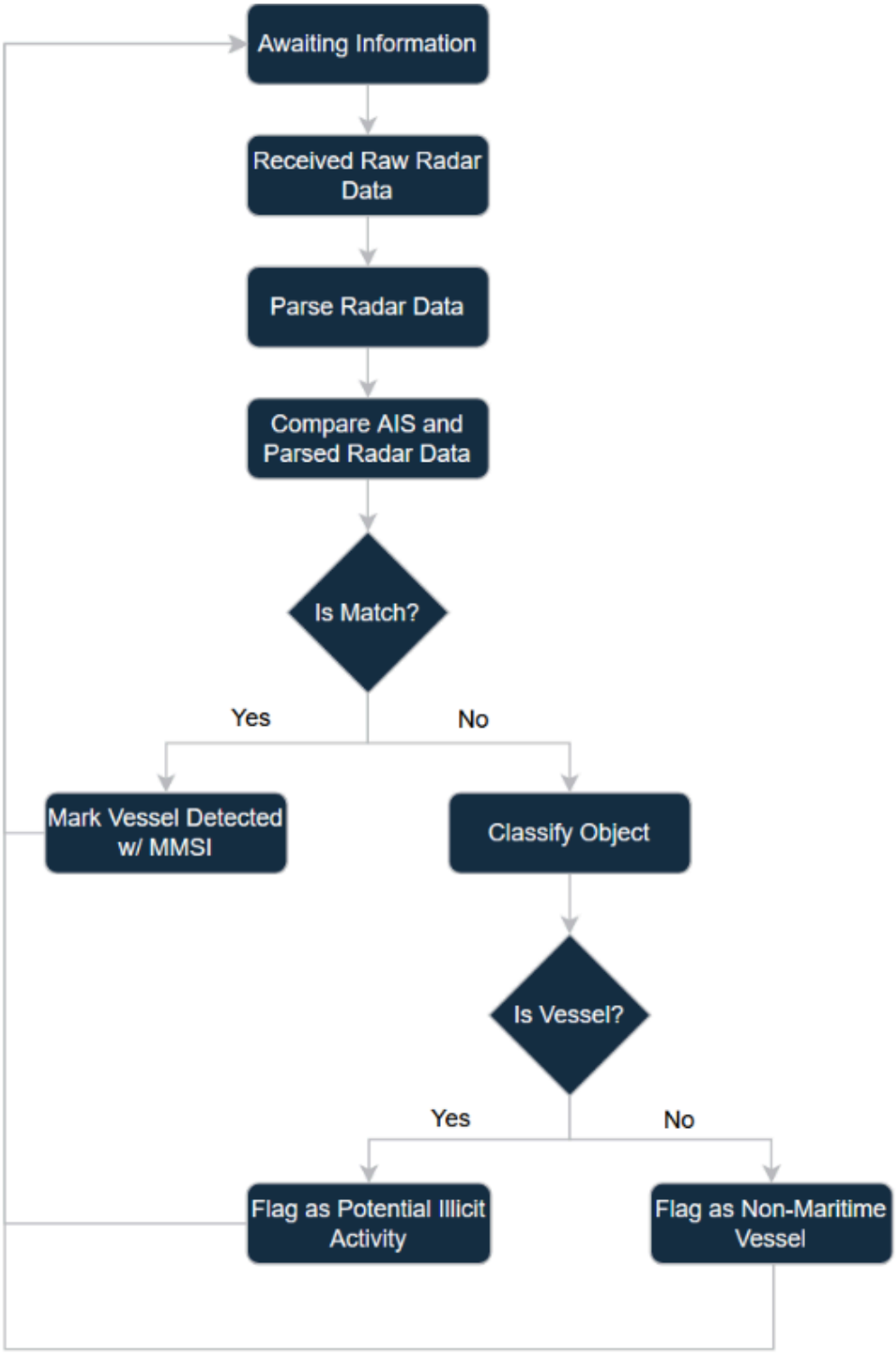


Figure 7.1 *- Dark Vessel Classification State Diagram*

7.1.1 Automatic Identification System

The Automatic Information System is used for Maritime Vessel Traffic identification. As mentioned above, the AIS system delivers information including, but not limited to, ship name, course and speed, classification, call sign, registration number, and Maritime Mobile Service Identity (MMSI). Each AIS system consists of one Very High Frequency (VHF) transmitter, two VHF TDMA receivers, one VHF DSC receiver, and standard marine electronic communications links (IEC 61162/NMEA 0183) to shipboard display and sensor systems. Position and timing information is normally derived from an integral or external global navigation satellite system (e.g. GPS) receiver, including a medium frequency differential GNSS receiver for precise position in coastal and inland waters. Other information broadcast by the AIS, if available, is electronically obtained from shipboard equipment through standard marine data connections. Heading information and course and speed over ground would normally be provided by all AIS-equipped ships. Other information, such as rate of turn, angle of heel, pitch and roll, and destination and ETA could also be provided. [36] As defined in section 4.1.4 of this document, vessels required to have AIS must operate it in U.S. navigable waters at all times that the vessel is navigating (underway or at anchor) and at least 15 minutes prior to unmooring.

7.1.2 Retrieving AIS Data

As noted in section 4.1.4, maritime vessels are legally required to publicly broadcast AIS dependent on their class designation and allow users to collect information about all vessels transmitting their VHF signal. Collecting this information however, can be a challenge financially if requiring a large amount of AIS data. Detecting local vessels, within VHF range, can be done freely using an AIS receiver with software such as OpenCPN, SeaClear, or NMEA Reader to parse and visualize the data. However, this limits the user locating only vessels within VHF range, around 100 miles. Should vessels outside of the VHF range need to be detected, a subscription to a web service API, like MarineTraffic or VesselFinder, or access to a satellite-based service, such as Orbcomm or ExactEarth. As previously mentioned, a web service or satellite-based service can be expensive as many services require subscription access to the respective applications or API's to use in alternative applications. Since our mounted radar has a range of approximately 72 miles, it is feasible to utilize an AIS receiver mounted on-board the ship. Section 4.1.4 describes the vessels that are legally required to carry all AIS equipment.

In the case that a VHF receiver is used to retrieve AIS information, the data would need to be communicated Ship-to-Shore, utilizing the satellite-based communication system on-board the ship. The information would be sent to the base station and parsed for comparison with the detected radar data. In the case that a web-based or satellite-based system is used, AIS information can be fetched from the system on request as the radar system detects a vessel. Implementation of the API and the service, however, can affect the exact details of the retrieval and accuracy of the data. Alternatively, many higher-end ARPA radar systems also include built-in radio receivers, which are capable of and in most cases do process AIS data for collision detection and identification. This technology

can also be leveraged to retrieve AIS data and map the corresponding object to an known AIS vessel.

7.1.3 Parsing and Formatting of Data

As radar information is collected, it is processed and sent to an on-shore base station for processing. This radar data is, as described in section 7.1.4, compared with AIS information from compliant vessels to filter for dark vessels. The radar information may not directly match AIS information, so we must parse and format the data for that comparison.

Table 7.1 *- AIS and ARPA Data*

Automatic Identification System	**ARPA Radar Data**
• Time • MMSI Number • Latitude, Longitude • Speed Knots • Heading • Course Over Ground • Rate of Turn • Ship Type • Ship Name • Ship Length, Beam, Draft • Navigation Status	Data Provided About the Detected Object • ID • Range • Bearing • Course • Speed MDAS Additions from GPS of Mounted Vessel • Longitude • Latitude

Determining the location of both the detected vessel and the MDAS equipped vessel remains the most important data point returned in each set. The mounted detection system includes a GPS module that can be utilized to transmit the vessel-being-mounted-upon location. The radar gives information such as the distance of detection (or range of the object), speed, direction (or azimuth), size and shape, altitude, etc. Utilizing the GPS location of the MDAS equipped vessel alongside the information returned by the MDAS radar on a detected object can help determine both the general area that a known vessel should be found and how it should be classified in AIS systems based upon its speed, size (and potentially weight based on size).

The vessel equipped with the MDAS system, MDAV, and the object detected, VD, can be compared using their current location and time each transmission was received. Because the MDAV is mounted with its own AIS transponder, its longitude, latitude, course over ground, speed over ground, etc. is calculated using its on-board GPS unit. As the radar detects VD, we can calculate VD's latitude and longitude using MDAV's location information and adding the distance vector returned by the radar. As the radar continues to detect VD, we can continue the same calculations and generate a Time-Series for the detected object mapped to a specific object ID.. Since we cannot be sure that the vessel is or is not transmitting AIS, this Time-Series allows us to compare

AIS data as it is collected and disjoint in time from radar detection and receival of an AIS transmission.

7.1.4 Subtracting Known Vessels

We can perform two methods for verifying that a detected object matches a known AIS vessel. Upon each instance of a detected object, after parsing its information, we can cross-reference the AIS information and attempt to match the detected object with its known location (i.e. longitude and latitude) and a predetermined time interval. This time interval should be sufficiently long enough to allow for any temporal discrepancies based on time of transmission, receival of AIS data, time to detect the object and pre-processing time of the radar data while also remaining short enough so there is no overlap creating a scenario where two vessels could reasonably exist. This method allows for the quickest processing time and verification time for a detected object. However, this method does not allow for the AIS transmission to be received after an object is detected by the radar and preprocessed to be in a given location at a specific time. For this case, we can use the Time-Series for the detected object.

By maintaining the Time-Series information for a detected object, we can compare AIS transmissions that encode their timestamps with the time received rather than the time sent. This method also allows us to better compare a detected object with AIS transmissions. In the case, such as closer to a port, that there are many vessels in a given area it can be difficult to determine whether an object is transmitting AIS. The detected object or another nearby object may be transmitting incorrect location information due to a faulty GPS unit or using the first method, the time interval may be large enough where it is reasonable that two or more vessels could be in the same location. Using the Time-Series information, we create a probability threshold that the detected object does match an AIS vessel by comparing the object's Time-Series features with all AIS vessels that match the same temporal and spatial features. Any vessel that exceeds that threshold can be practically mapped to an AIS vessel. If the object cannot be mapped, we should attempt to classify the object and determine whether it is a maritime vessel or any other object.

7.1.5 Classifying Dark Vessels

Classifying DarkVessels is the last step in identifying vessels performing illicit activity. As mentioned in the following section, vessels may not match to a known AIS vessel. For example, misreported width, breadth in the detected vessel's AIS transmission will cause a flag to be raised on the detected vessel. In fact, the most important features in vessel classification using AIS transmission are self-reported. [6] Many AIS features such as distance from GPS module to bow, stern, starboard, and port sides determine the breadth and width of a ship. Misreporting these features can cause a misclassification in size. Speed over ground and course over ground are also calculated by the GPS, so any fault in the GPS system, such as lock failure, will add noise to the data collected. Time of transmission of the AIS data is also dependent on whether its transmission time or receival time is utilized, however, given the speed of light, a few nanoseconds may not make a difference.

Given the spatial and temporal nature of the data and its complexity, different machine learning techniques have been employed as a computerized augmentation to traditional human approaches. A few algorithms for classification and anomaly detection have already been employed. Multi-Feature Ensemble Learning Classification Method (MFELCM) and LightGBM are classification algorithms based on AIS data and GeoTrackNet and Traffic Route Extraction and Anomaly Detection (TREAD) are techniques for detecting anomalies in maritime vessel traffic.

Because our current implementation requires the least amount of computational complexity, we cannot use models that contain computationally expensive models such as DBSCAN. Therefore we will use a classification algorithm that performs quickly and accurately on a smaller data set. Similar to MLEFCM and LightGBM, an ensemble model is the best approach for accurately classifying and reducing the noise in the data set while also being able to refine as larger amounts of data become available. A combination of Gradient Boosting, Random Forests, and Decision Trees will produce the best results with the resources we have.

7.1.6 Known Challenges

Dark Vessel Detection seemingly comes with more challenges than easy solutions. Vessels that are legally compliant and broadcasting accurate information may be difficult to detect if a receiver, web-based, or satellite based provider is delivering out-of-date information. Vessels may be detected on the mounted radar system, but their deduced location may corresponded to one-to-many different vessels in an AIS system depending on the likelihood that the time between the last received transmission corresponds to a vessel being in a given area deduced from its course and speed or other variables. Smaller vessels, using Class B AIS systems, are also not legally required to transmit their information as quickly, as noted in section 4.1.4, therefore many smaller vessels (which would likely be more preferable for illicit activity due to their increased stealth) become harder to geolocate and classify as “known” or “unknown”. Known vessels may also be transmitting AIS, but unknowingly broadcasting the wrong information, such as an incorrect MMSI, the unique identifier assigned to each vessel. This flaw can lead to errors in classification, identifying the correct “known” vessel, or maliciously be used to spoof a known vessel to avoid detection. While AIS information is primarily relied upon to determine whether a vessel is known or unknown, the radar itself has its own challenges.

Given the radar's maximum range of around 72 miles and angles of detection, the radar is susceptible to detect not only marine vessels, but anything within range of the radar. This could include flocks of birds, airplanes, large surfaced sea animals, etc. The detection algorithm should be able to identify the sort of object that the radar is detecting. There are many methodologies for this classification, but many include alternative methods such as detecting the same object using LIDAR and performing image recognition or classifying based on its general location, size, speed, and other factors and using pattern recognition to predict the object and can be time and resource expensive in tuning the accuracy of the predictions. The radar is also susceptible to varying sea conditions. Conditions such as sea spray, tilt as the vessel rides the troughs and crests of a wave at sea, and the altitude

as varying sea levels change can affect the ability to locate a vessel using the radar and calculate the on-board vessel's position relative to a detected object. The object that is being detected, for example a large ship, may also be detected with a different size based on which part of the ship the radar is detected. If the radar were to detect the bow or the stern of the ship, it may appear much smaller on the radar than if it were to detect the starboard or port side of the ship.

7.1.7 Alternative Applications

Some of the challenges in implementing a Dark Vessel Detection Algorithm also give rise to some additional applications. The range of detection of the radar, at a sufficient distance is equally likely to detect air traffic as well as marine traffic. If a radar detection system is mounted in a fixed location, the radar could be used to detect aircraft that follow a regular pattern or compared with air traffic information (which is more robust than AIS for marine traffic) to similarly detect aircraft performing illicit activities. Another challenge is vessels broadcasting invalid AIS information, such as incorrect size of the vessel or a duplicate MMSI. Using the mounted radar, vessels could detect a ship that is much larger than the broadcast AIS information, but still matched the vessel using the Dark Vessel Detection Algorithm. In this case, the algorithm could be used to create flags for Coast Guard, Naval, or other authorities so ships can be requested (or mandated) to update their AIS information accordingly, providing a more robust system.

7.2 Use Case Diagram

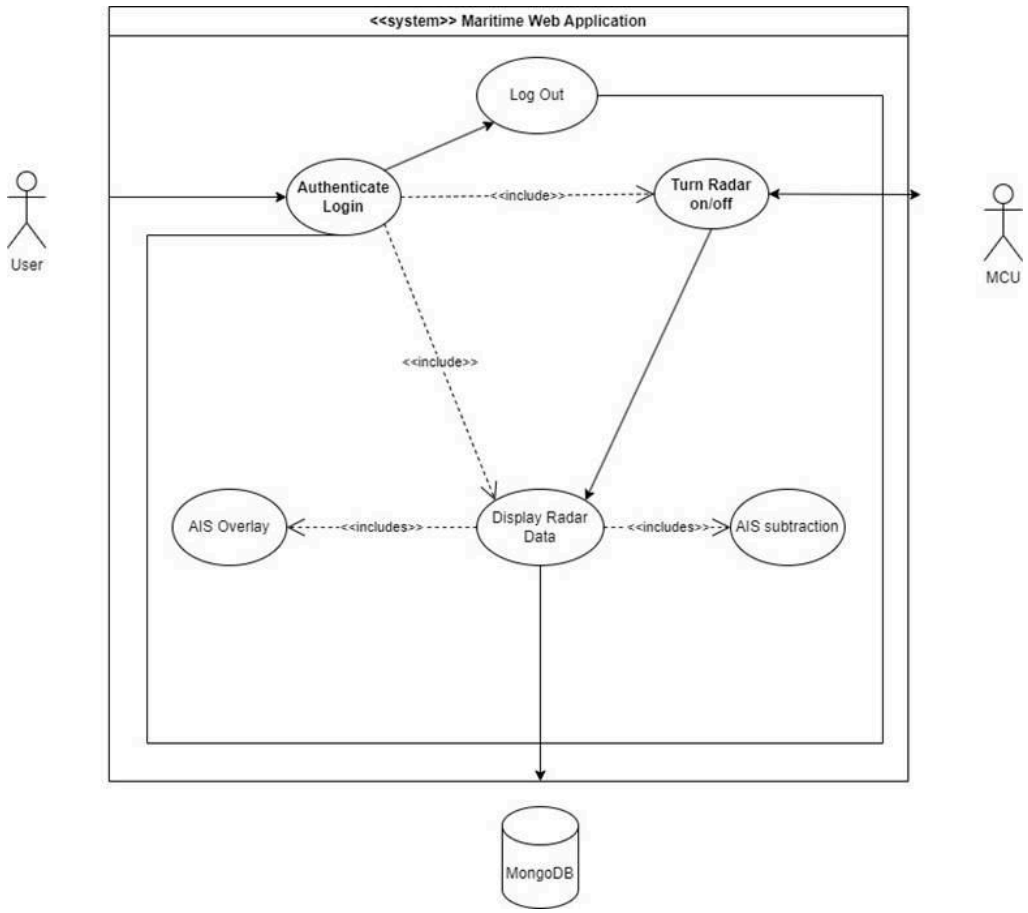


Figure 7.2 *- Use Case Diagram for Maritime Web Application*

The Use Case Diagram for the Maritime Web Application illustrates how different actors interact with the system to control and monitor radar data. The primary actors include the User (Base Station Operator), who interacts with the web dashboard, the MCU (Microcontroller) onboard the ship, which executes radar commands, and MongoDB, which stores processed radar and AIS data.

To access the system, the User must first authenticate, as authentication is a prerequisite for all core functionalities. This is represented by the Authenticate User use case, which is included in both Turn Radar On/Off and Display Radar Data use cases, ensuring that only verified users can operate the system. Once authenticated, the user can toggle the radar on or off, sending commands directly to the MCU, which executes these operations on the ship. The Display Radar Data use case allows users to visualize real-time radar detections on the maritime dashboard. This use case automatically includes AIS Overlay and AIS Subtraction, meaning that whenever radar data is displayed, the system also overlays AIS vessel locations and subtracts known vessels from the radar data to identify potential dark ships.

Additionally, all radar and AIS data processed through the system are stored in MongoDB, enabling future retrieval and analysis. The diagram effectively captures the

relationship between user actions, system functionality, and external components, emphasizing security through authentication and automation in maritime monitoring.

7.3 Activity Diagram

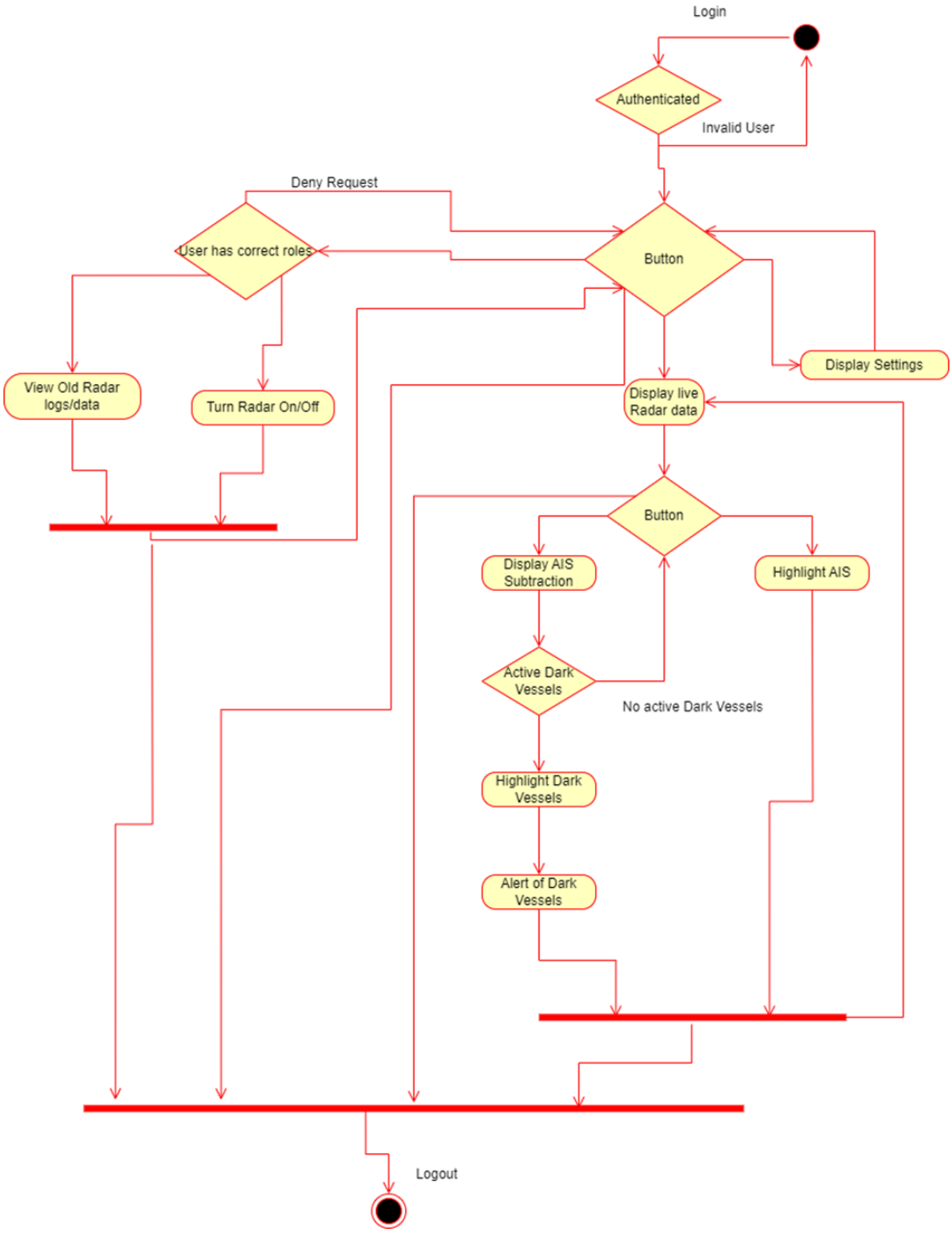


Figure 7.3 *- Activity Diagram for Maritime Web Application*

The activity diagram shown represents the primary user interaction flow within the maritime web application. Starting from user authentication, the diagram outlines how different decisions and interface actions guide the user through system functionalities. If the user possesses the correct roles, they may view historical radar logs, toggle radar activity, or proceed to interact with live radar data. Each action leads either to further decisions or returns to the main dashboard interface, allowing for continuous and flexible monitoring.

The core of the interface revolves around a decision hub ("Button"), from which the user may view live radar streams, display AIS data, or activate dark vessel detection via AIS subtraction. Once dark vessels are detected, they are highlighted and may trigger alerting mechanisms. Additionally, users can access settings at any time. After completing any interaction, the system redirects the user back to the dashboard or allows session termination through logout, shown as the end state in the diagram. This looping structure provides an efficient and real-time monitoring experience.

7.4 Component Diagram

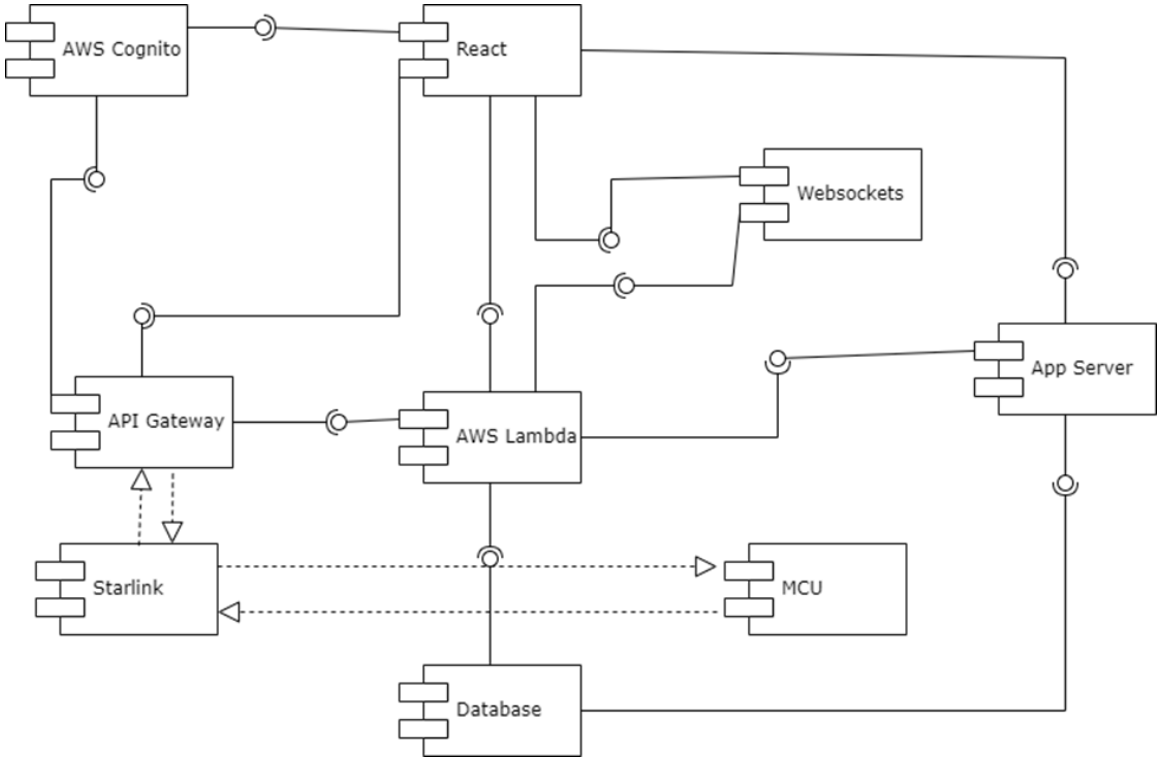


Figure 7.4 *- Component Diagram for Maritime Web Application*

The component diagram above outlines the key structural modules of the maritime radar data monitoring system, emphasizing how each software and hardware component interacts. This architecture is designed to support secure user authentication, real-time radar data transmission, cloud-based processing, and client-side visualization.

At the center of the system is the MCU (Microcontroller Unit), which is physically connected to the radar hardware. It collects raw detection data and packages it for transmission. Since the system will be deployed in maritime environments with limited connectivity, data is sent from the MCU over a Starlink satellite uplink. This network connection, shown with dashed arrows, provides a reliable ship-to-shore communication link.

Once the data is transmitted via Starlink, it reaches the API Gateway, which acts as the entry point to the cloud backend. The API Gateway validates requests and routes them to the appropriate AWS Lambda functions. Lambda performs processing, validation, or formatting of the radar data before storing it in a cloud-based Database. This ensures data persistence, enabling both real-time analysis and historical querying.

Authentication is handled by AWS Cognito, which provides secure login, token management, and optional multi-factor authentication. The React frontend communicates directly with Cognito to manage user sessions. It also sends requests to API Gateway and receives real-time updates from the WebSocket service, allowing users to monitor radar activity live. WebSockets are maintained by the backend to provide a persistent two-way communication channel between the system and the dashboard.

An App Server is shown as a deployment target or local monitoring node, which can host the web interface, facilitate development testing, or display alerts and logs. It receives updates from the backend and can interact with WebSockets and Lambda as necessary.

Each component is modular, allowing the system to remain scalable, cloud-native, and hardware-aware. By separating authentication, data transmission, processing, and presentation into well-defined components, the system ensures maintainability, security, and performance.

7.5 UI Design Goals

The UI design of the Maritime Web Application emphasizes clarity, simplicity, and responsiveness, with the primary goal of supporting rapid decision-making in real-world maritime operations. Our design philosophy is rooted in the principle that users likely operators at a shore-based station must be able to quickly interpret data, identify threats, and take action with minimal friction. As a result, the dashboard avoids unnecessary visual clutter and focuses on delivering essential information at a glance.

A map-first layout places the radar and AIS data at the center of the screen, as spatial awareness is critical to the user's understanding of vessel locations and movement. Key actions, such as toggling radar status or enabling AIS overlays, are positioned within immediate reach on the sidebar or in the top-right corner, reducing the number of clicks required to access vital features. The design borrows from common control center interfaces, using familiar placement patterns to reduce the learning curve for new users.

Consistency and feedback are also core to the user experience. Visual indicators such as a live radar status badge, distinct vessel icons (e.g., AIS ships vs. dark vessels), and color-coded overlays are used to signal system status and alert levels in real time. These

elements help ensure that users can spot unusual or critical activity at a glance in monitoring scenarios.

The UI is also designed with future extensibility and responsiveness in mind. While the initial version targets desktop display, the layout will adapt to different screen sizes to support mobile viewing. This ensures flexibility in deployment, especially as the system evolves or expands to other operational environments.

By grounding the design in usability best practices including visibility, consistency, and low cognitive load the Maritime Dashboard aims to enhance operator efficiency and support real-time maritime domain awareness with minimal training or confusion.

7.5.1 Login Display

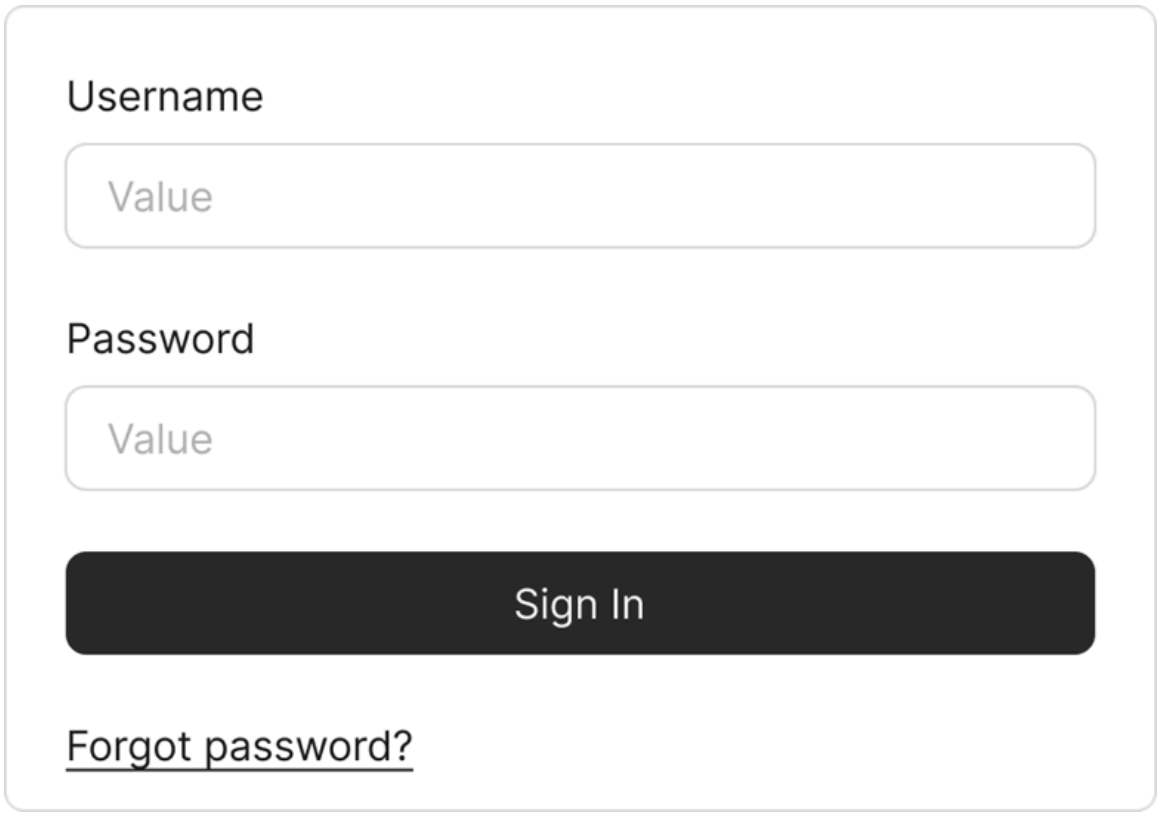


***Figure 7.5** - Login Display for Maritime Web Application*

The login screen provides a simple and intuitive interface for users to securely access the Maritime Web Application. It features two primary input fields: Username and Password, where users can enter their login credentials. Below these fields, a “Forgot Password?” prompt is included to assist users who may need to reset their login information.

The design focuses on clarity and ease of use, allowing base station operators or authorized personnel to quickly log into the system without unnecessary distractions. This authentication step is essential for protecting access to sensitive radar and AIS data, ensuring that only verified users can interact with the maritime tracking dashboard.

7.5.2 Homepage Design

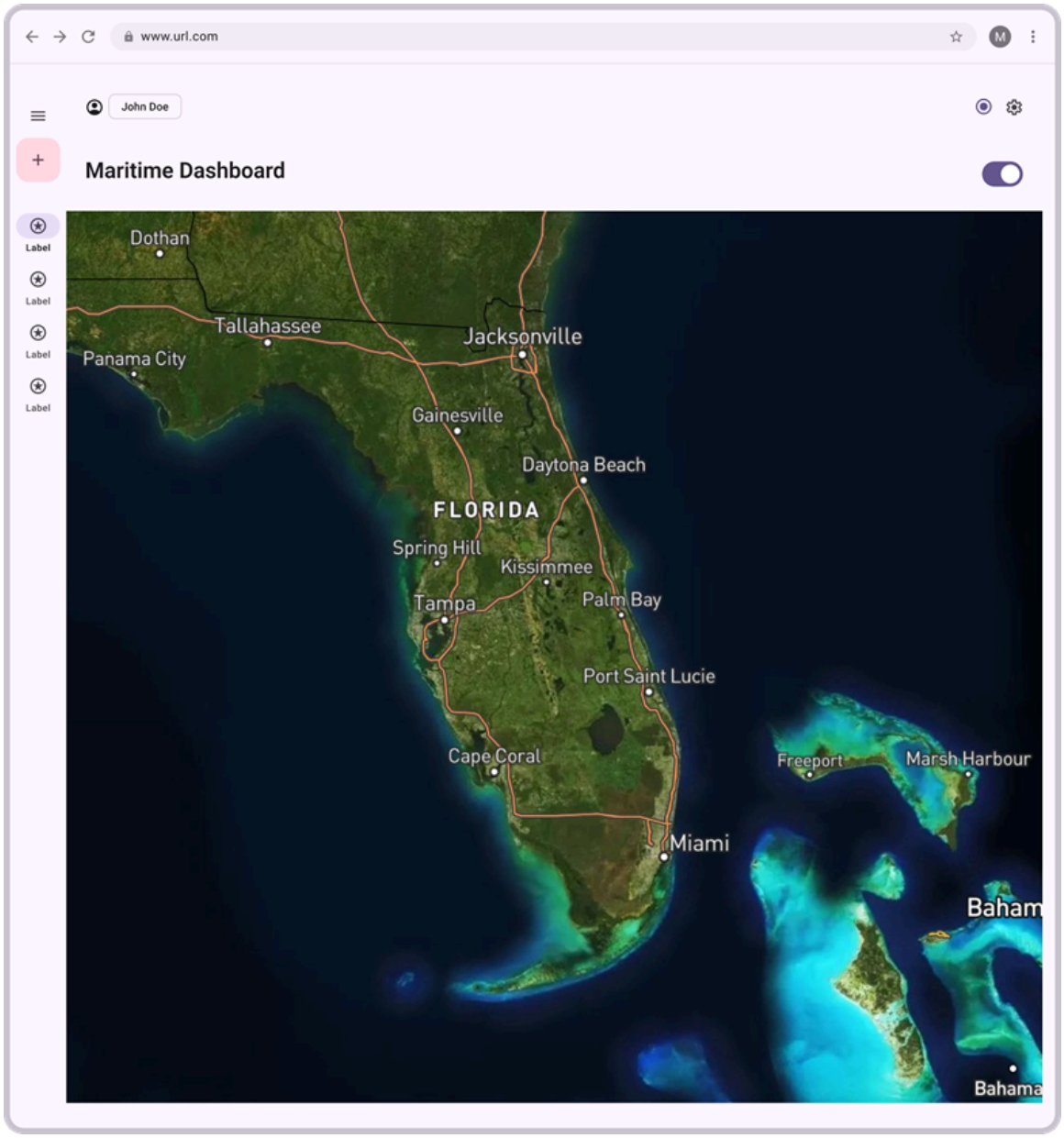


Figure 7.6 *- Homepage Display for Maritime Web Application*

The Maritime Dashboard mockup presents a clean and focused interface for visualizing vessel and radar data in real time. At the center of the dashboard is an interactive map, which serves as the primary display area for radar detections and AIS data overlays. Users can view the geographic location of tracked vessels and dark targets as they are detected, updated dynamically on the map.

On the left side of the screen, there is a vertical toolbar labeled “Labels,” which is intended to house toggle switches and filters for features such as the AIS overlay and AIS subtraction. This will allow users to quickly enable or disable different data layers based on their operational needs.

In the top-right corner, a radar toggle switch lets users turn the radar system on or off remotely. Just above this switch is a live status indicator, which will turn red when the radar is actively scanning — serving as a quick visual confirmation that the radar system is online. To the right of the status indicator is a settings icon (cogwheel), which provides access to configuration options.

The top-left corner features a user icon that represents the currently logged-in operator, helping to track account access and provide session context. This layout is intentionally minimal to prioritize the live map and reduce distractions, ensuring the operator can stay focused on vessel tracking and radar alerts.

To improve clarity and situational awareness on the Maritime Dashboard, a variety of icons and visual indicators will be used to represent vessels and radar events. These icons serve as essential UI elements that help users quickly interpret the operational status of the system and identify important targets on the map.

Radar Icon: A radar wave or pulse symbol will represent the active radar system. This icon may animate or pulse when the radar is on and will be tied to the live status indicator in the top right of the interface.

AIS Ship Icon: Vessels broadcasting AIS data will be represented with a standard ship icon, likely in blue or green. These are considered "known" vessels and will include metadata (MMSI, name, flag) on hover or click.

Undetected Icon: If the radar is on but no vessel is present in a particular zone, a subtle gray placeholder or neutral icon may be used to indicate no detections.

Dark Vessel Icon: The most important symbol, this icon will represent a vessel detected by radar but not matched to any AIS data — a potential "dark ship." This icon will appear in red or orange to draw the user's attention and may trigger an alert or appear in the system log.

These icons not only provide quick recognition but also help guide user decisions and responses. Consistent color-coding and clear symbology ensure that operators can focus on critical detections without needing to analyze raw data in real time. As the interface evolves, additional icons may be introduced for alerts, zoom level markers, or vessel classifications.

7.6 Security Design Goals

Beyond basic user authentication, the Maritime Web Application requires careful consideration of broader security practices to protect sensitive vessel tracking data, prevent unauthorized access, and ensure system integrity across both shipboard and shore-based deployments. Because this system transmits radar detections and AIS data over public networks,including Starlink and cellular, protecting this communication pipeline is essential.

To prevent interception or tampering of data in transit, the system will enforce end-to-end encryption using HTTPS with TLS. All API calls between the ship’s processing unit and

the cloud backend will occur over secure protocols to prevent man-in-the-middle attacks and ensure data confidentiality. This applies to both telemetry data (radar/AIS) and API endpoints used by the frontend dashboard.

Internally, the application will employ role-based access control (RBAC) to ensure users can only access data and functionality appropriate to their role. For example, a general operator may have permission to view radar data, but only an admin may be allowed to activate or deactivate the radar system. These permission levels will be enforced both in the frontend and backend for defense in depth.

To further strengthen backend security, database credentials and API keys could be stored using environment variables rather than hard coded into the application. This prevents secrets from being exposed in version control or frontend bundles. Additionally, access to the Database instance will be locked down using IP whitelisting and, if cloud-hosted, VPN tunnels or SSH certificates will be required for remote access.

In future versions of the system, we also plan to implement logging and auditing tools to track access. These logs will support post incident analysis in the event of a security breach or anomaly.

Overall, security in the Maritime Web Application goes beyond login credentials. From the protection of data in transit to the enforcement of access controls and proper handling of credentials, these measures are designed to safeguard maritime domain awareness operations as the system scales.

7.7 Scalability and Maintainability of Maritime Web App

Scalability and maintainability are core principles in the architecture of this maritime web application. As the system is expected to handle increasing volumes of radar data, user activity, and real-time updates in future phases.

One of the key decisions contributing to scalability is the adoption of a serverless architecture. Services such as AWS Lambda and API Gateway automatically scale based on demand, ensuring that backend processing capacity adjusts to workload without manual intervention. This is particularly useful for high-volume operations like streaming radar detections or triggering multiple concurrent alerts. Additionally, the use of AWS Cognito for authentication eliminates the need to manage login state or session scaling, as it handles user identity management independently of the backend application logic.

The frontend, built with React, is inherently scalable due to its component-based structure. Each interface element (such as radar displays, alerts, or user dashboards) can be developed, tested, and maintained independently. This supports incremental updates, simplifies onboarding new developers, and reduces the risk of system-wide regressions when changes are made.

On the backend, microservice-like separation via Lambda functions enables independent testing and deployment of individual features. For instance, a Lambda function handling user alerts can be updated without impacting the one processing raw radar input. This makes the system easier to maintain, debug, and expand over time.

Data flow within the system has been abstracted to support a pluggable database model, meaning the backend logic can remain largely unchanged if the team switches from one database provider to another. This adds flexibility in both scaling storage and adapting to project specific infrastructure requirements.

Future scalability is further supported by planned integration with WebSocket based real-time communication, which allows the server to push updates to users as soon as new radar data is available. This model reduces polling overhead and ensures that bandwidth is used efficiently.

Together, these design decisions create a system that is not only equipped to handle present needs, but also prepared to grow in complexity and scale without requiring significant architectural changes.

7.8 Web App Plans and Goals

As the system continues to evolve, the development team has outlined a feature roadmap to guide ongoing improvements and ensure timely delivery of high-priority capabilities. The roadmap is organized into short-term, mid-term, and long-term milestones, providing a structured vision for future iterations.

7.8.1 Short-Term Goals

- Finalize the integration of the AWS Cognito authentication system with role-based access.
- Complete the migration of the backend from the local Express.js prototype to fully serverless AWS Lambda functions.
- Expand the user dashboard to support more detailed vessel tracking and log access.
- Implement and test WebSocket functionality for real-time radar update broadcasting.

7.8.2 Mid-Term Enhancements

- Integrate an AIS overlay system to allow users to cross-reference radar detections with known vessel data.
- Develop user-facing alert logic to support manual and automatic tagging of dark vessels.
- Introduce a settings panel to allow customization of interface behaviors and alert thresholds.
- Build out full end-to-end logging for incoming data, user activity, and system events.

7.8.3 Long-Term Expansion

- Introduce predictive analytics based on radar movement patterns to forecast potential vessel paths.
- Enable secure data sharing between authenticated organizations and authorized third parties.

This roadmap ensures that the system not only meets its initial design and performance goals but also evolves to accommodate user feedback, scalability demands, and extended functionality in real-world maritime environments.

7.9 Embedded System Software Design

The embedded software system in our project is responsible for managing all onboard operations, including capturing sensor data, system control, and communication with external systems. This codebase runs on a Raspberry Pi 4 Model B, which serves as the central controller for the MDAS unit. The software is designed to efficiently interface with radar outputs, monitor onboard sensors, manage outbound communication through the Starlink satellite system, and trigger additional subsystems such as the gimbal-mounted camera.

The embedded system must remain lightweight, efficient, and reliable so it can operate on its own in a marine environment with limited power availability. It will also handle scheduling for sensor polling, logging, data formatting, and uplink transmissions.

7.9.1 Software Architecture

The embedded software follows a modular architecture built around Python scripts that execute specific system tasks. This modularity allows for greater maintainability and separation of functions, particularly for radar parsing, power monitoring, gimbal control, and data transmission.

Table 7.2 *- Proposed Modules*

Proposed Modules	Purpose
radar_interface.py	Parses radar data from NMEA 2000 interface or USB serial.
gps_handler.py	Continuously updates vessel coordinates.
gimbal_control.py	Sends control commands to gimbal upon detection triggers.
power_manager.py	Monitors system voltage and current.
data_uplink.py	Manages queued data uploads via Starlink interface.

The main script (main_controller.py) acts as the scheduler and orchestrator, polling sensors and triggering respective functions based on real-time conditions.

7.9.2 Flow of Execution

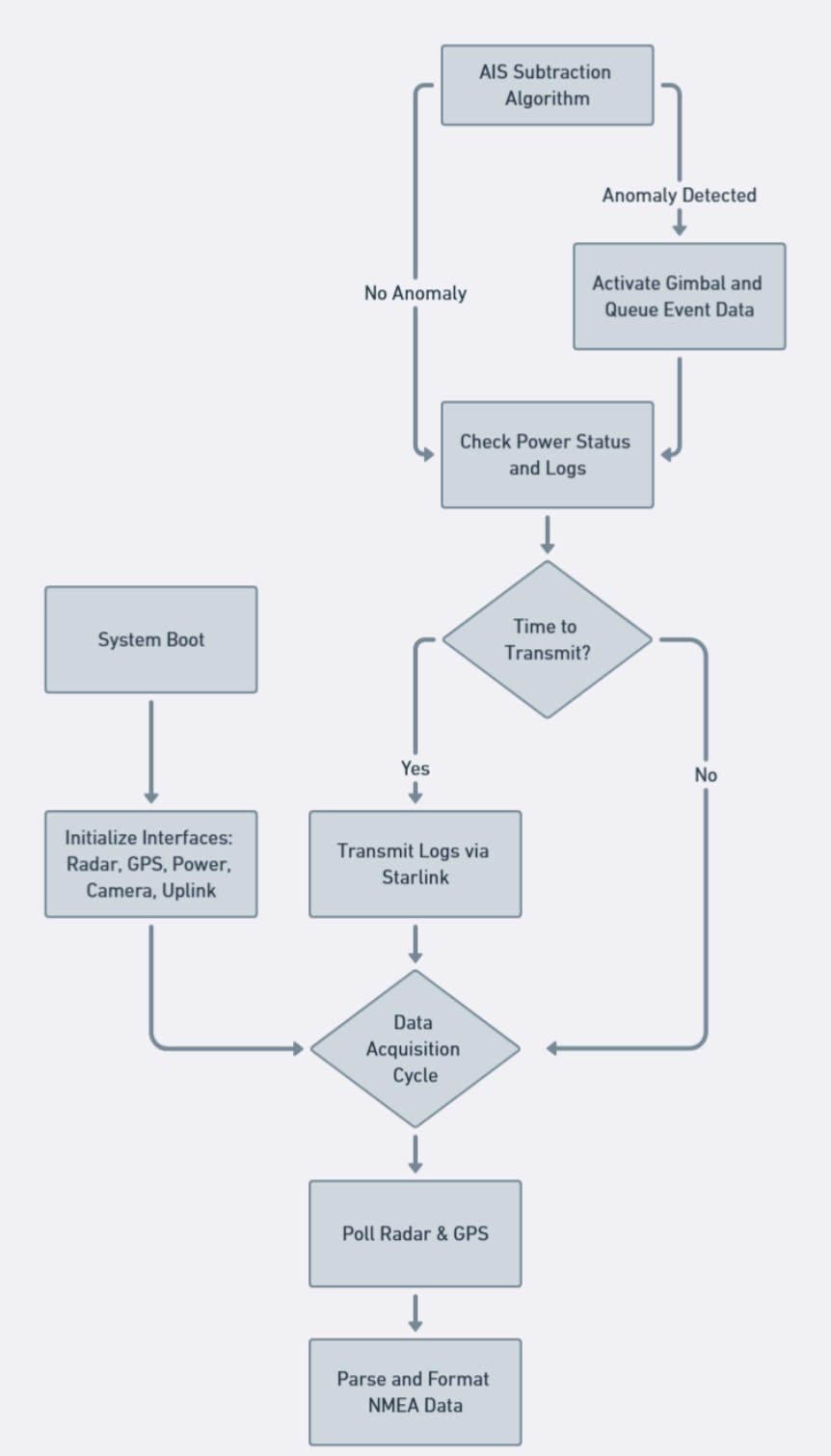


***Figure** 7.7 - Embedded System Software Logic*

The data acquisition loop kicks off the system's continuous monitoring cycle. Once all interfaces are initialized (radar, GPS, camera, etc.), the Raspberry Pi enters the main logic loop, which repeatedly performs polling, data formatting, anomaly detection, and telemetry updates.

7.9.3 Data Structures

The onboard system uses Python dictionaries and queues to manage asynchronous data flow. One example is the event data queue, which stores potential threats or system events before transmission [4]. This structure allows for quick JSON serialization for uplink through Starlink's network interface.

To further streamline data handling and reduce latency, each event in the queue is timestamped and assigned a unique event ID. This facilitates efficient tracking of messages during transmission and in post-mission analysis. In future implementations, these data structures could be extended to support prioritization flags, ensuring critical alerts are transmitted first when bandwidth is limited.

7.9.4 Communication Handling

As discussed in chapter 3, the embedded software supports multiple protocols and interfaces:

- **NMEA 2000 via USB/Serial** for radar and GPS parsing
- **PWM/UART** for controlling the pan/tilt gimbal
- **I2C** for battery and voltage sensors
- **Wi-Fi/Ethernet** to interface with Starlink for remote transmission

Each interface is configured asynchronously to ensure non-blocking operations, and data packets are formatted into standard telemetry frames before transmission.

7.9.5 State Machine for Anomaly Detection

We propose a simple finite state machine (FSM) to manage detection logic:

States:

- IDLE: Monitoring radar input
- ANALYZE: AIS cross-check in progress
- TRIGGER_GIMBAL: Anomaly detected
  LOG_EVENT: Package event details
- TRANSMIT: Send data to command center
- RESET: Return to IDLE

This FSM allows for robust tracking and handling of unusual vessel signatures while minimizing CPU load. The finite state machine design supports sequential, deterministic state changes avoiding race conditions or simultaneous actions that are most unwanted in time-critical maritime security. Every state includes entry and exit conditions to avoid system response ambiguity. For example, the ANALYZE state involves a timeout specification to return IDLE in case of indecisive AIS confirmation. Similarly, TRANSMIT will repeat up to three times before it records the event as a partial failure. The state machine design is also scalable to accommodate new behaviors such as image capture or remote override in subsequent development stages.

7.9.6 Failsafes and Watchdog

The embedded software includes watchdog logic to ensure system uptime. If a subsystem fails to respond or data becomes stale, the controller resets that module. Logging is also redundantly written to SD card storage in case of communication loss. These failsafes are essential in an autonomous naval deployment where there is not always immediate human intervention available. The watchdog timer is configured with subsystem-specific thresholds to enable differentiated recovery routines depending on whether or not a failure has occurred in the radar pipeline, gimbal controller, or network module. In a full, a soft reboot is triggered to enable return to normal operation without losing memory or logs. Redundant SD card logging enables data integrity to be maintained even in case of Starlink uplink failure, providing a backup record of system activity. As part of future development, an external heartbeat circuit can be added to further harden the system against power anomalies and silent failures.

Chapter 8: System Fabrication/Prototype Construction

8.1 MDAS System Fabrication

The fabrication of the Maritime Domain Awareness System (MDAS) was guided not only by the vision of the engineering team but also by direct input and operational requirements specified by our sponsor. From concept to construction, the system's design was focused on creating a self-contained and self-sustained maritime surveillance unit. With its main goal of being able to detect and track vessels operating without AIS transponders, the system integrates a solid-state marine radar, embedded processing, satellite communication, renewable energy generation and optical imaging components to ensure its success. Fabrication required careful consideration of marine-grade materials, energy efficiency, and environmental resilience to ensure long-term autonomous operation in harsh ocean conditions. The subsystems were researched and some parts were assembled into a functioning prototype.

8.1.1 MDAS System Prototype

Initially, the design process for the MDAS mechanical structure was clouded with uncertainty. While the functionality of the system was well understood: integrating a marine radar, wind turbine, satellite communication, and embedded electronics, the form that this structure should take remained unclear. With multiple components of varying shapes, sizes, and operational needs, the challenge was not only fitting everything physically but doing so in a way that was structurally sound, easily installable on different vessels, and optimized for harsh marine conditions.

To begin tackling this task, the team produced a conceptual blueprint (Figure X), sketching out a rough mechanical layout.

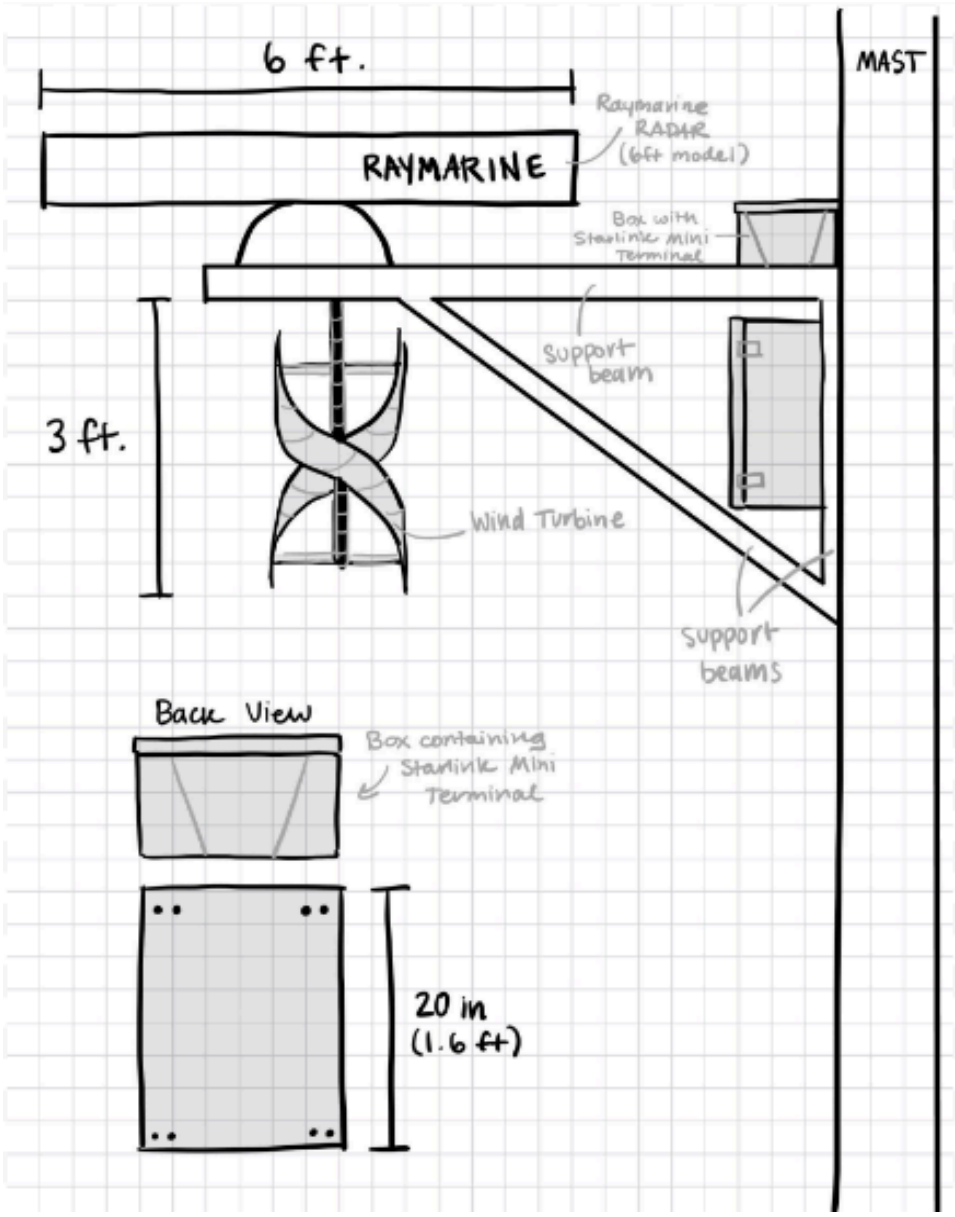


Figure 8.1 *- MDAS Initial Prototype Sketch*

This phase wasn't about precision but rather visualizing relationships between subsystems. The Raymarine 6-foot open-array radar was placed at the top to ensure maximum visibility and signal range. Beneath it, a compact horizontal platform was envisioned to mount the base of the radar, while angled support beams provided structural integrity and braced the unit against forces from wave and wind loads. Directly underneath the platform, the vertical wind turbine was suspended in a vertically centralized position, aimed at capturing omnidirectional wind currents. This placement reduced the risk of external forces from the radar or mast structure and kept the center of gravity balanced.

Along the back of the structure, a small electronics box to house the Starlink Mini terminal and other embedded hardware was positioned for easy access. The sketch also

included a simple rear platform, slightly elevated, to provide the Starlink dish with a clear view of the sky while remaining integrated with the structural frame. Each component in this drawing was placed with purpose: the radar needed to remain stable and unobstructed, the turbine required space and clearance from surrounding airflow obstructions, and the electronics needed to be protected yet accessible.

From this initial blueprint, the design was brought into Fusion 360 and developed into a more accurate and interactive CAD model. Modeling allowed the team to rapidly test component spacing, identify potential issues, and evaluate the overall integration. It also helped simulate how the system would look and function when mounted on a ship's mast, offering valuable insight into airflow, shading, and the physical footprint of the unit.

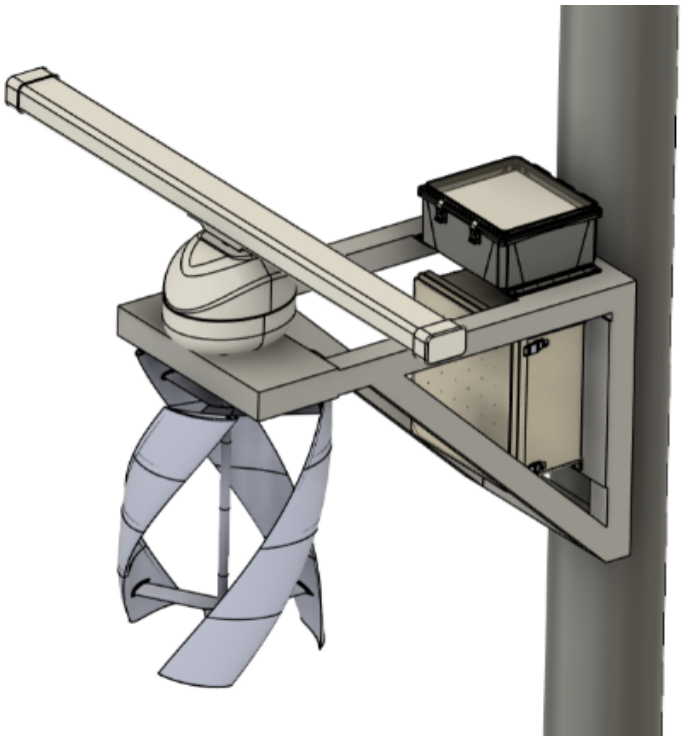


Figure 8.2 *- MDAS Initial Prototype CAD*

While the initial blueprint and CAD modeling helped clarify the system's spatial organization and component interactions, these early concepts were ultimately presented to our sponsor for review and were met with skepticism. The proposed design, which mounted the entire system to a ship's mast using a cantilevered bracket structure, raised concerns about long-term reliability. The cantilever approach, although conceptually possible and space-efficient, posed several mechanical risks when considered in a real maritime context. Questions were raised about the system's stability under constant vibration and weather conditions that would impose significant torque on a single-sided mounting solution. It was deemed too difficult to install due to correct alignments of the brackets.

The sponsor ultimately rejected the mast-mounted prototype, citing structural fragility and maintenance concerns. They emphasized the need for a more stable, self-contained

configuration, one that could be securely bolted to a flat deck surface and withstand harsh sea states without relying on a vessel's mast infrastructure.

This feedback marked a major turning point in the project. With the original mounting strategy no longer viable, the team was forced to return to the drawing board and reimagine the system's mechanical design from the ground up. The next phase of development focused on creating a standalone, deck-mounted structure that preserved the functionality of the original concept while improving mechanical robustness, modularity, and deployment flexibility.

8.1.2 MDAS System Redesign and Structural Revision

The team shifted focus to a deck-mounted implementation, prioritizing stability, modularity, and adaptability to different ship configurations. The revised concept centered around a compact, self-contained frame that could be rigidly bolted to a vessel's superstructure or deckhouse, rather than relying on pre-existing mast infrastructure. This shift not only improved overall stability by lowering the system's center of gravity, but also reduced stress on the structure during ship movement and made maintenance and installation much easier. The key to the structure's redesign was the "MDAS 5 Beam Frame"

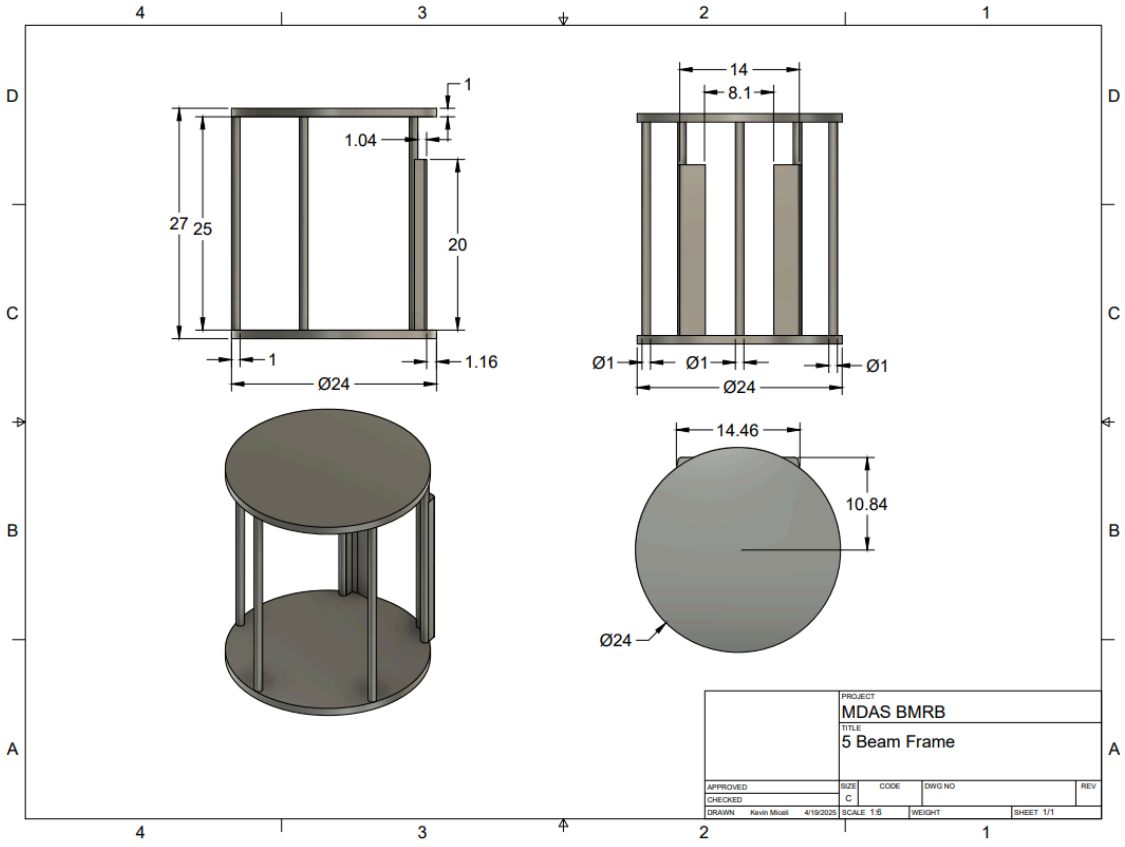


Figure 8.3 *- MDAS 5 Beam Frame CAD*

The frame utilizes a symmetric cylindrical layout featuring two 24” diameter circular plates connected by five evenly spaced vertical support beams. This structure was originally designed to support a vertical-axis wind turbine but was adapted to serve as the core mounting platform for the entire MDAS system. The beams, each measuring approximately 14” in length, are the vertical spacing between the plates. Keep in mind, this dimension is flexible, as the final height of the wind turbine is still to be determined. Through research, the team identified several potential vertical wind turbines with heights ranging from 40 to 60 inches, meaning the frame can be scaled as necessary to meet final component selection.

The base of the frame incorporates vertical flanges and angled reinforcements, each precisely dimensioned for integration with the turbine tower and electronics enclosure. Mechanical fastening points are reinforced with flanged bolt holes, enabling secure, through-deck mounting on a wide variety of marine surfaces. This deck-penetrating mounting strategy aligns with practices used on many existing research vessels, which often feature flat-roofed deckhouses specifically designed to accommodate additional scientific equipment. By adopting a multi-point securing strategy, the system can better resist vibrational loads and maintain positional stability during rough sea conditions.

To visualize how each core subsystem would be integrated into the updated frame, the team developed a series of CAD models, each version showing increasing addition of components to the structure.

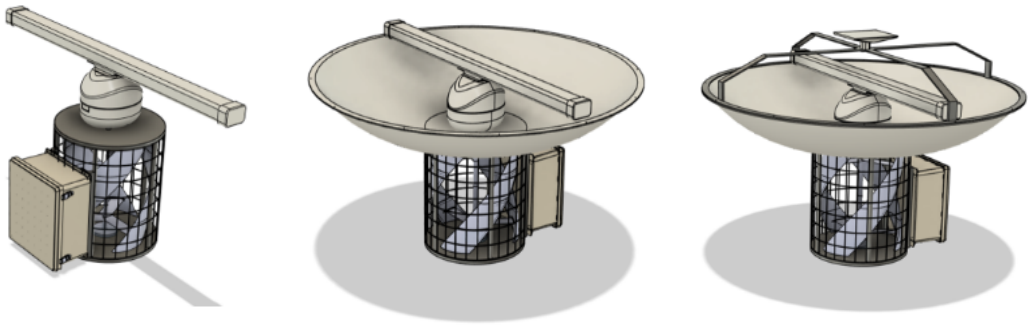


Figure 8.4 *- Iterative CAD models showing the step-by-step addition of MDAS system components*

After the new frame was developed, the team explored ways to protect the vertical-axis wind turbine by implementing a mesh shroud or cage. This addition serves to shield the moving blades from debris and external contact while still allowing for unrestricted airflow and efficient turbine operation.

Furthermore, the Raymarine radar system was added into the model. It was mounted directly to the top of the frame, aligned with the center to maintain balance and ensure optimal signal visibility. This confirmed that the structure could fully support the radar's size and weight while still leaving adequate clearance for cabling and reinforcement hardware beneath.

Next, the electronics enclosure was added and digitally mounted to the side of the frame using simulated brackets. This location was selected for accessibility and protection, ensuring that internal systems such as batteries, controllers, and communication modules could be serviced with ease while remaining protected.

A large protective cone was added to the top of the model to function as a salt spray shield and wind deflector. Finally, a mount was included for the Starlink Mini module, positioned upwards on structural beams to maintain a clear sky view and support continuous satellite connectivity.

8.1.3 MDAS System Structural Material Design

Material selection played a critical role in ensuring that the MDAS platform could endure the harsh and variable conditions found in marine environments. Early designs investigated marine-grade aluminum alloys like 5083 and 6061 for the system's beams and framing, given their corrosion resistance and good strength-to-weight characteristics. However, despite their favorable properties, concerns about long-term saltwater exposure, galvanic corrosion, and fabrication limitations led our team and sponsor to consider alternatives. After consultation, the recommendation was made to transition toward Fiber-Reinforced Polymers (FRPs), particularly epoxy-glass and polyurethane-glass composites, which offered a more customized balance of mechanical resilience, corrosion resistance, and lightweight behavior.

The roof panel where the radar dome is mounted was designed using Coosa's Bluewater FRPU board, commonly used in commercial boats and marine enclosures. Initially, a polyurethane topcoat was chosen for UV and color protection, but this was later changed to vinyl ester fiberglass resin. This change was based on a cost-performance tradeoff and fabrication feasibility, as vinyl ester resin was more readily available and sufficiently weather-resistant for our specific application.

At one stage, the team had anticipated the need to design a custom electronics enclosure but it was later determined that a commercially available weatherproof case would suffice. This simplified the design process significantly, as only the mounting solution and frame integration of the enclosure required custom work. The enclosure case itself, being a prefabricated IP-rated product, still needs to be chosen for its proven durability, waterproofing, and ease of integration with off-the-shelf cabling and seals.

The base plate, which will anchor the MDAS system to the vessel, was initially selected to be made of 316 stainless steel. Similarly, the radar mounting plate was intended to be fabricated from FRPU to maintain consistency with the surrounding frame materials. However, it is important to note that these materials have not yet been finalized or fabricated, as the actual build of the MDAS unit is still pending. These selections were just based on research, availability, cost and sponsor feedback during the pre-development phase.

Finally, a coating system was planned to further enhance material performance. Epoxy primers were chosen to act as a moisture barrier, while polyurethane or vinyl ester topcoats were applied for UV protection and durability. Antifouling coatings were not

included, as the MDAS unit is designed for elevated deck mounting rather than submerged deployment.

8.2 Custom 12DC to VCM Cable Design

As part of the power delivery path to the Raymarine Cyclone radar, a custom cable was constructed to bridge our portable DC power source and the radar's interface module (VCM100). This task involved the understanding of electrical safety guidelines provided by Raymarine and required the integration of appropriately gauged wire, standardized connectors, and an inline fuse rated for high current protection.

The VCM100 radar controller accepts input of 12–24V DC via a terminal block connector. We prototyped with a 12V Westinghouse portable battery bank as the primary power supply, which supplies via a common 5.5mm x 2.1mm DC barrel jack. To complete the circuit between the Westinghouse battery and the VCM, we needed a cable that transitioned the DC barrel output to pigtail leads suitable for terminal screw insertion. According to Raymarine installation guidelines, any 12V power to the VCM should be protected with a 20A inline fuse as close to the power source as practicable. This is to avoid too much current burning downstream electronics in the event of a short or overload. Raymarine also recommends 14 AWG wire as minimum for installations operating at 12V.

8.2.1 Cable Fabrication Process

To fabricate this cable, we sourced a 5.5mm DC male barrel connector pre-terminated to two wire leads. These leads were short, lightweight, and not rated for 20A currents, so they were extended and reinforced using 14 AWG stranded wire for both the positive (red) and ground (black) paths.

An automotive-grade inline fuse holder was then spliced into the positive line, approximately six inches from the DC connector. A 20A blade-style fuse was inserted into the housing, completing the high-current protection requirement. Heat shrink tubing was applied to all soldered joints to minimize risk of shorting and improve mechanical strength. On the VCM side, the exposed wire ends were stripped and directly inserted into the VCM's terminal block labeled "POWER IN," with positive and negative connected according to the polarity indicators on the PCB. The power cable to the VCM is shown here:

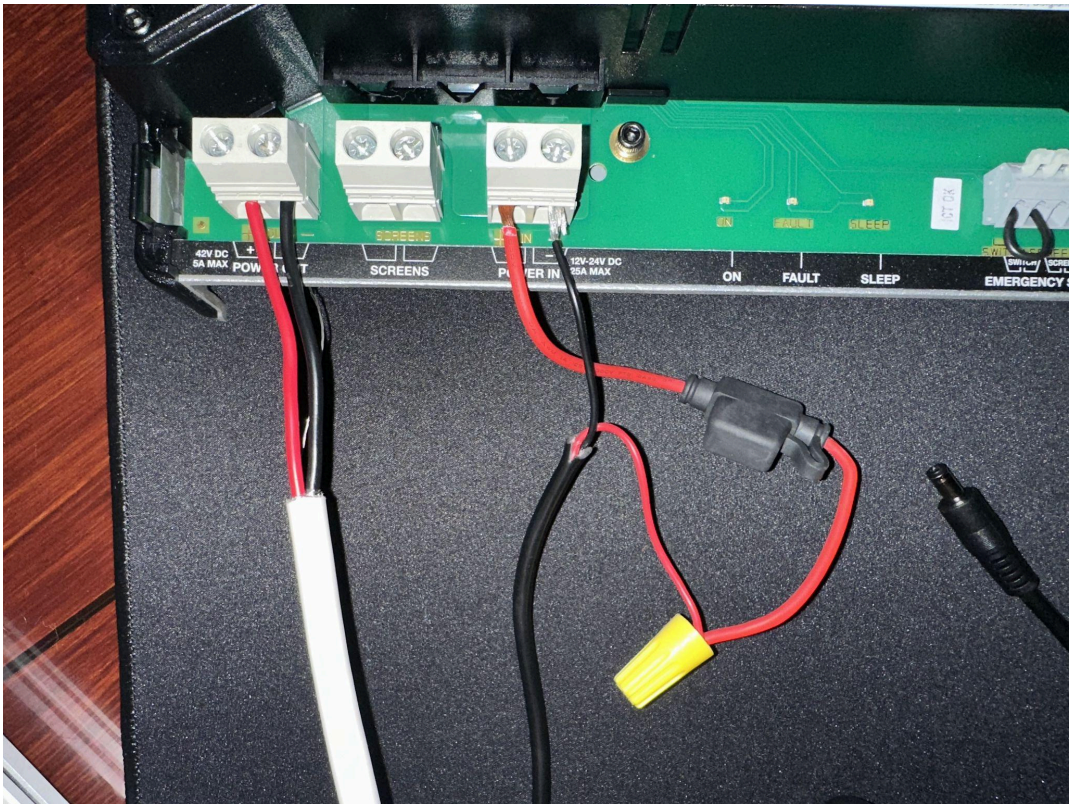


Figure 8.5 *- 12VDC to VCM fabricated cable*

8.2.2 Observations and Challenges

Initial testing of the DC-to-VCM cable confirmed that the cabling reliably delivered sufficient voltage and current from the Westinghouse battery to the VCM100 module. After approximately 35 to 40 seconds of consistent 12V draw, the VCM100 transitioned from a solid yellow indicator (signifying sleep mode) to a blinking yellow state, which indicates that the radar entered a true standby condition and is awaiting command. This successful transition validated that our custom-fabricated cable, including the 20A inline fuse, was not only functionally sound but aligned with Raymarine's expected startup behavior. This marked the first verifiable factor of power successfully moving from the 5.5mm DC jack, through our inline-fused 14 AWG wiring, into the VCM100, and ultimately being passed forward to the radar unit. During this process, there were no signs of overheating, power loss, or electrical anomalies, thus verifying the safety and effectiveness of the cable we manufactured.

8.3 Optics System

8.3.1 Lens System

For prototyping the system's imaging system, a fixed Galilean telescope system was assembled along an optical rail with a CMOS sensor placed at the end of the rail. This prototype system was to establish a foundation for the final project, and showcase the basic working principles of the system. The system consisted of a 150 mm biconvex lens

and a 25 mm biconcave lens, with a ThorLabs CS165CU CMOS camera sensor. The lenses were spaced using a dial caliper for precise measurements, as well the measurements along the optical rail. Lens tubes were used to decrease light entering the system, as well as a 3D-printed lens tube coated in Black 3.0 paint to prevent as much internal reflection as possible. An iris was placed in front of the objective to try to prevent aberrations along the edges of the image. Upon original construction, the field of view was incredibly narrow, and the optimal viewing distance was 7+ km away. Due to the constraints of the environment, a relay lens was added to be able to get an image at a shorter distance of two to three meters. There were two versions of this system tested in the lab. One version had a 25 mm biconvex lens placed in front of the sensor, while the second version had a 35 mm plano-convex lens in front of the sensor. These two setups are shown below:

Figure 8.6 *- Prototype Lens System v1 and v2*

While these two systems both produced an image, the rendition with the biconvex relay lens produced a clearer image at the specified distance of two meters. In addition to the two lens systems, two separate lens tubes were also 3D printed to accommodate for the two potential options. These tubes were modeled in Autodesk Fusion and printed, with the lengths corresponding to the respective focal length distances in the two setups. These tubes were printed with black PLA filament and then covered in Black 3.0 paint, a paint that is made with Black Magick pigment and designed to absorb 99 % of visible light. This was done to diminish the amount of ambient light entering into the system. These lens tubes are shown below:

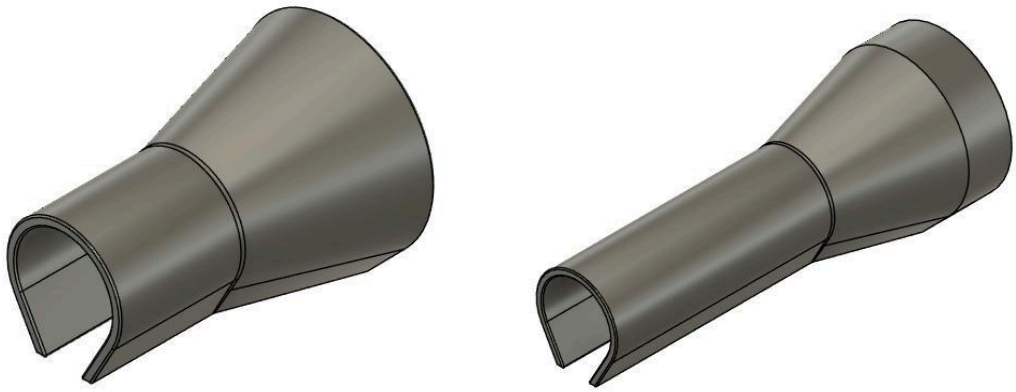


Figure 8.7 *- Lens Tube Models*

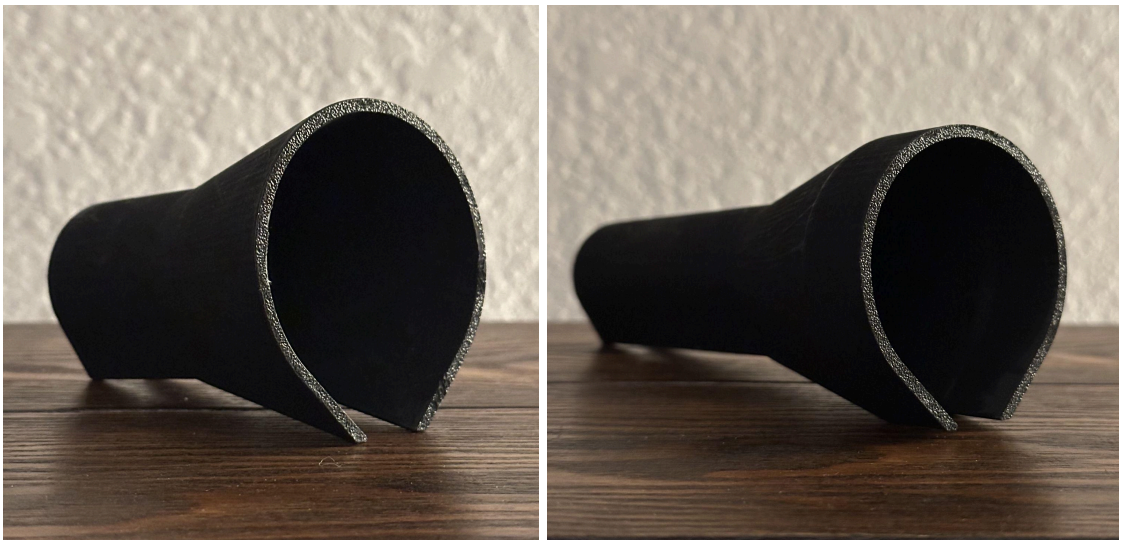


Figure 8.8 *- Printed Lens Tubes*

This system was a great start to the prototyping phase, but did not produce a clear image as intended. This led to different lens systems being explored through Zemax simulations, such as the achromatic doublet system. There were multiple renderings of varying lenses that were tested, looking at three key features: the spot size, the simulated image, and the overall system length. These three were crucial in determining whether or not the system would be successful. After looking at multiple designs, we determined that a two lens system would not be successful, because the image would not focus properly onto the CMOS sensor. Therefore, a few achromatic doublet lens setups were simulated of varying focal lengths and diameters. In the end, we determined that the 300 mm achromatic doublet lens would be the best option, for both the narrow simulated spot size as well as the relatively clear simulated image.

8.3.2 Gimbal

Originally for the gimbal portion of this project, we started with the Adafruit pan-tilt gimbal kit. This model connects through a Raspberry Pi, which was what we were originally using to connect all the parts of the MDAS system. Through some testing, it was determined that this model would not be able to work with this project. When attempting to mount the CMOS sensor and lens system, it was too heavy to be supported by the bracket, and it also did not fit correctly. In addition to this, we determined that the Raspberry Pi would not be compatible with our system, meaning that we would not be able to properly implement the gimbal. Taking these testing outcomes, we found the SCORP-C 3-Axis Mirrorless Gimbal, which solved all of the previously mentioned issues.

Chapter 9: System Testing and Evaluation

9.1 Prototype Setup

The prototyping for the Maritime Domain Awareness System (MDAS) was a seemingly straightforward effort initially which involved hardware assembly, network configuration, software provisioning, and component testing. This proved to be a much more difficult and involved task as construction began. This chapter details the development of our initial prototype and the steps taken to build a proof-of-concept for radar-based vessel detection and communication. Due to the complexity of the system, a phased construction approach was used, focusing on each modular subsystem, and an attempt to integrate in a methodical way. Our prototyping involved one Intel NUC standing in for the eventual Raspberry Pi, one HP laptop as the inland base station, a Raymarine Cyclone radar powered through its VCM100, a Westinghouse 12 VDC power bank, a Starlink Mini satellite terminal, and a Samsung S24 smartphone reserved for future remote-command tasks.

Figure 9.1 *- Complete Prototype Setup*

9.1.1 NUC-Based Prototype Platform

We used an Intel NUC running Windows 11 as the central hub for initial prototype construction. This offered us greater flexibility and computing power during development than a Raspberry Pi. The NUC was configured with Python, Visual Studio Code, and supporting packages for networking, serial communication, and socket programming. The decision to use Windows allowed a familiar OS, and the conformance to the Raymarine SDK which claims to need a Windows machine.

The NUC also served as the transmission node for radar metadata, system diagnostics, and command relays. Several Python-based tools were written and tested to emulate core behaviors including radar activation, data pushing, and telemetry dispatch. These scripts are included in Appendix D and demonstrate modular client-side logic. This data was seen through tools like WireShark and the supplied software development kit's command line interface.

9.1.1.1 Software Setup

To enable radar initialization, we installed TFTPD64, a lightweight DHCP server that allows the NUC to allocate an IP address to the radar. This emulated the role of a Raymarine multifunction display (MFD), which normally handles radar configuration. Python-based scripts were also written to monitor IP assignment logs and automate command dispatch to the radar once connected.

- Development stack: VS Code + Python 3.13.3 installed on both the NUC and the HP laptop for identical development environments.
- DHCP/TFTP service: TFTPD64 runs as an elevated task on the NUC. The DHCP scope is 198.18.0.2 with 255.255.240.0 subnet; the radar is expected to claim the first free lease.
- Radar SDK: Two encrypted LightHouse SDK archives supplied by Raymarine Support were extracted to 'C:\RaymarineSDK'.
- WireShark: Used to track network packages sent and received to the radar from the NUC
- Tailscale: Created a link from the base station to the NUC using a similar server, bypassing starlink's limited Carrier-Grade NAT (CGNAT) properties.

9.1.1.2 Account and Credential Management

During prototyping a single Windows user was created for the NUC so that all radar-related software lived in one profile, separate from the factory admin account. The password is documented in the team's shared files. VS Code, Python, Tailscale, the Raymarine SDK, and TFTPD64 were installed under this profile, if possible and the default "C:\Documents\MDAS" folder is the location for every config file (DHCP scope XML, SDK examples, and test logs). No further hardening (SSH keys or vaults) have been applied yet as the point was simply to isolate prototype work and keep the setup reproducible for any teammate who needs to operate on the NUC.

9.1.2 HP Base Station Integration

The HP laptop serves as the back-end base station in our MDAS network and command relay. This device receives data transmissions from the NUC and will eventually process radar metadata and build the application used by military members. We installed the same development stack as on the NUC to maintain compatibility. The HP base station ran Python servers configured to accept incoming packets, log connection attempts, and echo

confirmation of received messages. We developed test scripts to simulate future integration of radar tracking messages, and these were run on the base station during LAN and Starlink experiments.

Logging was implemented to track each connection and message exchange. When data is received from a NUC, the server logs the message size, sender address, and content into a persistent file with timestamps. This architecture ensures full traceability for all prototype communication exchanges.

9.1.2.1 Software Setup

The HP 17 (Windows 11) mirrors the NUC tool-chain, VS Code and Python 3.13.3, so scripts can be executed or debugged on either machine without much modification. All scripts can be launched and maintained independently from either machine, with no OS-specific dependencies. Server logic includes error handling for timeouts, message decoding, and controlled shutdown through a custom-built GUI that serves as a stop to the server upon its button press. The base station also logs session-level events such as startup, NUC health/ system data, shutdown, and connection errors to support debugging and reproducibility.

- Development stack: VS Code + Python 3.13.3 installed on both the NUC and the HP laptop for identical development environments.
- Configuration: Port forwarding and Tailscale will have been set up for different communication protocols during prototyping.

9.1.2.2 Port-Forwarding and Wireguard

We initially tested communication using basic port-forwarding and DDNS for direct TCP socket access. However, Starlink's CGNAT limitations prevented this from being viable long term. We transitioned to using Tailscale, a VPN mesh that supports secure IP-layer connectivity without requiring any port forwarding or public IP addresses. The NUC now connects to the base station using its assigned Tailscale address, and all messages are routed through this encrypted tunnel.

The Tailscale approach dramatically simplified our networking configuration and ensured that NUCs in the field would be able to reliably send system data to the inland base station. Firewall rules were adjusted to allow incoming traffic on specific ports used by our socket listeners.

9.1.3 Starlink Mini Satellite Link

Starlink offers the only commercially viable way to push radar data from a remote maritime module to a base station without relying on cellular coverage or line-of-sight RF relays. We chose the '*Unlimited Roam'* subscription tier at $165 per month because it imposes no hard data cap. Activation was completed using a newly created email dedicated towards starlink mini device information and handling. The disk is mounted on a flat plain, capable of easy transport during testing. All related device information, status, and credentials are available to our team via shared files.

9.1.3.1 Subscription Setup

We selected the Unlimited Roam plan for $165/month, which offered consistent performance without throttling or hard data caps. The device was activated through a dedicated Starlink project email to separate operational credentials from personal accounts. Account information, login credentials, and configuration details are all stored in the team's secure shared documentation folder.

The activation process followed standard procedures via the Starlink smartphone application. The app guided the configuration of the dish, orientation alignment, signal strength verification, and statistics tracking. Download speed, usage, and obstruction metrics were all documented during initial testing. The ability to move the Starlink Mini between test locations proved valuable, and the flat compact design simplified setup in outdoor test environments.

9.1.3.2 Physical Connection

Although wireless transmission was technically possible via Wi-Fi, we opted for a hardwired Ethernet connection from the Starlink Mini to the NUC to ensure stability and reduce interference. A 50-foot CAT6 Ethernet cable was used to connect the Starlink Ethernet Adapter directly to the NUC via USB-to-Ethernet converter. This provided consistent performance and guaranteed full connectivity.

The NUC obtained its IP from the Starlink router's DHCP, and no additional Starlink-side configuration was required. Because Starlink implements CGNAT, we confirmed that direct inbound connections were not possible. Instead, the NUC was configured to initiate outbound connections to the base station using Tailscale, which provided private mesh routing and persistent reachability despite Starlink's NAT behavior. Early communication experiments using port forwarding and dynamic DNS were unsuccessful due to CGNAT limitations, but after switching to Tailscale, the NUC could reliably send system telemetry and logs to the base station using Python-based scripts. Message acknowledgments, health checks, and error handling logic were also tested over this connection.

9.1.4 Raymarine Cyclone Radar + VCM100

The Raymarine Cyclone is the main component of the MDAS system, making it the most important piece to incorporate in the prototype. Out of the box, we were lacking needed fuses and power/ network adaptors. The radar included a 3-prong marine power cable (pigtail-style) for connection to the VCM100, a RayNet-to-RayNet network cable, and physical documentation for installation. However, the system was not fully plug-and-play. In order to adapt it for standalone use outside of Raymarine's proprietary MFD ecosystem, several additional components were sourced: a fused DC pigtail cable to power the VCM100 directly from our regulated battery bank, and a RayNet-to-Ethernet adapter for routing radar network traffic to our development machine.

The initial challenge we encountered was operating the radar independently of the ship's onboard systems. The VCM100 is not designed with an external power adapter; accordingly, we designed a custom DC input cable featuring an inline fuse so that we could provide the regulated 12V safely from our portable battery bank. During initial power-up, the VCM100 exhibited a solid yellow LED, signifying that it was in standby mode and awaiting subsequent commands. This is to be expected behavior without an MFD, as the radar waits idly until it is instructed to wake or transmit by a software-based transmit or wake command. Bypassing the MFD entirely was accomplished using Raymarine's SDK tools, which allow for injecting commands via Ethernet. This created a viable control interface, and we determined the radar could indeed be woken, shut down, and possibly sent into transmit mode via direct commands.

9.1.4.1 Power Path

The radar unit was powered through the use of a lithium power cell made by Westinghouse due to its portability and ability to deliver a stable 12VDC output. The 14-AWG cable was split into pigtails and protected by a 10-amp inline fuse before making a connection with the VCM100. When power is connected, the VCM100 engages an onboard DC-to-DC converter that boosts the voltage to approximately 42V. The high voltage is transmitted through the radar mast cable to the Cyclone radar through the 3-pin marine power connector. All electrical connections and gauge of wire were checked as per Raymarine's guidelines to ensure correct current management and fuse protection. On power-up, the VCM100 enters standby mode at once, waiting in readiness for an appropriate software command prior to starting radar operation.

9.1.4.2 Ethernet/ DHCP Connection

In order to communicate with the radar, we utilized a RayNet-to-Ethernet adapter to link the network output of the radar to a USB-Ethernet dongle on our development machine, an Intel NUC. Since the Cyclone radar does not assign itself a static IP, we needed a local DHCP server to provide network configurations on boot. We achieved this by utilizing Tftpd64, which is a lightweight DHCP server program.

We first configured the host PC with a static IP of 198.18.0.2 under Windows network settings. Within Tftpd64, we configured the IP pool to hand out addresses in the 192.18.0.x subnet, which was the expected subnet of the radar. Upon powering up the radar and allowing sufficient boot time, the radar's MAC address was present in the DHCP lease table and correctly assigned 192.18.0.3.

9.2 Hardware Testing

The hardware testing phase of the MDAS prototype was important for creating a foundation of readiness to the system’s physical components. Our testing aimed to confirm that all core hardware units, the Raymarine Cyclone radar and the Starlink Mini terminal, could operate under self-contained, off-grid conditions and respond appropriately to test configurations. These tests were not solely about functional verification but also about confirming standards and ensuring the power distribution design met voltage and current requirements. Particular attention was paid to the

conditions under which systems failed or became unstable, as these insights could directly inform our final system design for SD2. The sections that follow detail our step-by-step process for validating radar and Starlink hardware interfaces and include power-up behavior, voltage verification, connection stability, and problems encountered throughout.

9.2.1 Radar Power Integration and Troubleshooting

Hardware testing for the Raymarine Cyclone radar was one of the more important phases of our project as it deems our ability to communicate with it. Our primary goal was to ensure stable power delivery from our remote power system using Raymarine's provided electrical guidelines. Following the wiring schematic in the Cyclone installation manual, we fabricated a fused DC pigtail cable using 14 AWG leads. The cable featured a 5.5 mm DC input plug compatible with our Westinghouse lithium battery bank, and the output end terminated in pigtails spliced to match the VCM100's marine-grade power input (Figure 8.5). Initial voltage readings confirmed a clean 12 VDC output. When connected to the VCM100, the LED status immediately turned solid yellow, which we initially interpreted as a successful startup. However, further testing revealed that the VCM was outputting 0 VDC to the radar head. According to Raymarine documentation, solid yellow denotes a fault state where the radar is to self recover after 20 seconds.

Figure 9.2 *- Multimer Reading 12VDC Input and ~0VDC Output*

We initially assumed that the VCM100 required a command signal from an MFD to fully enable output. However, we discovered that the Westinghouse battery would shut off automatically due to low current draw. To resolve this, we connected a second load, a phone charger, to another output port. This ensured the battery remained active long enough for the VCM to engage properly. Approximately 35 to 40 seconds after startup, the VCM transitioned to a blinking yellow state, indicating entry into sleep mode. Once the radar's physical power switch was toggled, the VCM100 shifted to a solid green state, confirming successful activation. A final voltage check verified 42 VDC at the VCM output, affirming correct high-voltage transmission to the radar head and full hardware readiness.

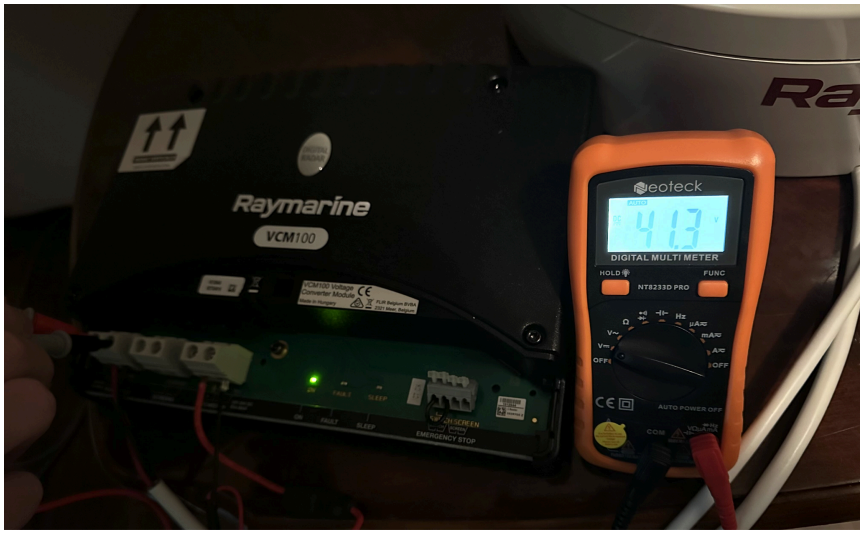


Figure 9.3 *- Multimer Reading ~42VDC with VCM100 Green 'ON' LED*

9.2.2 Starlink Connectivity Testing

Initial Starlink hardware setup followed the onboarding steps within the Starlink smartphone app. After account creation and subscription to the $165/month "Unlimited Roam" plan, the Starlink Mini was activated using a dedicated team email. The setup process involved aligning the dish, applying firmware updates, and verifying connectivity statistics such as download speed and signal obstruction.

Figure 9.4 *- Starlink Mini Terminal*

Though functional, Wi-Fi performance was initially unreliablewith only 5 to 10 Mbps and frequent dropouts. We suspected signal obstruction due to our limited testing location outside on a patio. To address this, we repositioned the dish to a more open area and utilized a 50-foot CAT6 cable to connect the Starlink Ethernet Adapter directly to the

Intel NUC using a USB-to-Ethernet converter. With the hardwired connection and a less obstructed view of the sky, speeds improved to around 50 Mbps and stability was significantly better. This setup was retained for all future testing where the Starlink was needed to ensure reliable connectivity between the NUC and our base station.

9.3 Software Testing

The software testing phase encompassed all networking scripts, radar control logic, and messaging routines between the NUC and the HP base station. These tests served both to validate protocol compatibility and confirm command-based radar control. Further software testing included the prototype base station's web application interface.

9.3.1 Radar Command Interface and SDK Testing

Once the radar was physically powered and the VCM100 reached a stable green state, our next objective was to establish communication via Raymarine's encrypted SDK tools. After securing the SDK encryption key, we requested the proprietary files from Raymarine. This included C++ source files, compiled command line tools, and configuration examples necessary for command-based radar control. The files were extracted to C:\RaymarineSDK and from following initial guidelines, a DHCP was required to communicate with the radar. We installed TFTPD64 to serve as a DHCP server. This was essential because Cyclone radar does not have a static IP and requires a lease from a DHCP host. We configured the NUC to have a static IP address of 198.18.0.2 (a reserved non-routable IP range often used for radar testing). Subnet mask was set to 255.255.240.0, matching Raymarine's documentation. No gateway or DNS server was configured since no external network was required. With the DHCP server active, we powered the radar and monitored the lease table. Within 30 to 60 seconds, an entry with the radar's MAC address appeared with an assigned IP of 198.18.0.3. This confirmed that the DHCP server was functioning, and the radar was requesting a lease, or requesting an IP.

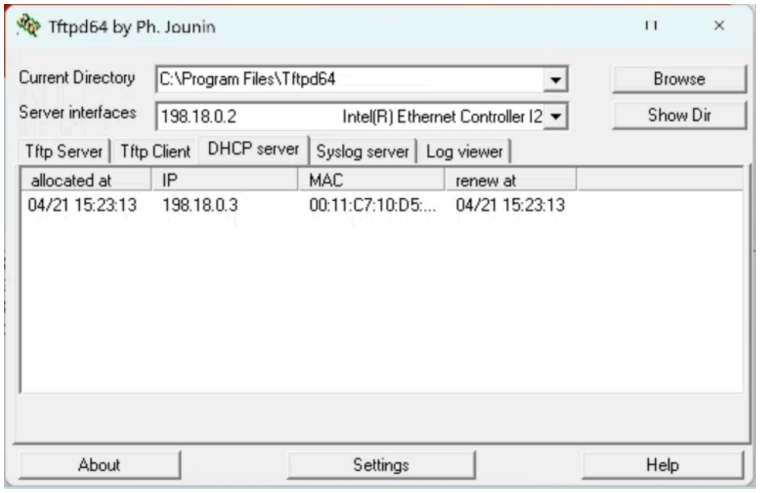


Figure 9.5 *- DHCP Server With Radar Populated*

We next launched the CLI tool supplied by the SDK to test for radar responsiveness. However, the initial run of the SDK's CLI tool returned no visible response from the radar. Our next step was based on the idea that a runtime environment issue was at fault, so we rebooted the NUC and power-cycled the radar. After this, the CLI successfully output logs indicating radar module initialization. We used WireShark to validate Ethernet packet activity. The interface captured steady UDP and TCP traffic between 198.18.0.2 (NUC) and 198.18.0.3 (radar), confirming bidirectional communication. This confirmed that our control layer was operating.

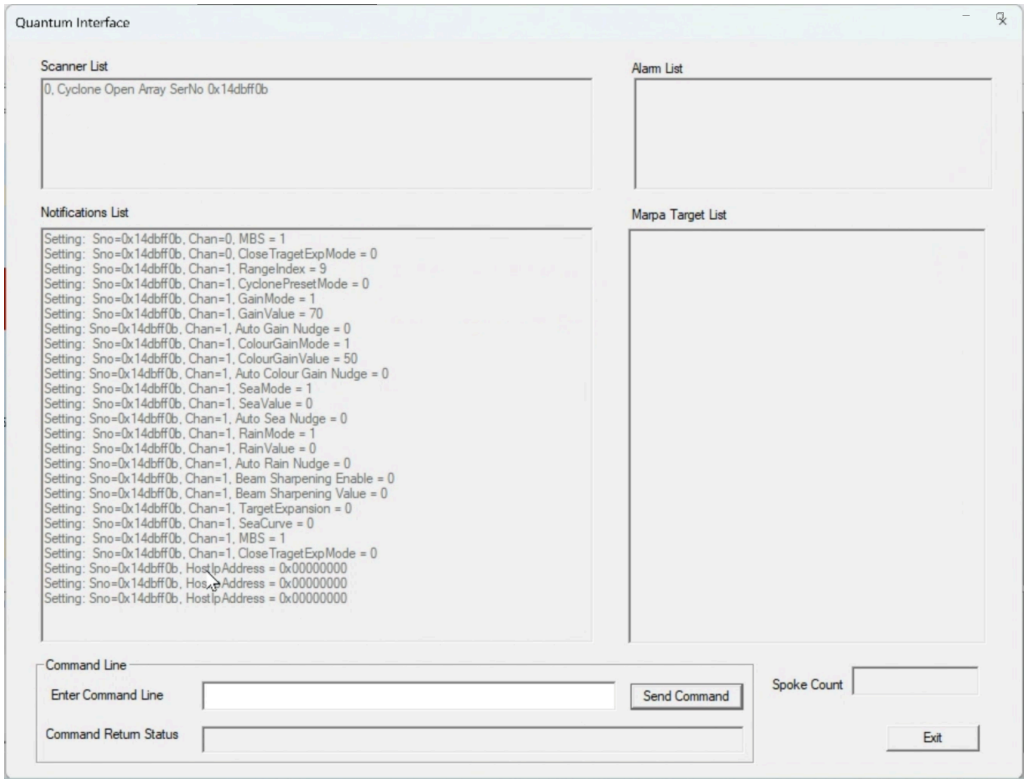


Figure 9.6 *- Quantum Interface, CLI from Raymarine SDK With Radar's Instantiation*

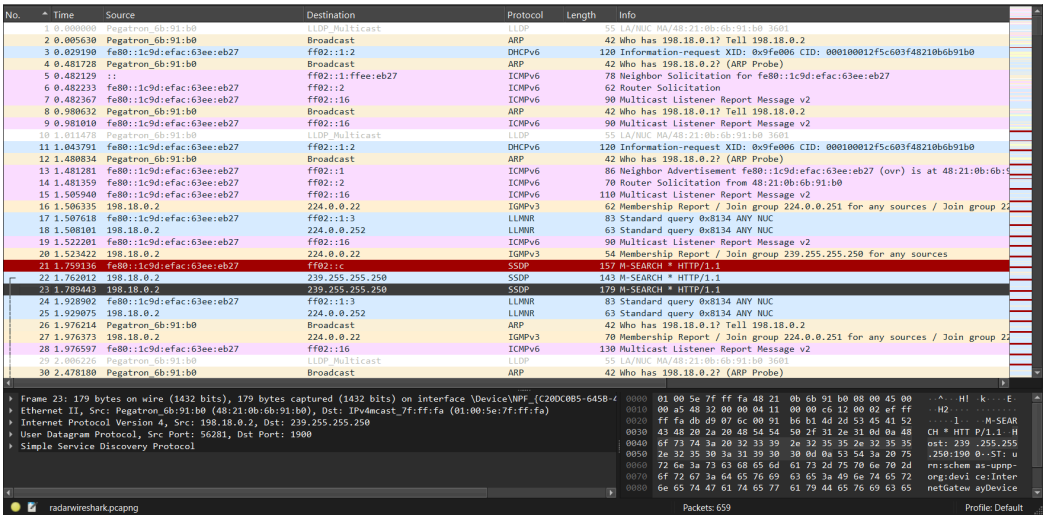


Figure 9.7 *- Wireshark Packets When Connected to Radar*

We then tested control commands:

- 0,1,3; instructs scanner index 0 to use setting index 1 (eRadarMode) with value 3 (PowerDown). This command resulted in the VCM light transitioning from solid green to blinking yellow, and radar packets ceased in WireShark within 10 seconds.

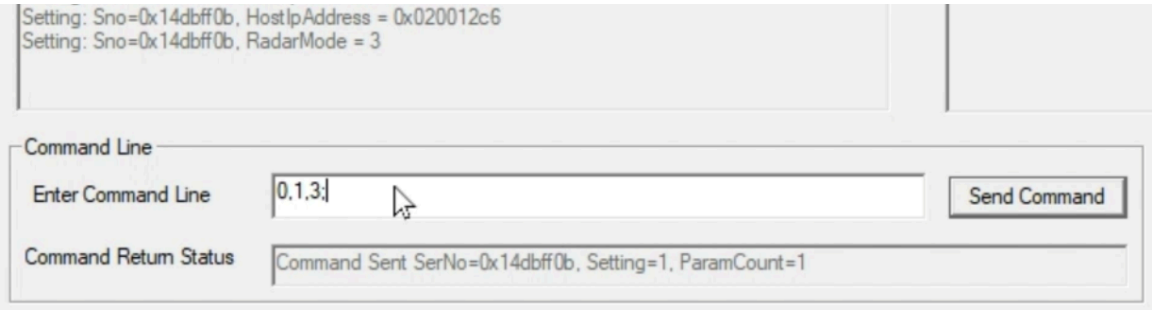


Figure 9.8 *- ePowerDown Command Sent to Radar Through Raymarine Quantum Interface*

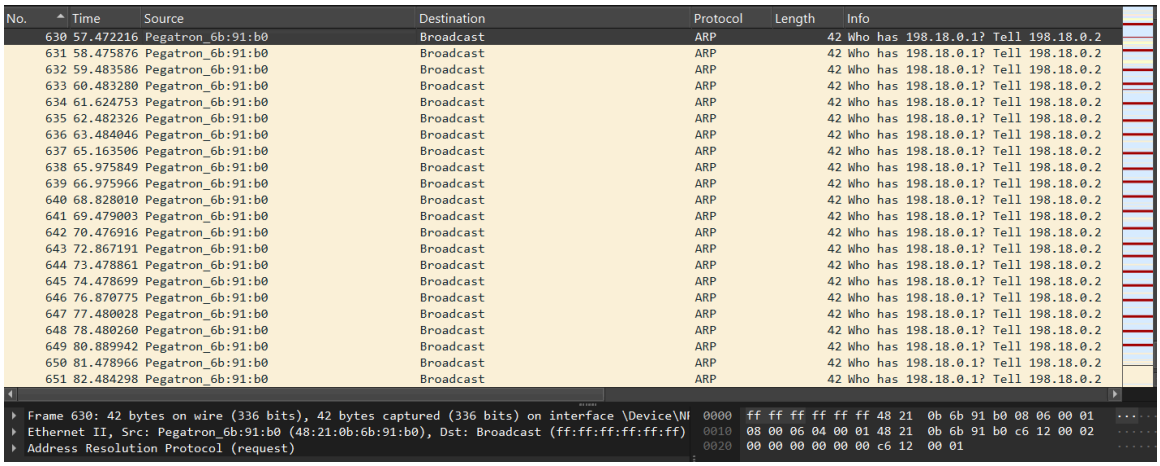


Figure 9.9 *- Wireshark Transmission After Radar Was Sent Power Down Command*

- 0,117; was the next test which is the eWakeUp function. Sent to scanner 0, this reinitializes radar transmission. Around 20–30 seconds later, packet traffic began to trickle in and the VCM indicator returned to green.

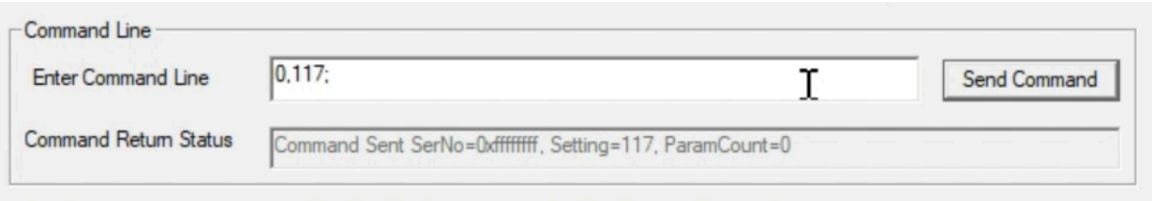


Figure 9.10 *- eWakeUp Command Sent to Radar Through Raymarine Quantum Interface*

Figure 9.11 *- Wireshark Transmission After Radar Was Sent to Wake Up*

Each of these confirmed data receipt and radar health checks. This entire sequence validated that the SDK could not only initialize radar but also support ongoing telemetry and command control through direct Ethernet and CLI interface. With a stable IP assignment, consistent packet exchange, and verifiable LED feedback corresponding to our issued commands, we gained critical assurance that the radar could be remotely operated and monitored with no dependency on a proprietary MFD. Beyond this, we have yet to truly turn the radar to transmit mode and collect concrete data, but the SDK's ability to communicate with the radar has been put to the test.

9.3.2 Starlink Communication and Data Exchange

Our next task was to validate remote communication from the radar node (NUC) to the base station (HP laptop) over the Starlink network. We began testing using basic TCP socket code written in Python. Initially, when both NUC and base station were connected to the same local Wi-Fi, our test server-client model succeeded. The server (base station) opened a socket listener on port 5000 and printed messages received. This script was named server.py and simply listened and printed any/ all messages received. The NUC client script, client.py, sent messages "Ping from NUC" which were printed on the server's terminal and saved to a text log. This was done with the client python script hard coding the IP of the base station computer, thus knowing exactly where to ping.

However, when we moved the NUC onto the Starlink network (Ethernet port of the router), communication failed. After researching, we discovered that Starlink implements GGNAT, which prevents inbound connections due to lack of public IP assignment. Even with DDNS and port forwarding attempts, all tests failed. We implemented WireGuard and other VPNs but encountered similar NAT traversal limitations. Our final approach was to install Tailscale, a mesh-based VPN that uses WireGuard underneath but supports NAT punching without any need for port forwarding or static IPs. After account creation and device authorization, both the NUC and HP laptop were added to the same Tailscale tailnet. This provided each device with a stable IP address in the 100.x.x.x range, reachable from either endpoint.

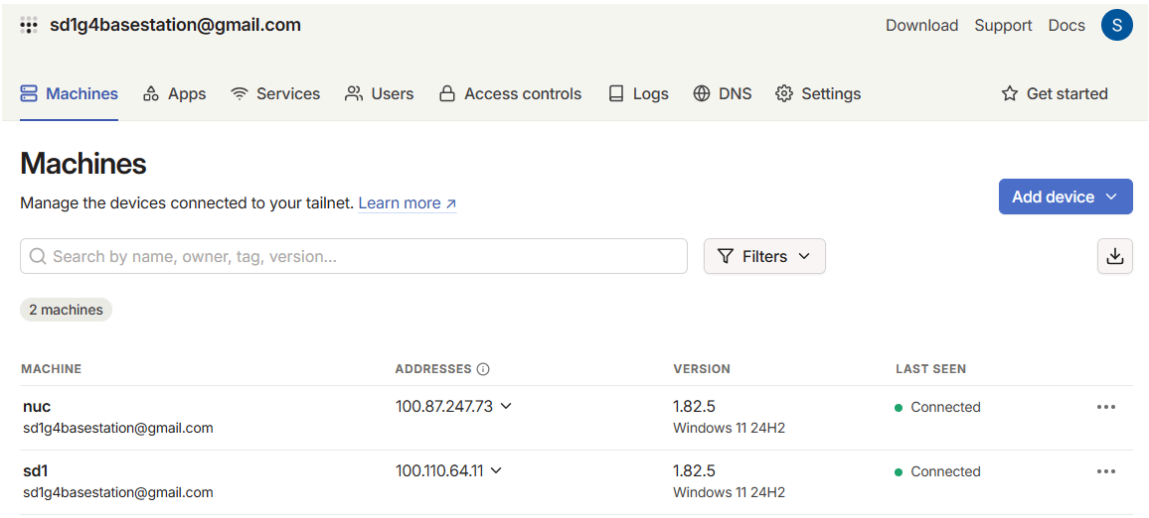


Figure 9.12 *- Tailscale Interface With NUC and HP Base Station Connected*

We modified our socket scripts to use the Tailscale IP addresses. This resulted in successful message exchange, even when devices were on different networks. This was a great success in that the NUC was now connected to Starlink, and the base station was connected to a generic wifi. Communication between the two was fluid. We expanded the communication logic:

- The client script on the NUC now sends system info relating to a timestamp, CPU usage, and system uptime.
- The base station ran a GUI app with a text log keeping track of all messages, and a shutdown button that closed the socket on command (Appendix D - [4]).
- The client also received an acknowledgement statement from the base station. (Appendix D - [5])
- Base station printed out connected and disconnected stamps.
- The client python script correctly timed out after 5 seconds when connection is not successful or if the server is done (Appendix D - [6]).

Each incoming message was parsed, logged, and appended to a text file with timestamps This architecture confirmed real-world data delivery over a commercial satellite link and secure mesh routing. The entire pipeline, NUC to base station, was now functionally complete. These tests also proved that despite the inherent network limitations of Starlink's CGNAT, we could enable bi-directional message flow using a lightweight overlay network without introducing latency or instability. With live data, log persistence, and remote shutdown controls functioning as intended, our software and networking solution now demonstrates reliable performance that is ready to be tested to its limits regarding transmission rate and server reliability. These are important factors when the modules will be overseas.

9.3.3 Web Application Testing

Testing is a critical part of the software development lifecycle, ensuring system reliability, data integrity, and performance under various conditions. In MDAS, the testing strategy includes a combination of manual and automated approaches across the frontend, backend, and cloud infrastructure components.

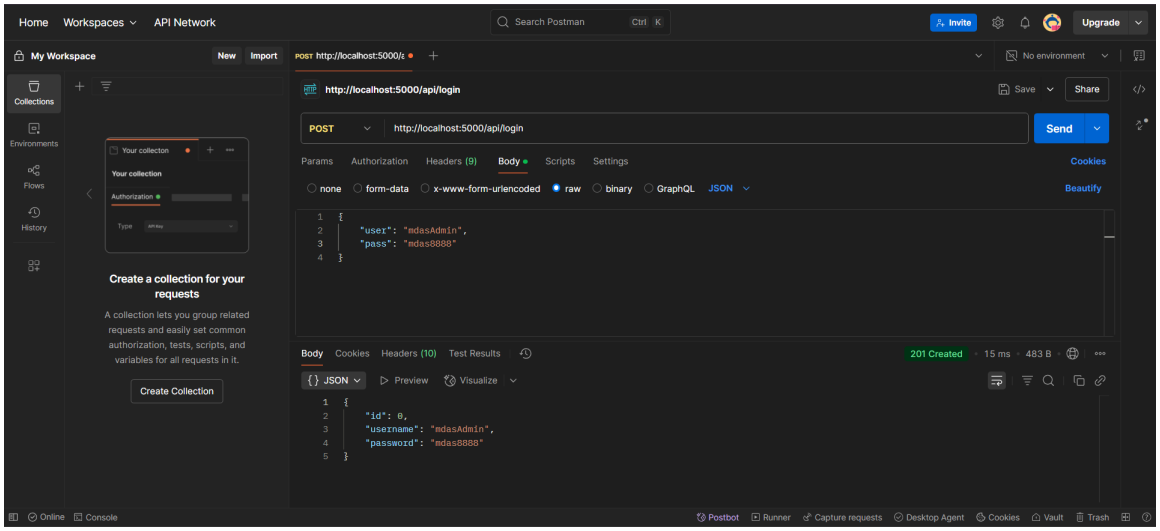


Figure 9.13 *- Postman API Login Test*

Initial testing efforts focused on validating the login functionality through a local Express.js backend prototype. The team used Postman, a widely adopted API testing tool, to send HTTP requests and verify responses for endpoints such as user authentication and role-based access. This allowed early-stage validation of login logic, session handling, and expected API behavior before migrating the backend logic to serverless Lambda functions.

As development continues, Postman will remain a primary tool for manually testing API Gateway endpoints. These tests will verify expected behaviors for endpoints that receive data from the MCU, authenticate users via Cognito, or retrieve vessel-related records. Each route will be tested with various payloads to ensure error handling, input validation, and response consistency.

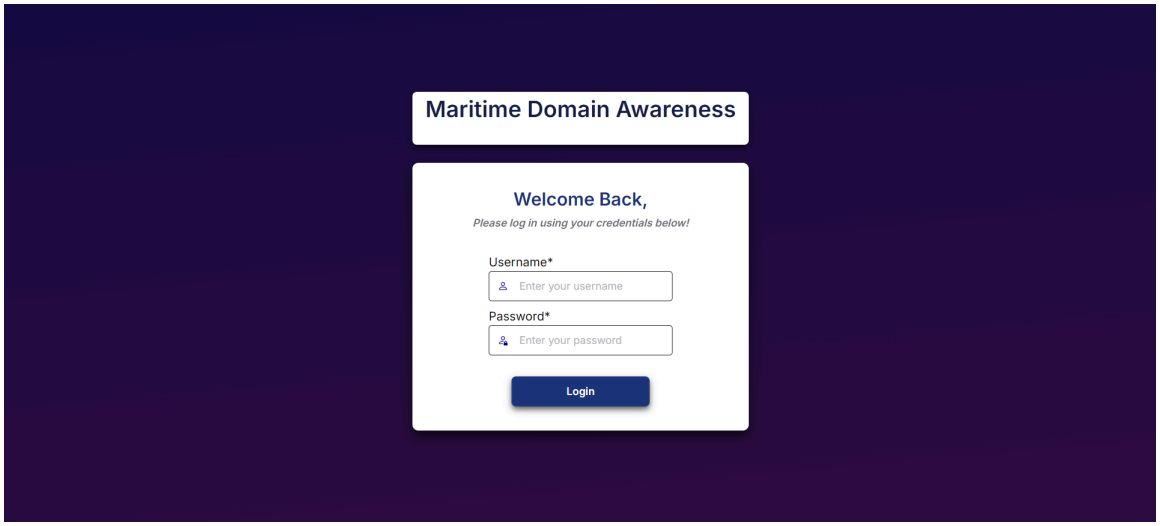


Figure 9.14 *- Login Page Prototype*

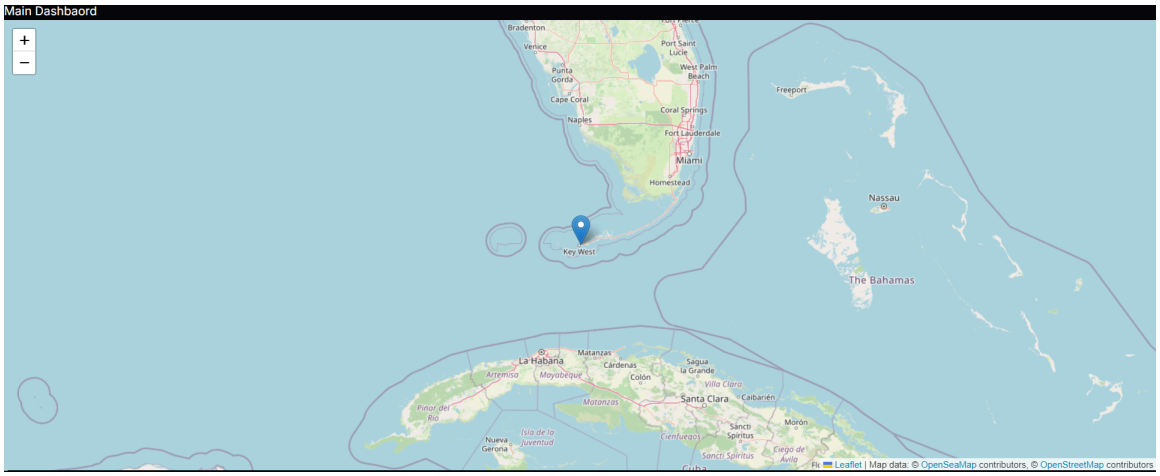


Figure 9.15 *- Dashboard Map Prototype*

For the frontend, manual testing will be conducted to validate user interactions with the React interface, including form submission, authentication flows, dashboard navigation, and the handling of real-time updates. Frontend unit tests will be implemented where appropriate using tools such as Jest or React Testing Library, particularly for validating components like login forms, toggles, and alert displays.

Backend testing for AWS Lambda functions will include validating function triggers, payload parsing, and expected outputs. Where feasible, unit tests may be implemented using frameworks like Mocha or Jest, paired with local emulation tools like AWS SAM CLI or Serverless Framework to simulate Lambda behavior. The correctness of data passed to and retrieved from the database will be a focus during integration testing.

Later in the development cycle, the team may explore automated testing and continuous integration through GitHub Actions, which will help enforce code quality and reduce regressions. Automated workflows could include basic linting, unit tests, and deployment pipeline steps.

9.4 Optical Feasibility and Testing

In order to see if the chosen imaging system will perform well, we must put the system through multiple rounds of testing. This testing will be done both physically as well as simulated, specifically in Zemax. For the physical testing, a prototype was assembled using lenses acquired from the undergraduate lab, as described in Chapter 8.3. The goal of this was to justify if the system could work, and showcase the fundamentals of the project. For the simulation, Zemax was used to design the lens system and optimize the values, as well as showcase potential flaws in early designs. Zemax can simulate many different image quality outcomes, including the spot size, aberrations, and point spread function (PSF) of the system. By using these options, we can get a better idea of how the system will perform, and make adjustments accordingly. The physical tests were done in the CREOL senior design lab using materials from the undergraduate lab, therefore they are not the components that will be used in the final project. These tests were done to get proof of concept and demonstrate basic working principles.

9.4.1 Lens System

For the Galilean system, the two lenses are composed of a positive focal length (biconvex) and a negative focal length (biconcave). These two must be precisely spaced and aligned so that the light travels through both of them to hit the sensor properly. For our specifications, we are looking for between 5 - 7.5x magnification. This means the ratio of the focal lengths of the lenses must equal 5 - 7.5. The basic lens system that was tested was a Galilean system with a biconvex objective lens of 150 mm focal length, with a biconcave eyepiece that had a 25 mm focal length. The system also included a relay lens to increase field of view. As mentioned in Chapter 8.5 there were two versions of this system tested in the lab. One version had a 25 mm biconvex lens placed in front of the sensor, while the second version had a 35 mm plano-convex lens in front of the sensor. These two setups both produced images at the target distance of one meter, however the version with the biconvex lens provided a clearer image, and so testing moved forward with this design in mind.

With this specific system in mind, the field of views were calculated to get a better idea of the image that would be produced. The calculations are as follows:

- HFOV = $2 \cdot tan^{-1}(\frac{4.968}{2 \cdot 150}) = 1.987°$
- VFOV = $2 \cdot tan^{-1}(\frac{3.726}{2 \cdot 150}) = 1.423°$
- DFOV = $2 \cdot tan^{-1}(\frac{6.2}{2 \cdot 150}) = 2.367°$

Since the CS165CU has a 1 / 2.9 optical format, the pixels are (4968 um x 3726 um), with the diagonal being 6.2 mm. These field of view values are incredibly narrow, and would be good for producing an image at a farther distance, around 7 km. To keep the testing confined to the lab, the relay lens was added to get an image in a wider field of view. The angular resolution of the system was calculated to be

$\alpha = \frac{1.22 \cdot (550 \times 10^{-9})}{50.8\,mm} = 1.32 \times 10^{-5}\,rads = .000757°$. This value is still incredibly narrow.

During testing of the system, although the combined three lenses produced an image, it was very blurry, and it was clear there were a multitude of aberrations present. An iris was added in front of the system to try to minimize field curvature, which resulted in the top image below. This image is visually very blurry, with the edges being slightly distorted. This can be due to lens spacing not being precise enough, spherical aberrations, the system being misaligned, or stray light entering the system. With this in mind, multiple interactions and spacings of the system were executed, with the image below being the best result. Upon further investigation, it was also discovered that the objective lens did not have a 150 mm focal length. The focal length was more accurately 100 mm, with the misunderstanding coming from not properly labeled equipment and not double checking the focal length before system assembly. Therefore, the target magnification was not met, as this only provides a 4x magnification. This also meant the calculated field of view was incorrect as well, with the correct values being:

- HFOV = $2 \cdot tan^{-1}(\frac{4.968}{2 \cdot 100}) = 2.846°$
- VFOV = $2 \cdot tan^{-1}(\frac{3.726}{2 \cdot 100}) = 2.135°$
- DFOV = $2 \cdot tan^{-1}(\frac{6.2}{2 \cdot 100}) = 3.551°$

These values are still narrow, but not as much as the previous calculations.

As this setup could not present accurate data in the testing scenario, the system was reduced to just one lens, the 150 mm biconvex objective lens, and the 3D printed lens tube to minimize light entering the system. This configuration produced a much clearer bottom image, with the target distance remaining one meter. This image was used for the angular resolution calculations in MATLAB.

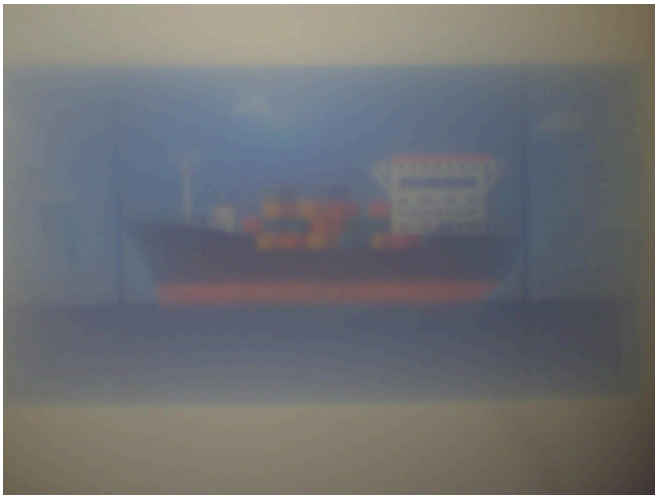
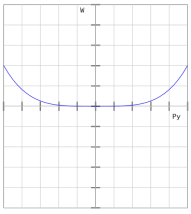
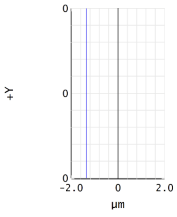
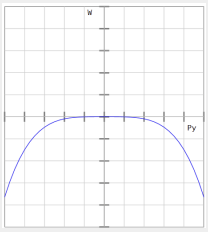
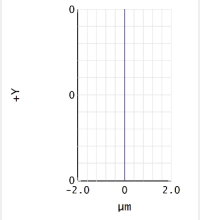


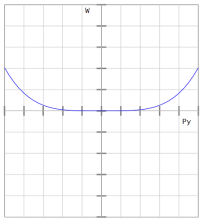
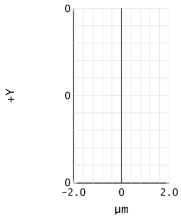
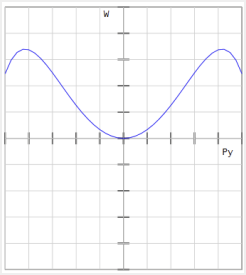
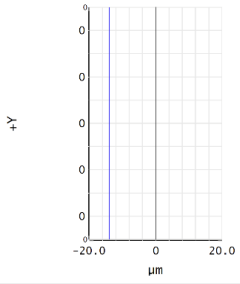
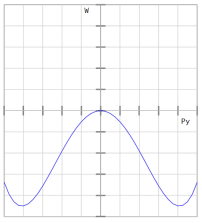
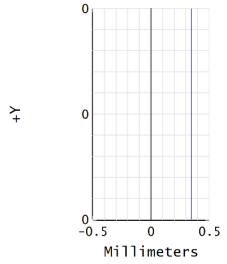
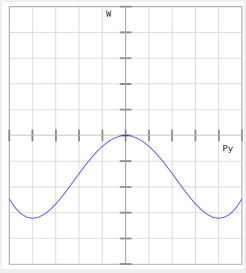
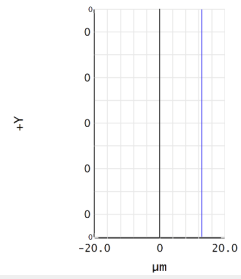
Figure 9.16 *- Original Image Test*

Figure 9.17 *- Final Image Test*

Zemax helped determine the combination of lenses that would be best fit for this project. The goal of the system is 3-5x magnification, without ending up with a super bulky imaging system. Something to keep in mind is needing to look at the spot diagram to see how the image will focus on the CMOS sensor. The spot size diameter should fit within a pixel of the sensor, so since the Allied Vision sensor we are using has a pixel size of 4.8 microns, we are looking for a spot size of around 4.8 microns. The spot diagram also needs to be symmetrical. There were multiple renditions of a lens system modeled in Zemax and they all yielded different results, which have been summarized in the table below.

Table 9.1 - *Zemax Lens Testing*

Lens Combination	**RMS Spot Radius**	**Total Axial Length**	**Wavefront Error**	**Field Curvature**
500 Biconvex -100 Biconcave (⌀2”)	1526 um	3123.38547 mm		Off-axis at -1.77 um 
750 Plano-convex -150 Plano-concave (⌀1”)	9.112 um	505.708 mm		On-axis 

Lens Combination	RMS Spot Radius	Total Axial Length	Wavefront Error	Field Curvature
400 Plano-convex -100 Plano-concave (⌀1”)	8.763 um	401.329 mm		On-axis 
250 achromat doublet -75 bi-concave (⌀1”)	0.331 um	255.32072 mm		Off-axis at -13.9 um 
400 Plano-convex -100 Plano-concave 40 achromat doublet (⌀1”)	3.280 um	414.329 mm		Off-axis at 0.355 um 
300 achromat doublet -75 biconcave (⌀1”)	0.138 um	305.51165 mm		Off-axis at 13.7 um 

After running a few options through Zemax, the best option for the proposed system involves an achromatic doublet objective lens and a bi-concave eyepiece lens. Although the magnification is only 4x, the spot size is incredibly small, and there is minimal wavefront error. This imaging system also is just barely longer than one foot, which while still long, is one of the smallest of the proposed tests. These tests help us get a better idea of how the system will perform once we get the physical components.

9.4.2 Angular Resolution

For testing the CMOS camera sensor, we looked at the angular resolution of the system. This was tested considering both the sensor and the lens system. In the lab testing, we used a Thorlabs CS165CU CMOS Sensor for the angular resolution calculations. The calculations for field of view and angular resolution were done by hand, then a MATLAB code took the image and additionally calculated the angular and pixel separation. Using the specifications of the CS165CU sensor (Appendix C, [2]), the following calculations were completed:

- $\theta_{pixel} = \frac{pixel\ size}{f_{obj}} = \frac{3.45\ \mu m}{100\ mm} = 34.5\ \mu rad = 0.0020°$ (sensor-limited)
- $\alpha = \frac{1.22 \cdot (550 \times 10^{-9})}{50.8\ mm} = 1.32 \times 10^{-5}\ rads = .000757°$ (diffraction-limited)

For further testing of the angular resolution, we looked to calculate angular separation of the target image. To do this, lines were drawn on the image target to create a stark contrast level, and the distance in between these lines was measured. The lines can be shown above in Figure X. The distance between these two lines was measured as 59.5 mm, which means the angular separation at one meter is 0.0595 radians, or 3.41 degrees. This value is well above the angular resolution calculated for the sensor, so the image is resolvable. To get a more accurate calculation of this from the image itself, we wrote a MATLAB script to take the image and create a contrast graph, which would then give pixel spacing and therefore angular separation (Appendix D, [3]). The code reads the intensity profile of the image, and uses the two peak values to measure the pixel separation. It then produces an evaluation of the target spacing, pixel separation, the related distance in mm, and the measured and expected angular resolutions. It also gives the sensor-limited resolution value. The intensity profile associated with the image, as well as the calculated results, is shown below:

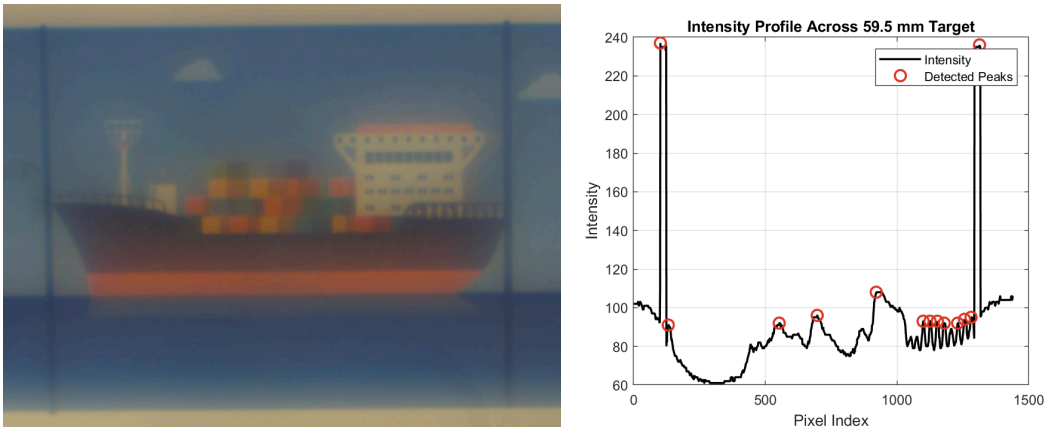


Figure 9.18 *- Test Image (left) vs. Intensity Plot (right)*

- Target spacing: 59.50 mm
- Measured pixel separation: 30 px
- Measured separation: 0.10 mm
- Measured angular separation: 0.006°
- Expected angular separation: 3.409°
- Sensor-limited resolution (2 px): 0.0000026°

9.4.3 Senior Design 2 Testing

After senior design 1, further testing was done as the final components got delivered and we were able to properly assemble the final prototype setup. While the previous sections highlighted a two lens system being best for the project, further Zemax simulations and physical tests determined that a single achromatic doublet lens would produce the best image. This is due to the two lens system not focusing onto the flat camera sensor properly. Additionally, the 3D printed lens tubes were letting too much scattered light into the system, causing aberrations (in addition to the wrong lens setup). For the finalized prototype, we used Thorlabs standard lens tubes with a total length of 303 mm which accurately represents what was simulated in Zemax.

Below is the first test picture we got with the Alvium CMOS sensor, with little calibration:

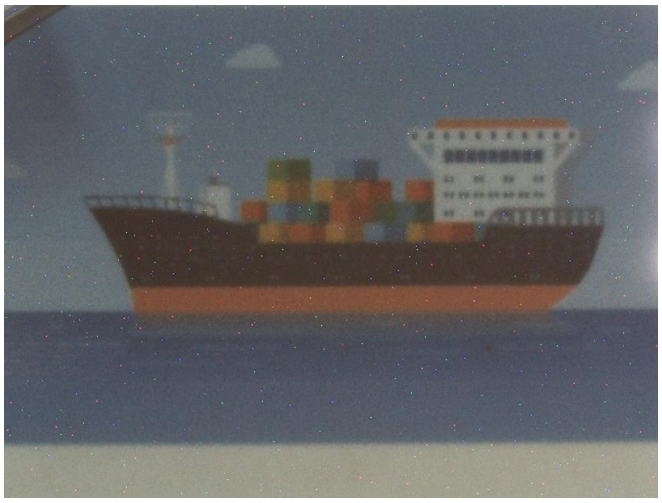


Figure 9.19 *- Test Image at 10 Feet*

This image is already much clearer than previous tests, and performed at a much longer distance. There was a new challenge introduced with the much higher resolution camera sensor, and that is defective pixels. All over the image there are little RGB specs which each represent a defective pixel on the sensor. While certainly noticeable, they do not compromise the overall integrity of the image. There are a few methods to fix this issue: Allied Vision's defective pixel mapping software, a dead pixel replacement algorithm, or trying to fix each pixel through a python or MATLAB script. These came with their own

challenges however, and due to time and budget constraints was unable to be fully resolved, however operating conditions were optimized to get the least amount of defective pixels per image. One of the changes made to optimize the images was switching the targets from 2D printed images to actual 3D models. The poor quality of the printed images made all the testing photos look lower quality than they actually were, which contributed to the low system performance. After the switch was made to the 3D models of different boats, the images looked much clearer, and were more detailed. Here is one of the final optimized images:

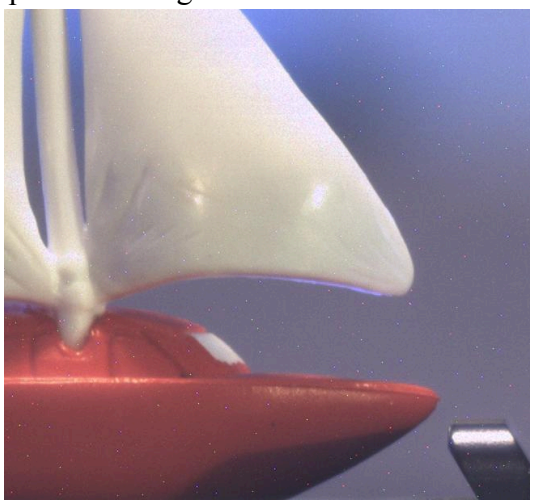


Figure 9.20 *- Final Test Image*

There were still a few issues that affected the live-video feed on the camera's SDK (VimbaX). One of these issues was the gimbal's auto adjustments. Since the gimbal was not made specifically for this system, the movements of the three-axes were not as precise as needed for the small field of view. Therefore, on the SDK the image is constantly adjusting, going from blurry to focused every few seconds. Another issue was the 10 second lag on the SDK. This is partially due to a couple underlying factors. The first is that a high processing sensor such as the Alvium normally requires computers with heavy processing capabilities, as well as external storage to run efficiently as possible. Due to budget constraints, neither of these were able to be acquired. The second issue was the length of the cord used to connect the sensor to the laptop. It was a 3 foot long cord, which meant decreased bandwidth and increased lag on the SDK.

Additionally, we switched the gimbal to the SCORP-C model described previously, which is a handheld model that demonstrated the working principles of the gimbal integration with the imaging system. There was a bit of trial and error involved with setting the system up properly, the first error being trying to balance the camera without a proper counterweight. The servos couldn't support the weight of the sensor and the lens system (~100 g) and would turn off immediately after trying to auto-balance. After the counterweight was installed, this was no longer an issue and the gimbal worked perfectly.

9.5 Overall Integration

At the conclusion of Senior Design 1, the MDAS system exists as a series of well-functioning but loosely coupled modules. Each subsystem, radar integration, Starlink transmission, base station web app, and optical demonstration, have been verified independently through functional testing. However, a true fully integrated trial where the radar acquires data, the NUC processes it which is then transmitted via starlink to the base station to be displayed results has not yet been conducted.

This current phase has prioritized individual validation and reproducibility of each component, laying a strong technical foundation for complete integration. Our team has demonstrated command-based radar control via SDK, established secure network transmission from remote nodes via Tailscale over Starlink, and implemented structured message logging and telemetry parsing at the server side. The base station's prototype interface currently accepts login/ account creation criteria with a rudimentary dashboard view. There are also concrete optical foundations for a lens system to extend range. What remains is the integration of these functions into a seamless loop. The groundwork is complete. The radar can be powered, identified on a local network, and issued commands remotely. The NUC can package system diagnostics and relay them. The base station can respond, display, and record activity. Despite this progress, the modules remain independent. The radar does not currently output useful scan data in any digestible format. There are no processing scripts in place on the NUC to translate radar responses into structured packets, and the base station GUI is not prepared to ingest or visualize such information.

9.6 Plan for SD2

Senior Design 2 will focus on full vertical integration and real-world field testing. The first major task will be converting the NUC into a data-processing hub, capable of translating radar responses into potentially JSON-structured messages. This will require a deeper study of the SDK's scan mode and packet structure. Additional CLI commands may need to be reverse-engineered or trialed iteratively to achieve meaningful output. Once data can be interpreted, it will be transmitted using the already proven Tailscale-Starlink setup. The base station will require a complete overhaul of its backend to enable automated packet listening, parsing, and storage. The current prototype GUI will be upgraded to include a real-time dashboard that displays scan status, timestamps, and key performance indicators.

A second major SD2 priority will be the development of an AIS subtraction algorithm to distinguish passive radar returns from known ship transponder data. This will involve pairing radar detections with AIS signals and filtering them out to highlight silent, unregistered targets. A third effort, though closer to advanced goals, entails complete testing of a full up system where we can track uptime, latency, packet loss, and fault triggers via a smartphone application and on the base station. This includes verifying system recovery from power interruptions and ensuring sustained performance under real-world bandwidth constraints. Finally optics integration will continue. The current

lens system will be tested alongside the radar module to determine alignment feasibility. This also must be integrated via a controlled gimbal for swivel capability If data from both sources can be co-located, the groundwork will be laid for sensor fusion.

In total, SD2 will move the MDAS project from a prototype composed of independent parts to a fully integrated, partially autonomous surveillance system with a field-ready architecture.

Chapter 10: Administrative Content

10.1 Budget

The tentative budget allocated for the development and deployment of the Maritime Domain Awareness System (MDAS) prototype is $15,000, pending final approval and possible revision based on sponsor feedback and field test results. This budget ceiling reflects both the innovative scope of the system and the need for cost-effective scalability in potential future deployments.

The majority of the budget has been allocated to the core components, those being the radar unit and the Starlink module. The remaining budget has been primarily allocated to the creation of the power system to run the core components using renewable resources. This BOM also includes most foreseeable expenses used during the actual construction of SD2, not just what is used for prototyping.

10.2 Bill of Materials

Table 10.1 *- Bill of Materials*

Bill of Materials (BOM)				
Component	**Model / Description**	**Power Requirements**	**Additional Notes**	**Approximate Price**
Raspberry Pi	Raspberry Pi 4 Model B (8GB RAM)	5V DC, 3A	Requires waterproof, vibration-resistant enclosure and passive cooling (heat sinks, thermal pads)	$80.00
NUC PC	Intel NUC (Core i7)	19V DC, 30-50W	Replaces Raspberry Pi during prototyping. Used purely for integration and test.	$750.00
Laptop PC	HP 17 (Core i7)	19.5V DC, 20-60W	Functioned as the Base Station for inland communications. Ran off AC wall power during testing.	$600.00
Smartphone	Samsung S24 FE	5V, 10W	Eventually will be used in place of the starlink terminal. Currently will be used as a remote control application.	$800.00
Starlink	Starlink Mini Terminal	48V DC, 50-100W	Needs unobstructed mounting for optimal signal reception, a weather-resistant design	$450.00

Starlink	Starlink Mini Subscription	N/A	‘Unlimited Roam’ Plan	$165/month (2 months)
Cyclone Radar	Raymarine Cyclone 6-Foot Model with Development Kit	28V DC, 100-120W	Requires corrosion-resistant housing and must be mounted to minimize interference	$8,250
SDK	Raymarine Software Development Key	N/A	Allows the software communication with the cyclone series radars.	N/A
Wind Turbine	Pikasola 400W Wind Turbine Generator	12V DC	Compact and lightweight, suitable for moderate wind conditions	$269.99
Battery	Ampere Time 12V 200Ah LiFePO4	12V DC	Long lifespan, maintenance-free, reliable for continuous operation	$539.99
Portable Battery	Westinghouse 100 Watt Solar Power Portable Station	Depends on draw, takes AC wall input.	Replaces the battery during prototyping. Lightweight and fully functional	$129.00
Charge Controller	Victron SmartSolar MPPT 100/30	12V/24V DC	Maximizes solar energy harvesting, Bluetooth-enabled for monitoring	$245
Power Distribution Unit (PDU)	Mean Well RSD-300 Series	Multi-output: 5V, 28V, 48V DC	Industrial-grade, high-efficiency, overcurrent, and thermal protection	$135.85
DC-DC Converter	DROK 48V to 5V Step-Down Converter	5V DC Output	Powers Raspberry Pi efficiently from Starlink power source	$19.99
Radar Networking Adaptor	Raymarine A80513 RayNet Male to RJ45 Male Adaptor Cable	N/A	Connecting marine electronics devices (Raymarine Radar) via Ethernet	$25
Mounting Hardware	Stainless Steel Brackets and Fasteners	N/A	Corrosion-resistant, ensures secure attachment of components	Varies

Enclosure	IP67 Waterproof Electronics Box	N/A	Protects components from water and environmental damage	$18.88
Coatings	Polyurethane Epoxy Vinyl-ester	N/A	Barrier against moisture, oxygen and salts UV resistance, abrasion and impact resistant Prevents marine growth on structural components	All varies based on amounts used
Electrical Connectors	Raynet to Ethernet, Ethernet to USB, DC to Pigtails, Fuses, Misc electrical components	N/A	General adaptors for networking and power. Including fuses and misc electrical pieces.	~$60.00
Camera	Allied Vision Alvium 1800 U-050c, 1/3.6" 0.5MP C-Mount, USB 3.1 Color Camera	12V DC	Protection from environmental damage, high resolution	$359.02
Gimbal	SCORP-C Camera Stabilizer 3-Axis Mirrorless Gimbal	12V DC	Non-corrosive material, vertical and horizontal manipulation	$249.00
Achromatic Doublet Lens	AC254-300-A Achromatic Doublet	N/A	Decreases overall spot size for precise imaging	$89.10
Mounting Plate	AV Alvium M6 and ¼-20 Mounting Plate	N/A	Enables tripod/gimbal mounting ability for the CMOS sensor	$20.52
Total:				**$12,333.35**

10.3 Distribution of Workload

The table below shows the distribution of tasks each group member has agreed to be responsible for. While everyone has their own tasks, it is important that we all collaborate and support each other. Everyone is responsible for the success of not only their components but making sure every system can integrate together and work seamlessly in the final product. ECE students should strive to support the development of the optical system and CREOL students can provide assistance in implementing the electrical or software designs. To maintain accountability, weekly meetings are held either in person or digitally and each member is expected to provide updates on their progress. These check-ins help the group address any issues and make adjustments accordingly.

***Table 10.2** - Workload distribution*

Member Name	Discipline	Responsibilities
Alexander Drew	Electrical Engineering	Development of the power system that will power the main components of the device.
		Power Distribution
		Aid with MCU programming
Derek Florer	Computer Engineering	Project Lead
		Radar and Starlink Integration
		Embedded and Mobile App Secondary
Jet Chen	Computer Engineering	GPS implementation using Starlink
		Design weatherproof enclosure and mounting hardware
		Smartphone Application Lead
Marco Plasencia	Computer Engineering	Embedded Systems
		Software Integration & System Communication
		Documentation
Gianna Montevago	Photonics	Camera system
		Camera enhancement details
		Gimbal for cameras
Garrett Miller	Computer Science	AIS Algorithm Lead
		Database Manager
Devin Parham	Computer Science	Web Application
		AIS Algorithm Implementation

10.4 Project Milestones

The table below outlines the key project milestones our senior design group hopes to accomplish. They consist of task details, timelines, deadlines and each member's responsibility. It is very important to follow these timelines and ensure everything is finished on time.

Table 10.3 *- Project milestone plan*

Senior Design I Project Milestones: Spring 2025			
Task	**Time Period**	**Due Dates**	**Who's Responsible?**
Develop Project Idea	Weeks 1-3	1/10	Everyone
Team Formation	Weeks 1-2	1/14	Everyone
Bootcamp	Week 3	1/24 @ 12 pm	Everyone
Divide and Conquer Document	Weeks 2-3	1/24 @ 12:00 pm	Everyone
Kickoff Meeting (with sponsor and other teams)	Weeks 2-3	N/A	Everyone
Divide and Conquer Meeting	Week 4	1/28 @ 1:30 pm	Everyone
Committee Formation	Weeks 1-3	1/24 @ 12:00 pm	Everyone
Optics Midterm Demo	Week 8	2/25 @ 9:30 - 11:00 am	Gianna
ABET Quizzes	Weeks 8-12	3/7 @ 11:59 pm	Marco, Jet, Derek, and Alex
Midterm Report	Week 10	3/24 @ 12:00 pm	Everyone
Midterm Group Meeting	Week 11	3/28 @ 12:00 pm	Everyone
30-Page Paper Milestone	Weeks 4-7	N/A	Everyone
60-Page Paper Milestone	Weeks 8-12	N/A	Everyone
Sub-System Testing	Weeks 14-16	N/A	Everyone
Optics Final Demo	Week 15	4/15 @ 9:00 - 11:30 am	Gianna
Final Paper Draft	Week 16	4/22 @ 12:00 pm	Everyone
Final Video Demo	Week 16	4/22 @ 12:00 pm	Everyone

Chapter 11: Conclusion

Throughout the development and prototyping of our self-powered radar system, named the Maritime Domain Awareness System (MDAS), many key elements have been designed and, through the prototyping, components such as the Raymarine radar, Starlink Mini, NUC, and Base Station have all been used to communicate with each other. This integration has allowed us to validate not only the physical and electrical connectivity of the system, but also the operational reliability of real-time data transmission and control across devices. By successfully establishing communication between these critical components, we've laid the foundation for a robust and scalable maritime monitoring solution. The system's ability to sustain itself with wind energy while maintaining high-bandwidth connections through Starlink and executing data processing locally via the NUC or Raspberry Pi showcases its potential for long-term, off-grid deployments in remote or maritime environments. Our MDAS system also hopes to aid as a computer-assisted approach to traditional human-based Maritime Domain Awareness (MDA) methods. By leveraging AIS's radio frequency transmission system and protocols, the range of radar detection, with the RayMarine Cyclone Radar it is approximately 72 miles, real-time data transmission of Starlink ship-to-shore, and software-based classification, anomaly detection, and graphical user interfaces we can provide support and reduce much of the tedious work needed in traditional MDA. As the quantity of our MDAS increases, we can provide coverage for an increasing surface area, as well as collect larger amounts of data for better software-based classification and anomaly detection methods such as machine learning or artificial intelligence. Scalability of the MDAS also increases its effectiveness. As more nodes are introduced, we can more easily manage power constraints on each individual node and optimal surface area of MDA coverage, by remotely triggering each node on/off as necessary.

Throughout the design process of the MDAS system, several challenges have emerged that highlight the complexity of developing a self-sustaining maritime platform. One of the most persistent difficulties has been balancing power consumption with available renewable generation, especially given the intermittent nature of wind energy and the relatively high startup and operational loads of the radar and Starlink Mini. Additionally, integrating a diverse range of components—each with different voltage and current requirements—has required careful circuit planning, efficient power conversion, and precise component selection to avoid system instability or inefficiencies. In addition to the hardware and power-related challenges, software development has presented its own set of difficulties—particularly in integrating the Raymarine radar using the proprietary Raymarine SDK. The SDK itself is not openly documented for general use, requiring extensive reverse engineering, trial-and-error testing, and careful analysis of data packets to establish reliable control and data retrieval from the radar unit. Developing a software interface that could communicate with both the radar and the base station proved to be a major hurdle, especially as we worked to establish secure and stable links between the NUC PC, base station, and eventually the Raspberry Pi 5. As the SDK and radar packet transmission remains a hurdle for raw data processing, it continues its challenge into the entirety of the software side of the MDAS. Without the ability to process and visualize radar packets, we cannot begin to create a data set, let alone diagnose the time signatures to create a plan on radar-to-AIS data conversion for comparison and eventual display on

the user interface. As we continue field testing and system refinement in the coming weeks, we expect to gain a much deeper understanding of the radar data structures, improve our ability to parse and process that data reliably, and formalize the downstream workflows required for visualization, classification, and actionable decision support.

While much of the MDAS system has been prototyped and verified in terms of communication and data handling, several critical elements remain to be fully implemented. Chief among these is the power generation system, which will rely on a vertical wind turbine paired with a marine-grade MPPT charge controller and $LiFePO_4$ battery for energy storage. Although the electrical integration has been carefully designed and simulated, physical deployment and real-world performance under varying wind conditions still need to be tested. Additionally, the mechanical aspects of the system—such as the mounting structures for the wind turbine, radar, and antennas—require custom fabrication to ensure durability in harsh marine environments. Proper weatherproof housing for sensitive electronics like the Raspberry Pi, charge controllers, and converters must also be implemented to prevent corrosion, water intrusion, and overheating. Also still needing to be integrated is the imaging system. Physical and simulated tests have been run with test components, but final components have not yet been acquired, meaning that suitable test results have not been fully processed yet. Utilizing Artificial Intelligence/Machine Learning Techniques to augment the MDAS also requires obtaining a dataset for training. While this is feasible, we need to verify our methodology to parse the data received by the radar and categorize it accordingly to be compared with AIS information as this is critical to the MDAS. After confirming that the data can be parsed and compared, we will also simulate and refine our vessel classification algorithm and evaluate the runtime and spatial complexity required. Finalizing these components will be essential for Senior Design 2 when forming a functioning MDAS unit capable of long-term operation at sea.

Appendices

Appendix A - References

[1] United States, Department of Homeland Security. *National Plan to Achieve Maritime Domain Awareness*. Accessed 22 Jan. 2025 https://www.dhs.gov/sites/default/files/publications/HSPD_MDAPlan_0.pdf.

[2]SSR Engineering, Inc. *Shipboard Surveillance Systems – Surface Search Radar*. SSR Engineering, Inc. Accessed 22 Jan. 2025 https://ssreng.com/shipboard-surveillance-system/..

[3] ThayerMahan, Inc. *System: SeaPicket*. Accessed 22 Jan. 2025. https://www.thayermahan.com/systems/seapicket.

[4] HENSOLDT. *Commercial S Band Radar – Kelvin Hughes SharpEye Radar*. Accessed 21 Jan. 2025. https://www.hensoldt.net/products/92/commercial-s-band-radar-kelvin-hughes-sharpeye-radar.

[5] United States Southern Command. *United States Southern Command Official Website*. Accessed 21 Jan. 2025. https://www.southcom.mil/.

[6] Skolnik, Merrill I., editor. *Radar Handbook*. 2nd ed., McGraw-Hill, 1990. Accessed 18 Mar. 2025. https://ftp.idu.ac.id/wp-content/uploads/ebook/tdg/ADNVANCED%20MILITARY%20PLATFORM%20DESIGN/Radar%20Handbook.pdf.

[7] Furuno USA. *Furuno DRS6ANXT (NXT Solid-State Radar, Dome or Open Array)*. Accessed 18 Mar. 2025. https://www.furunousa.com/en/products/drs6anxt.

[8] Garmin Ltd. *Garmin GMR Fantom 126 (6-foot Open Array)*. Accessed 18 Mar. 2025. https://www.garmin.com/en-US/p/597049.

[9] B&G. *HALO20+ Dome Radar (X-band)*. Accessed 18 Mar. 2025. https://www.simrad-yachting.com/simrad/type/radar/halo20simradradar/.

[10] Simrad. *HALO® 2006 Open Array Radar (X-band)*. Accessed 18 Mar. 2025. https://www.simrad-yachting.com/simrad/series/halo-open-array-radar/halo-2006-radarw6-arrayri5020m-cbl/.

[11]Furuno Electric Co., Ltd. *Furuno Radar Basics*. Accessed 18 Mar. 2025. https://www.furuno.com/en/technology/radar/basic/.

[12]ScienceDirect. *Sea Clutter*. Accessed 18 Mar. 2025. https://www.sciencedirect.com/topics/engineering/sea-clutter

[13] Raymarine. *Cyclone Series*. Accessed 18 Mar. 2025. https://www.raymarine.com/en-us/our-products/marine-radar/cyclone/cyclone-radar

[14] Raymarine. *Cyclone Pro Series*. Accessed 18 Mar. 2025. https://www.raymarine.com/en-us/our-products/marine-radar/cyclone/cyclone-pro

[15] TESUP. *TESUP Tri-Blade Wind Turbine*. Accessed 20 Mar. 2025. https://tesup.com/us/tesup-horizontal-wind-turbines-for-homes.

[16] TESUP. *TESUP Atlas Vertical Wind Turbine*. Accessed 20 Mar. 2025. https://tesup.com/us/tesup-vertical-wind-turbines-for-homes.

[17] Nugraha, R., et al. "Comparison of Rose, Aeroleaf, and Tulip Vertical Axis Wind Turbines (VAWTs) for Alternative Electricity Generation." *Energy Reports*, vol. 10, 2024, pp. 1234–1245. Accessed 20 Mar. 2025. https://www.sciencedirect.com/science/article/pii/S2590123024021285.

[18]Flower Turbines. *Tulip Vertical Axis Turbine*. Accessed 20 Mar. 2025 https://www.flowerturbines.com/small-turbine.

[19] TESUP. *Charge Controller*. Accessed 20 Mar. 2025 https://tesup.com/row/spare-parts/charge-controller

[20] ATO. *500W Wind Turbine MPPT Charge Controller, 12V/24V*. Accessed 20 Mar. 2025 https://www.ato.com/500w-wind-turbine-mppt-charge-controller

[21] PIKASOLA Inc. *Pikasola Wind Turbine Charge Controller Mini Wind Turbine Generator Controller IP67 Waterproof 12V/24V Automatic Controller Suitable for 400Watt 500Watt 600Watt Wind Turbine System*. Accessed 20 Mar. 2025 https://www.amazon.com/PIKASOLA-Controller-Generator-Waterproof-Automatic/dp/B07ZNK1C76.

[22] Dakota Lithium. *12V 100Ah Deep Cycle Heated LiFePO4 Battery*. Accessed 20 Mar. 2025. https://dakotalithium.com/product/dakota-lithium-12v-100ah-deep-cycle-lifepo4-marine-solar-battery/

[23] Harbor Freight Tools. *Predator 350 Watt Power Station, 294 Wh Capacity*. Accessed 20 Mar. 2025. https://www.harborfreight.com/350-watt-power-station-294-wh-capacity-70082.html

[24] Amped Outdoors. *24V 50Ah Trolling Motor Lithium Battery (LiFePO4) - Bluetooth - Heated*. Accessed 20 Mar. 2025. https://ampedoutdoors.com/products/50ah-24v-trolling-motor-lithium-battery-lifepo4

[25] MobileMustHave.com. *Starlink Mini PoE DC Power Conversion Kit with 12v Accessory Output*. Accessed 20 Mar. 2025. https://mobilemusthave.com/products/starlink-mini-poe-dc-power-conversion-kit-with-12v-accessory-output-for-peplink

[26] RV&Marine Parts. *Starlink Mini Step Up Power Cable 12V/24V to 30V 3.5A Converter with DC Power Cable*. Accessed 20 Mar. 2025. https://www.amazon.com/Starlink-Power-Converter-Step-Up-Adapter/dp/B0DM5XBY4S

[27] WILLHI LTD. *PlusRoc Waterproof 12V/24V to 5V Converter DC-DC Step Down Module Power Adapter Compatible with Raspberry Pi*. Accessed 20 Mar. 2025. https://www.amazon.com/PlusRoc-Waterproof-Converter-Compatible-Raspberry/dp/B09DGDQ48H

[28] NVIDIA. *Large Language Models Explained*. Accessed 10 Mar. 2025. https://www.nvidia.com/en-us/glossary/large-language-models/

[29] OpenAI. *What is ChatGPT?*. Accessed 10 Mar. 2025. https://help.openai.com/en/articles/6783457-what-is-chatgpt

[30] Appmatics. *The Advantages & Disadvantages of Chat GPT*. Accessed 10 Mar. 2025. https://www.appmatics.com/en/blog/vorteile-nachteile-chat-gpt

[31] Microsoft. *Introducing Microsoft 365 Copilot – Your Copilot for Work*. Accessed 10 Mar. 2025. https://blogs.microsoft.com/blog/2023/03/16/introducing-microsoft-365-copilot-your-copilot-for-work/

[32] Google. *Introducing Gemini: Our Largest and Most Capable AI Model*. Accessed 10 Mar. 2025. https://blog.google/technology/ai/google-gemini-ai/

[33] Proudfoot, Oliver. *What is Claude AI, and How Does It Compare to ChatGPT?*. Pluralsight, Accessed 10 Mar. 2025. https://www.pluralsight.com/resources/blog/ai-and-data/what-is-claude-ai

[34] IEEE. *IEEE Standard C95.1-2019: IEEE Standard for Safety Levels with Respect to Human Exposure to Electric, Magnetic, and Electromagnetic Fields, 0 Hz to 300 GHz*. Accessed 19 Mar. 2025. https://ieeexplore.ieee.org/document/8859679

[35] United States, Department of Homeland Security. "§ 164.46 Automatic Identification System." *Electronic Code of Federal Regulations*, Accessed 19 Mar. 2025. https://www.ecfr.gov/current/title-33/chapter-I/subchapter-P/part-164/section-164.46

[36] United States Coast Guard. *Automatic Identification System (AIS) Overview*. Accessed 19 Mar. 2025. https://www.navcen.uscg.gov/automatic-identification-system-overview

[37] United States Coast Guard. *How AIS Works*. Accessed 19 Mar. 2025. https://www.navcen.uscg.gov/how-ais-works

[38] United States Coast Guard. *AIS Frequently Asked Questions*. Accessed 19 Mar. 2025. https://www.navcen.uscg.gov/ais-frequently-asked-questions

[39]GeeksforGeeks. "Angular vs ReactJS." Accessed 22 Mar. 2025. https://www.geeksforgeeks.org/angular-vs-reactjs/

[40] GeeksforGeeks. "Vue vs Angular." Accessed 22 Mar. 2025. https://www.geeksforgeeks.org/vue-vs-angular/

[41]MDN Web Docs. "REST." Accessed 22 Mar. 2025. https://developer.mozilla.org/en-US/docs/Glossary/REST

[42]MDN Web Docs. "WebSockets API." Accessed 22 Mar. 2025. https://developer.mozilla.org/en-US/docs/Web/API/WebSockets_API

[43] MongoDB Inc. "MongoDB Manual." *MongoDB Documentation*. Accessed 22 Mar. 2025. https://www.mongodb.com/docs/manual/

[44] PostgreSQL Global Development Group. "PostgreSQL Documentation." Accessed 22 Mar. 2025. https://www.postgresql.org/docs/

[45]Redis. "About Redis." Accessed 22 Mar. 2025. https://redis.io/about/

[46] Angular Team. "Angular v17 Documentation." Accessed 22 Mar. 2025. https://v17.angular.io/docs

[47] GeeksforGeeks. "Express.js." Accessed 22 Mar. 2025. https://www.geeksforgeeks.org/express-js/

[48] NestJS. "NestJS Documentation." Accessed 22 Mar. 2025. https://docs.nestjs.com/

[49] Django Software Foundation. "Django Documentation (v5.1)." Accessed 22 Mar. 2025. https://docs.djangoproject.com/en/5.1/

[50] Node.js Foundation. "Node.js API Documentation." Accessed 22 Mar. 2025. https://nodejs.org/docs/latest/api/

[51] DigitalOcean. "DigitalOcean Cloud Hosting." Accessed 22 Mar. 2025. https://www.digitalocean.com/

[52] Amazon Web Services. "What is Amazon Lightsail?" *AWS Documentation*. Accessed 22 Mar. 2025. https://docs.aws.amazon.com/lightsail/latest/userguide/what-is-amazon-lightsail.html

[53] Microsoft. "Azure Developer Overview." *Microsoft Learn*. Accessed 22 Mar. 2025. https://learn.microsoft.com/en-us/azure/developer/intro/azure-developer-overview

[54] Google Cloud. "AlloyDB Product Page." Accessed 22 Mar. 2025. https://cloud.google.com/products/alloydb

[55]Leaflet. "Leaflet.js – JavaScript Mapping Library." Accessed 22 Mar. 2025. https://leafletjs.com/index.html

[56]Kpler. *Maritime Data Services*. Accessed 20 Mar. 2025. https://www.kpler.com/product/maritime/data-services.

[57]Camagni, P., and S. Sandroni, editors. *Optical Remote Sensing of Air Pollution: Lectures of a Course Held at the Joint Research Centre, Ispra (Italy), 12–15 April 1983*. Elsevier Science, 2013. Accessed 20 Mar. 2025. https://books.google.com/books?id=oiElBQAAQBAJ.https://books.google.com/books?id=oiElBQAAQBAJ.

[58] Wang, Cheng, and J. Wang. *LiDAR Remote Sensing and Applications*. Springer, 2022. Accessed 20 Mar. 2025. https://books.google.com/books?id=KxyuEAAAQBAJ

[59]Gao, J., et al. "Analysis of LiDAR Point Cloud Denoising Methods Using Deep Learning." *Computers & Electrical Engineering*, vol. 114, 2024. Accessed 20 Mar. 2025. https://www.sciencedirect.com/science/article/pii/S002980182402626X

[60] Pyroistech. "LiDAR Technology." Accessed 20 Mar. 2025. https://www.pyroistech.com/lidar/

[61] YellowScan. "LiDAR vs. Radar: Understanding the Differences." Accessed 20 Mar. 2025. https://www.yellowscan.com/knowledge/lidar-vs-radar/

[62] Outsight. "How Does LiDAR Compare to Cameras and Radars?" Accessed 20 Mar. 2025. https://insights.outsight.ai/how-does-lidar-compares-to-cameras-and-radars/

[63] MeetOptics. "Image Sensors." Accessed 20 Mar. 2025. https://www.meetoptics.com/academy/image-sensors

[64] Canon Europe. "Image Sensors Explained." Accessed 20 Mar. 2025. https://www.canon-europe.com/pro/infobank/image-sensors-explained/

[65] Keim, R. "Types of CCD Image Sensors: Interline Transfer, Frame Transfer, Full-Frame CCD." *All About Circuits*. Accessed 20 Mar. 2025. https://www.allaboutcircuits.com/technical-articles/types-of-ccd-image-sensors-interline-transfer-frame-transfer-full-frame-ccd/

[66] Baumer. "Operating Principle and Features of CMOS Sensors." Accessed 20 Mar. 2025. https://www.baumer.com/us/en/service-support/function-principle/operating-principle-and-features-of-cmos-sensors/a/EMVA1288

[67] Fujifilm. "Camera Sensors: What Are They and How Do They Work?"

Accessed 20 Mar. 2025.
https://www.fujifilm-x.com/en-us/series/fundamentals-of-photography/camera-sensors-what-are-they-and-how-do-they-work/

[68] Muchmore, A. "What's the Difference Between CMOS, BSI CMOS, and Stacked CMOS?" *PCMag*. Accessed 20 Mar. 2025.
https://www.pcmag.com/how-to/whats-the-difference-between-cmos-bsi-cmos-and-stacked-cmos

[69]Teledyne Vision Solutions. "CCD vs. CMOS: Key Differences."
Accessed 20 Mar. 2025.
https://www.teledynevisionsolutions.com/learn/learning-center/imaging-fundamentals/ccd-vs-cmos/

[70] Photonics Media. "Fiber Optics: Understanding the Basics." Accessed 20 Mar. 2025.
https://www.photonics.com/Articles/Fiber_Optics_Understanding_the_Basics/a25151

[71] Photonics Media. "Adaptive Optics: Taming Atmospheric Turbulence."
Accessed 20 Mar. 2025.
https://www.photonics.com/Articles/Adaptive_Optics_Taming_Atmospheric_Turbulence/a25129

[72] Correia, S., et al. "Enhanced Turbulence Correction Using Adaptive Optics."
Applied Optics, vol. 63, no. 16, 2024, pp. E48–E59. Accessed 20 Mar. 2025.
https://opg.optica.org/ao/fulltext.cfm?uri=ao-63-16-E48&id=549401

[73]Opto Engineering. "Optics Basics." Accessed 20 Mar. 2025.
https://www.opto-e.cn/basics/optics-basics

[74] Teledyne Vision Solutions. "Pixel Size and Camera Resolution."
Accessed 20 Mar. 2025.
https://www.teledynevisionsolutions.com/learn/learning-center/imaging-fundamentals/pixel-size-and-camera-resolution/

[75] Edmund Optics. "Understanding Focal Length and Field of View."
Accessed 20 Mar. 2025.
https://www.edmundoptics.com/knowledge-center/application-notes/imaging/understanding-focal-length-and-field-of-view/

[76]Todd, B. *Nickel-Containing Materials in Marine and Related Environments*.
Presented at the 25th Annual Conference of Metallurgists, Edgbaston, Birmingham, England, 1986. Accessed 23 Mar. 2025
https://nickelinstitute.org/media/1734/nickel_containingmaterialsinmarineandrelatedenvironments_10011_.pdf

[77]Ruvati USA. "304 vs 316 Stainless Steel: Manufacturing Insights."
Accessed 23 Mar. 2025.
https://www.ruvati.com/t-304-vs-t-316-grade-stainless-steel/

[78]“Why 316 Stainless Steel is Ideal for Marine Environments.” Accessed 23 Mar. 2025. https://www.resourcewebsite.com/316-marine-benefits/

[79]Wahid, M. A., A. N. Siddiquee, and Z. A. Khan. “Aluminum Alloys in Marine Construction: Characteristics, Application, and Problems from a Fabrication Viewpoint.” *Marine Systems & Ocean Technology*, vol. 15, Dec. 2020, pp. 70–80. Accessed 23 Mar. 2025. https://link.springer.com/article/10.1007/s40868-019-00069-w

[80]Sathishkumar, T. P., S. Satheeshkumar, and J. Naveen. “Glass Fiber-Reinforced Polymer Composites – A Review.” *Journal of Reinforced Plastics and Composites*, vol. 33, no. 13, 2014, pp. 1258–1275. Accessed 23 Mar. 2025. https://www.researchgate.net/publication/265346634_Glass_fiber-reinforced_polymer_composites_-_a_review

[81]Takim, S. A., and T. F. Okeoma. “Characterization of Glass Fiber Reinforced Epoxy Composites.” *GUU-Journal of Interdisciplinary Research & Innovations*, vol. 1, no. 1, 2022. Accessed 23 Mar. 2025. https://www.researchgate.net/publication/373247517_Characterization_of_Glass_Fiber_Reinforced_Epoxy_Composites

[82]Vizentin, G., D. Glujić, and V. Špada. “Effect of Time-Real Marine Environment Exposure on the Mechanical Behavior of FRP Composites.” *Sustainability*, vol. 13, 2021, p. 9934. Accessed 23 Mar. 2025. https://www.researchgate.net/publication/354406571_Effect_of_Time-Real_Marine_Environment_Exposure_on_the_Mechanical_Behavior_of_FRP_Composites

[83]Tognan, G. “Study of Steel-Glass Fibre-Reinforced Polymer (GFRP) Bonded Joints for the Design of Anchorage Systems for Flat Bottom GFRP Tanks Subjected to Seismic Actions.” Ph.D. dissertation, Università degli Studi di Trieste, 2020. Accessed 23 Mar. 2025. https://www.sciencedirect.com/science/article/abs/pii/S0950061824005622

[84]Dexter, S. C. *Handbook of Oceanographic Materials*. Krieger Publishing, 1985. Accessed 23 Mar. 2025. https://www.usna.edu/NAOE/_files/documents/Courses/EN380/Course_Notes/zAppendix_B_Material_Properties.pdf

[85]Pistone, A., C. Scolaro, and A. Visco. “Mechanical Properties of Protective Coatings Against Marine Fouling: A Review.” *Polymers*, vol. 13, 2021, p. 173. Accessed 23 Mar. 2025. https://www.researchgate.net/publication/348299356_Mechanical_Properties_of_Protective_Coatings_against_Marine_Fouling_A_Review

[86]Santos-Simón, M., J. Ferrario, B. Benaduce-Ortiz, M. Ortiz-Zarragoitia, and A. Marchini. “Assessment of the Effectiveness of Antifouling Solutions for Recreational Boats in the Context of Marine Bioinvasions.” *Marine Pollution Bulletin*, vol. 200, 2024, Art. no. 116108. Accessed 23 Mar. 2025. https://www.sciencedirect.com/science/article/pii/S0025326X24000857

[87]Sørensen, P. A., S. Kiil, K. Dam-Johansen, and C. E. Weinell. “Anticorrosive Coatings: A Review.” *Journal of Coatings Technology and Research*, vol. 6, no. 2, 2009, pp. 135–176. Accessed 23 Mar. 2025. https://www.researchgate.net/publication/225668165_Anticorrosive_coatings_A_review

[88]Santos-Simón, M., J. Ferrario, B. Benaduce-Ortiz, M. Ortiz-Zarragoitia, and A. Marchini. “Assessment of the Effectiveness of Antifouling Solutions for Recreational Boats in the Context of Marine Bioinvasions.” *Marine Pollution Bulletin*, vol. 200, 2024, Art. no. 116108. Accessed 23 Mar. 2025. https://www.sciencedirect.com/science/article/pii/S0025326X24000857

[89] Avantier Inc. “Understanding Field of View (FOV) and Angular Field of View (AFOV) in Camera Lenses.” Accessed 23 Mar. 2025. https://avantierinc.com/resources/knowledge-center/understanding-field-of-view-fov-and-angular-field-of-view-afov-in-camera-lenses/

[90] Lumen Learning. “Telescopes.” *University Physics*. Accessed 23 Mar. 2025. https://courses.lumenlearning.com/suny-physics/chapter/26-5-telescopes/

[91] Newport Corporation. “How to Build a Beam Expander.” Accessed 10 Apr. 2025. https://www.newport.com/n/how-to-build-a-beam-expander

[92] European Southern Observatory (ESO). “Adaptive Optics.” *Extremely Large Telescope (ELT)*. Accessed 10 Apr. 2025.

https://elt.eso.org/telescope/adaptiveoptics/.

[93] CAD Models by Kevin Micelo - E4D MDAS Team

[94] Lodriguss, J. "Astrophotography Formulae." *Astropix*. Accessed 10 Apr. 2025. https://www.astropix.com/html/astrophotography/astrophotography-formulae.html

[95] Nguyen, Duong, et al. "GeoTrackNet—A Maritime Anomaly Detector Using Probabilistic Neural Network Representation of AIS Tracks and A Contrario Detection." *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 6, 2022, pp. 5655–5667. Accessed 10 Apr. 2025 https://arxiv.org/pdf/1912.00682

[96] Pallotta, G., Vespe, M., and Bryan, K. "Vessel Pattern Knowledge Discovery from AIS Data: A Framework for Anomaly Detection and Route Prediction." *Entropy*, vol. 15, no. 6, 2013, pp. 2218–2245. Accessed 10 Apr. 2025. https://doi.org/10.3390/e15062218

[97]Nguyen, D., Vadaine, R., Hajduch, G., Garello, R., and Fablet, R. "A Multi-Task Deep Learning Architecture for Maritime Surveillance Using AIS Data Streams." *2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA)*, Turin, Italy, 2018, pp. 331–340. Accessed 10 Apr. 2025 https://arxiv.org/pdf/1806.03972

[98] "A Security Evaluation of AIS Automated Identification System." *ACM Digital Library*. Accessed 10 Apr. 2025. https://dl.acm.org/doi/abs/10.1145/2664243.2664257

[99] Wang, Y., et al. "A Multi-Feature Ensemble Learning Classification Method for Ship Classification with Space-Based AIS Data." *Applied Sciences*, vol. 11, no. 21, 2021, p. 10336. Accessed 10 Apr. 2025. https://doi.org/10.3390/app112110336

[100] Meyer, R., and Kleynhans, W. "Vessel Classification Using AIS Data." *Ocean Engineering*, vol. 319, 2025, Article no. 120043. Accessed 10 Apr. 2025. https://doi.org/10.1016/j.oceaneng.2024.120043

[101] Amazon Web Services. *Amazon Cognito Documentation*. Accessed 13 Apr. 2025. https://docs.aws.amazon.com/cognito/

[102] Amazon Web Services. *Amazon Cognito Features*. Accessed 13 Apr. 2025. https://aws.amazon.com/cognito/features/

[103] Google. *Firebase Authentication Documentation*. Accessed 13 Apr. 2025. https://firebase.google.com/docs/auth

[104] Google. *Get Started with Firebase Authentication on Web*. Accessed 13 Apr. 2025. https://firebase.google.com/docs/auth/web/start

[105] Google Cloud. *Identity Platform Product Comparison*. Accessed 13 Apr. 2025. https://cloud.google.com/identity-platform/docs/product-comparison

[106] Auth0. *Auth0 Documentation*. Accessed 13 Apr. 2025. https://auth0.com/docs

[107] Auth0. *Auth0 Features Overview*. Accessed 13 Apr. 2025. https://auth0.com/features

Appendix B - Copyright Permission

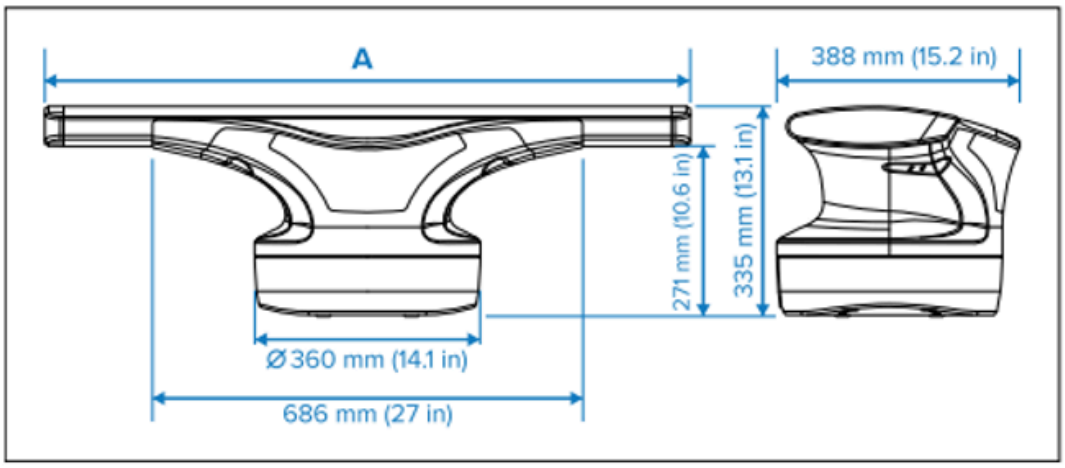
[1] United States Southern Command Emblem on Title Page

Received verbal permission by our sponsor, Roy Wubker. Contact information for Roy: roy.h.wubker2.ctr@mail.mil

Appendix C - Data Sheet

[1] References from RayMarine

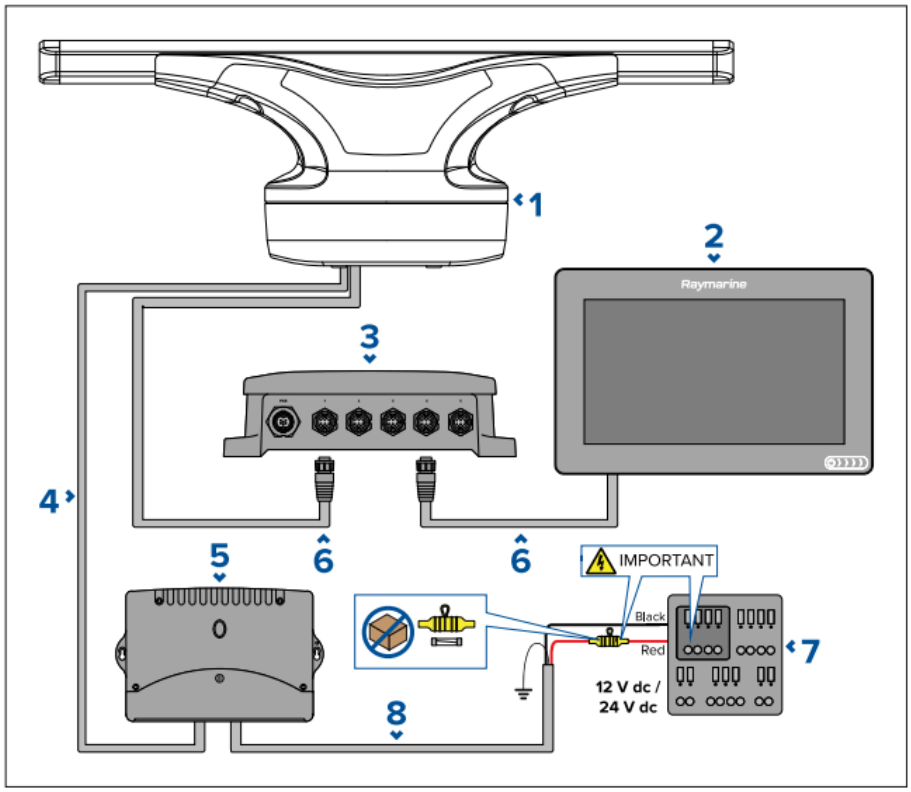
Cyclone series

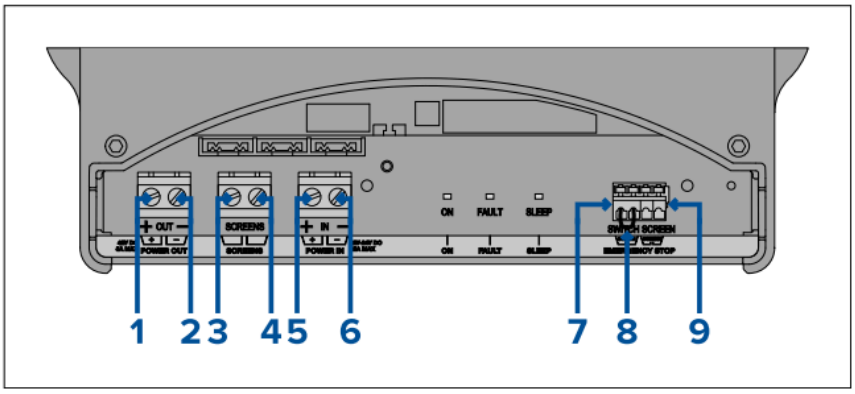


Dimension A — Antenna size is determined by model:

Model	Dimension
E70628 – (36 in array)	1031 mm (40.5 in)
E70629 – (48 in array)	1336 mm (52.5 in)
E70630 – (72 in array)	1945 mm (76.5 in)

Radar connected using RayNet switch





1. **POWER OUT (Positive)** — connect to the RED wire of the Radar's power cable.
2. **POWER OUT (Negative)** — connect to the BLACK wire of the Radar's power cable.
3. **SCREEN** — connect to the bare screen (drain) strands of the Radar's power cable.
4. **SCREEN** — connect to your vessel's RF ground system.
5. **POWER IN (Positive)** — connect to the positive terminal of the DC distribution panel or battery isolator switch.
6. **POWER IN (Negative)** — connect to the negative battery terminal.
7. **EMERGENCY STOP (Switch)** — if you have the optional VCM100 emergency stop button, remove the wire bridging link from the VCM100 EMERGENCY STOP terminals, and connect the emergency stop button SWITCH wire to the VCM100 EMERGENCY STOP SWITCH terminal.
8. **EMERGENCY STOP wire bridging link** — only remove if fitting the optional emergency stop button.
9. **EMERGENCY STOP (Screen)** — if you have the optional VCM100 emergency stop button, remove the wire bridging link from the VCM100 EMERGENCY STOP terminals, and connect the emergency stop button SCREEN (drain) wire to the VCM100 EMERGENCY STOP SCREEN terminal.

[2] optics testing CMOS sensor Specs

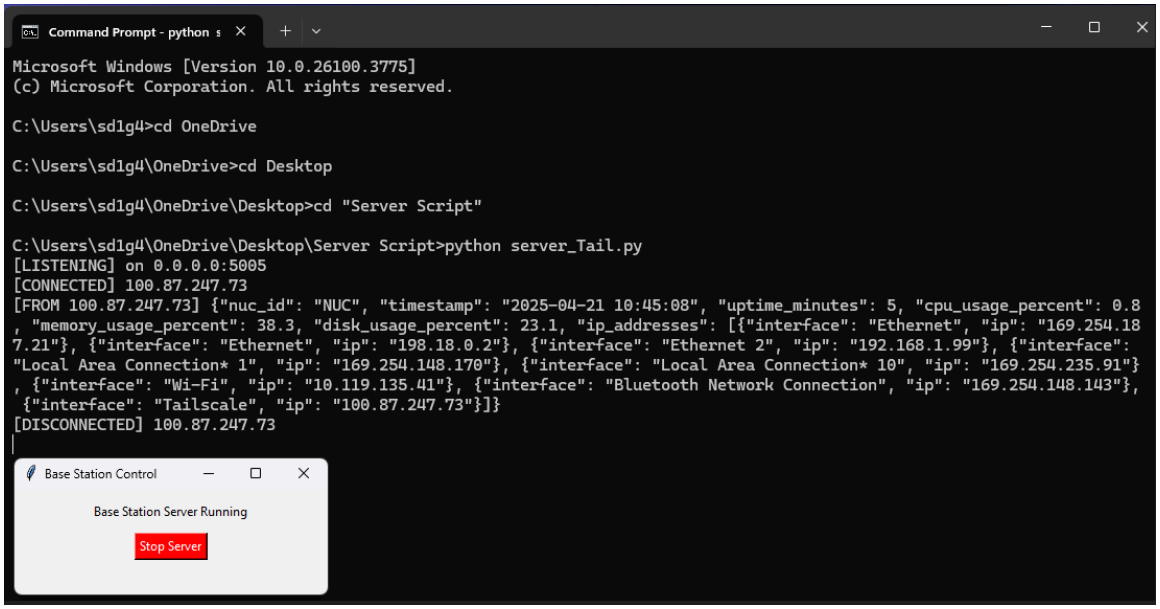
Thorlabs CS165CU CMOS Sensor:

Sensor Specifications	
Sensor Type	CMOS Color
Number of Active Pixels	1440 (H) x 1080 (V) (~1.6 MP)
Pixel Size	3.45 µm x 3.45 µm
Optical Format	1 / 2.9 Format (4968 µm x 3726 µm)
Peak Quantum Efficiency	65% at 535 nm
Dynamic Range	Up to 69 dB
Full Well Capacity	≥11,000 e^-
Shutter Type	Global

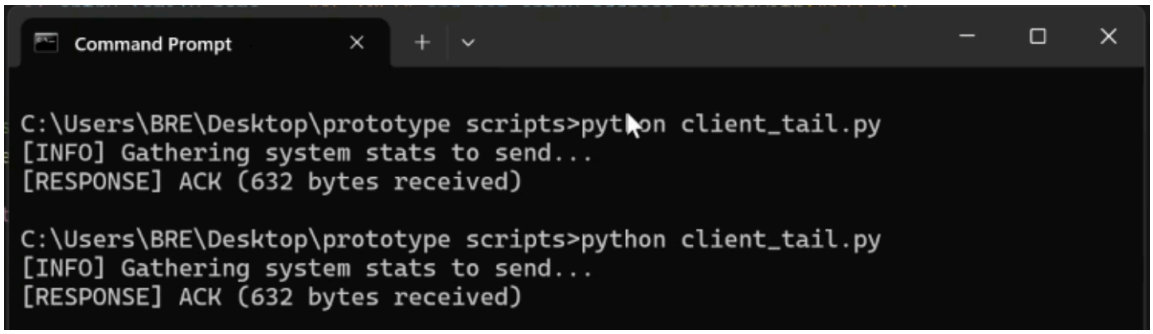
[3] Allied Vision CMOS sensor

Sensor	
Image Buffer:	256KB
Imaging Sensor:	ON Semi PYTHON 480
Type of Sensor:	Progressive Scan CMOS
Type of Shutter:	Global
Camera Sensor Format:	1/3.6"
Resolution (MegaPixels):	0.50
Pixels (H x V):	808 x 608
Pixel Size, H x V (µm):	4.8 x 4.8
Sensing Area, H x V (mm):	3.9 x 2.9
Frame Rate (fps):	116.00
Pixel Depth:	8/10 bit
Exposure Time:	64µs - 10s
Dynamic Range (dB):	54
Machine Vision Standard:	USB3 Vision v1.0, GenICam

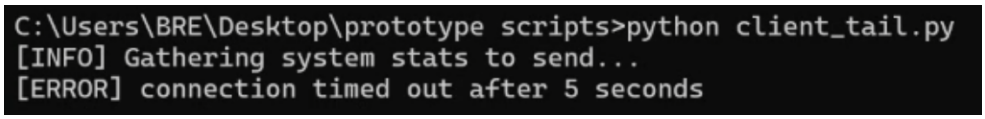
[4] Base Station Server Script Receiving Information From NUC



[5] NUC Client Side Sent Data With Acknowledgment



[6] NUC Client Side Timeout Error Message



Appendix D - Software Code

[1] server_tail.py - Ran on the base station computer to host a communication server.

```
# imports for networking, threading, logging, and gui
import socket
import threading
import datetime
import os
import tkinter as tk
import sys

# accept connections from any ip
HOST = "0.0.0.0"
PORT = 5005
LOG_FILE = "nuc_message_log.txt"

# flag to shut down server from gui
shutdown_event = threading.Event()

# create log file if it doesn't exist
if not os.path.exists(LOG_FILE):
        with open(LOG_FILE, "w") as f:
    f.write("Timestamp\tNUC_IP\tMessage_Size\tMessage_Content\n")

# write system events like startup or shutdown to file
def log_event(event_str):
        timestamp = datetime.datetime.now().strftime('%Y-%m-%d %H:%M:%S')
        with open(LOG_FILE, "a") as f:
    f.write(f"{timestamp}\tSYSTEM\t-\t{event_str}\n")

# write each message received from nuc to log file
def log_message(nuc_ip, message):
```

```
            timestamp = datetime.datetime.now().strftime('%Y-%m-%d %H:%M:%S')
    message_size = len(message.encode('utf-8'))
            with open(LOG_FILE, "a") as f:
        f.write(f"{timestamp}\t{nuc_ip}\t{message_size} bytes\t{message}\n")

# handle each client that connects to the server
def handle_client(conn, addr):
            nuc_ip = addr[0]
            print(f"[CONNECTED] {nuc_ip}")
            try:
            while not shutdown_event.is_set():
          data = conn.recv(1024)
            if not data:
             break
          decoded = data.decode().strip()
            print(f"[FROM {nuc_ip}] {decoded}")
          log_message(nuc_ip, decoded)

            # respond to ping message
            if decoded.lower() == "ping":
             response = "pong"
            else:
             response = f"ACK ({len(decoded)} bytes received)"
          conn.sendall(response.encode())
            except Exception as e:
            print(f"[ERROR] with {nuc_ip}: {e}")
            finally:
        conn.close()
            print(f"[DISCONNECTED] {nuc_ip}")

# start the server and listen for connections
```

```
def start_server():
        with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as server:
    server.bind((HOST, PORT))
    server.listen()
        print(f"[LISTENING] on {HOST}:{PORT}")
    log_event("Server started")

        while not shutdown_event.is_set():
        try:
         server.settimeout(1.0)
         conn, addr = server.accept()
         thread = threading.Thread(target=handle_client, args=(conn, addr))
         thread.start()
        except socket.timeout:
         continue
        except Exception as e:
         print(f"[SERVER ERROR] {e}")
         break

        print("[SHUTDOWN] Server is stopping...")
    log_event("Server shutdown")

# create simple gui to stop the server
def create_gui():
        def stop_server():
    shutdown_event.set()
    root.destroy()

        root = tk.Tk()
  root.title("Base Station Control")
  root.geometry("300x100")
```

```
        label = tk.Label(root, text="Base Station Server Running")
    label.pack(pady=10)

    stop_button = tk.Button(root, text="Stop Server", command=stop_server, fg="white", bg="red")
    stop_button.pack()

    root.protocol("WM_DELETE_WINDOW", stop_server)
    root.mainloop()

# run the server and gui together
if __name__ == "__main__":
    threading.Thread(target=start_server, daemon=True).start()
        create_gui()
```

[2] client_tail.py - Ran on the NUC computer to communicate with the basestation.

```
# imports for networking and system data
import socket
import platform
import psutil
import time
import datetime
import json
import os

# base station tailscale ip and port
BASE_IP = "100.110.64.11"
PORT = 5005

# gather system info like cpu and memory
def get_system_info():
        info = {
```

```
            "nuc_id": platform.node(),  # hostname
            "timestamp": datetime.datetime.now().strftime('%Y-%m-%d %H:%M:%S'),
            "uptime_minutes": int(time.time() - psutil.boot_time()) // 60,
            "cpu_usage_percent": psutil.cpu_percent(interval=1),
            "memory_usage_percent": psutil.virtual_memory().percent,
            "disk_usage_percent": psutil.disk_usage("/").percent,
            "ip_addresses": get_ip_addresses()
            }

            # try to get temperature sensors if available
            if hasattr(psutil, "sensors_temperatures"):
            temps = psutil.sensors_temperatures()
            if temps:
            for sensor_name, entries in temps.items():
             for entry in entries:
                info[f"temp_{sensor_name}_{entry.label or entry.device}"] = entry.current

            return json.dumps(info)

# find all non-local ip addresses
def get_ip_addresses():
            addrs = psutil.net_if_addrs()
            ip_list = []
            for iface, entries in addrs.items():
            for entry in entries:
            if entry.family.name == "AF_INET" and not entry.address.startswith("127."):
             ip_list.append({"interface": iface, "ip": entry.address})
            return ip_list

# send message to base station and print reply
def send_message(message):
```

```
        try:
        with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
        # set timeout for send and receive
      s.settimeout(5)
      s.connect((BASE_IP, PORT))
      s.sendall(message.encode())
      response = s.recv(1024).decode()
        print(f"[RESPONSE] {response}")
        except socket.timeout:
        print("[ERROR] connection timed out after 5 seconds")
        except Exception as e:
        print(f"[ERROR] {e}")


# main code to run when script starts
if __name__ == "__main__":
        print("[INFO] Gathering system stats to send...")
        message = get_system_info()
        send_message(message)
```

[3] demo3.mlx - Calculates angular separation of the system

```
% Resolution Analysis
clear; clc; close all;


% Known Variables
pixel_size = 3.45e-6;       % meters (sensor pixel size)
target_spacing_m = 0.0595;   % meters (59.5 mm)
distance_to_target = 1;      % meters
row_to_sample = 540;         % adjust if needed


% Load Image
[filename, pathname] = uigetfile({'*.png;*.jpg;*.tif'}, 'Select resolution image');
img = imread(fullfile(pathname, filename));
```

```
if size(img, 3) == 3
        img_gray = rgb2gray(img);
else
        img_gray = img;
end

% Extract horizontal intensity profile
intensity_profile = double(img_gray(row_to_sample, :));

% Find peaks
[peaks, locs] = findpeaks(intensity_profile, 'MinPeakProminence', 10);
num_peaks = numel(locs);

if num_peaks >= 2
        % Calculate measured pixel spacing and angular separation
        measured_pixels = abs(locs(2) - locs(1));
        measured_sep_m = measured_pixels * pixel_size;
        measured_angle_rad = measured_sep_m / distance_to_target;
   measured_angle_deg = rad2deg(measured_angle_rad);

        % Ground-truth angular separation
        true_angle_rad = target_spacing_m / distance_to_target;
        true_angle_deg = rad2deg(true_angle_rad);

        % Sensor-limited angular resolution (2 pixels)
        sensor_limit_rad = (2 * pixel_size) / 150;
        sensor_limit_deg = rad2deg(sensor_limit_rad);

        % Print results
        fprintf('--- Angular Resolution Evaluation ---\n');
        fprintf('Target spacing: %.2f mm\n', target_spacing_m * 1e3);
```

```
        fprintf('Measured pixel separation: %d px\n', measured_pixels);
        fprintf('Measured separation: %.2f mm\n', measured_sep_m * 1e3);
        fprintf('Measured angular separation: %.3f°\n', measured_angle_deg);
        fprintf('Expected angular separation: %.3f°\n', true_angle_deg);
        fprintf('Sensor-limited resolution (2 px): %.4f°\n', sensor_limit_deg);
else
        warning('Less than two peaks detected. Adjust sampling row or image contrast.');
end


% Plot the intensity profile
figure;
plot(intensity_profile, 'k', 'LineWidth', 1.5); hold on;
plot(locs, peaks, 'ro', 'MarkerSize', 8, 'LineWidth', 1.5);
xlabel('Pixel Index'); ylabel('Intensity');
title('Intensity Profile Across 59.5 mm Target');
legend('Intensity', 'Detected Peaks'); grid on;
```

[4] event_data queue

```
event_data = {
   "timestamp": "2025-04-10 13:45:22",
   "gps_location": {"lat": 25.774, "lon": -80.193},
   "vessel_info": "UNKNOWN / Dark Ship",
   "radar_signature": {...},
   "camera_angle": 32.5
}
```

[5] server.js - This JavaScript file sets up a basic Express server that handles JSON requests and routes login-related API calls through a separate loginRoute module on port 5000.

```
const express = require("express");
const bodyParser = require("body-parser");
const cors = require("cors");
```

```
const app = express();

const loginRoute = require("./routes/login");


app.use(bodyParser.json());

app.use("/api", loginRoute);

app.listen(5000);
```

[6] login.js -This JavaScript file defines the backend logic for our prototype login route, handling CORS setup and verifying user credentials against a hardcoded user list using a POST request.

```
const express = require("express");

const cors = require("cors");

const router = express.Router();

router.use((req, res, next) => {

 res.setHeader("Access-Control-Allow-Origin", "*");

 res.setHeader(

  "Access-Control-Allow-Headers",

  "Origin, X-Requested-With, Content-Type, Accept, Authorization"

 );

 res.setHeader(

  "Access-Control-Allow-Methods",

  "GET, POST, PATCH, DELETE, OPTIONS"

 );

 next();

});

 const users = [

  { id: 0, username: "mdasAdmin", password: "mdas8888", role: "admin" },

];

 router.post("/login", (req, res, next) => {

  const { user, pass } = req.body;

  let error = "";

  let id = -1;

  let username;
```

```
  let password;
  let role = "admin" | "user";
  try {
    const findUser = users.find(
      (u) => u.username === user && u.password === pass
    );
    if (!findUser) {
      return res.status(401).json({ error: "Invalid username or password" });
    }
    id = findUser.id;
    username = findUser.username;
    password = findUser.password;
  } catch (err) {
    error = err.toString();
  }
  let ret = { id: id, username: username, password: password };
  res.status(201).json(ret);
});
module.exports = router;
```

[7] main.tsx - This React entry point sets up the root of the application, wrapping the App component with StrictMode for highlighting potential problems and a custom Provider for shared UI context or theming.

```
import { StrictMode } from "react";
import { createRoot } from "react-dom/client";
import { Provider } from "./components/ui/provider.tsx";
import App from "./App.tsx";


createRoot(document.getElementById("root")!).render(
  <StrictMode>
    <Provider>
      <App />
    </Provider>
```

```
  </StrictMode>
);
```

[7] App.tsx -This React App component sets up client-side routing using React Router, defining routes for the login and dashboard pages, and includes a Toaster component for displaying user notifications.

```
import { BrowserRouter, Route, Routes } from "react-router-dom";
import Login from "./pages/LoginPage";
import Dashboard from "./pages/DashboardPage";
import { Toaster } from "./components/ui/toaster";
export default function App() {
  return (
    <>
      <Toaster />
      <BrowserRouter>
        <Routes>
          <Route path="/" element={<Login />} />
          {/* TODO: Add Sign Up */}
          <Route path="/dashboard" element={<Dashboard />} />
          {/* TODO: Add Page Not Found */}
        </Routes>
      </BrowserRouter>
    </>
  );
}
```

[8] LoginPage.tsx -This file defines the Login page, which simply renders the DoLogin component responsible for handling the login UI and logic.

```
import DoLogin from "../components/LoginComponent";

export default function Login() {
  return <DoLogin />;
}
```

[9] LoginComponent.tsx -This file implements the main login UI and logic using Chakra UI components. It collects user credentials, sends them to the backend for authentication, displays success or error messages via a toaster system, and redirects to the dashboard upon successful login.

```
import DoLogin from "../components/LoginComponent";


export default function Login() {
 return <DoLogin />;
}


import { ChangeEvent, FormEvent, useState } from "react";
import {  Box, Button, Flex, Heading, Input, VStack, InputGroup, Text } from "@chakra-ui/react";
import { toaster } from "./ui/toaster";
import { FormControl, FormLabel } from "@chakra-ui/form-control";
import { useNavigate } from "react-router-dom";


import { BsPersonLock } from "react-icons/bs";
import { BsPerson } from "react-icons/bs";


import "@fontsource/inter/400.css";
import "@fontsource/inter/600.css";
import "@fontsource/quicksand/500.css";


type loginUser = { username: string; password: string };


export default function DoLogin() {
 const navigate = useNavigate();
 const [user, setUser] = useState<string>("");
 const [password, setPassword] = useState<string>("");


 async function handleLogin(e: FormEvent, { username, password }: loginUser) {
  e.preventDefault();
```

```
try {
  const response = await fetch("http://localhost:5000/api/login", {
    method: "POST",
    headers: { "Content-Type": "application/json" },
    body: JSON.stringify({ user: username, pass: password }),
  });

  if (!response.ok) {
    const errorData = await response.json();
    toaster.create({
      title: "Login Failed",
      description: errorData?.error || "Unknown error",
      type: "error",
    });
    setUser("");
    setPassword("");
    return;
  }

  const data = await response.json();
  toaster.create({
    title: "Login Successful",
    description: "Welcome back!",
    type: "success",
  });

  navigate("/dashboard");
} catch (err: unknown) {
  toaster.create({
    title: "Network Error",
```

```
      description: err instanceof Error ? err.message : "Unexpected error",
      type: "error",
    });
  }
}

return (
  <>
    <Flex
      bgImage="url(LoginBackground.svg)"
      bgSize="cover"
      bgRepeat="no-repeat"
      minH="100vh"
      direction="column"
      align="center"
      justify="center"
      p={4}
      gap={6}
    >
      <Box
        bg="white"
        px={4}
        py={2}
        borderRadius="md"
        boxShadow="md"
        textAlign="Center"
        maxW="md"
        w="full"
      >
        <Heading mb={6} fontSize="3xl" color="blue.900" fontFamily="'Inter'">
          Maritime Domain Awareness
```

```
  </Heading>
</Box>

<Box
  bg="white"
  p={8}
  borderRadius="lg"
  boxShadow="lg"
  width="full"
  maxW="md"
>
  <Heading
    mb={6}
    textAlign="center"
    size="2xl"
    color="blue.800"
    fontFamily="'Inter', sans-serif"
  >
    Welcome Back,
    <Text
      fontSize="sm"
      color="gray.500"
      fontStyle="italic"
      textAlign="center"
      fontFamily="'Inter', sans-serif"
    >
      Please log in using your credentials below!
    </Text>
  </Heading>
  <form
    onSubmit={(e) =>
```

```
    handleLogin(e, { username: user, password: password })
  }
>
  <VStack>
    <FormControl isRequired>
      <FormLabel color="black" fontSize="medium">
        Username
      </FormLabel>
      <InputGroup startElement={<BsPerson color="navy" />}>
        <Input
          type="text"
          color={"black"}
          value={user}
          onChange={(u: ChangeEvent<HTMLInputElement>) =>
            setUser(u.target.value)
          }
          placeholder="Enter your username"
        />
      </InputGroup>
    </FormControl>

    <FormControl isRequired>
      <FormLabel color="black" fontSize="medium">
        Password
      </FormLabel>
      <InputGroup startElement={<BsPersonLock color="navy" />}>
        <Input
          color={"black"}
          type="password"
          value={password}
          onChange={(p: ChangeEvent<HTMLInputElement>) =>
```

```
            setPassword(p.target.value)
          }
          placeholder="Enter your password"
        />
      </InputGroup>
    </FormControl>

    <Button
      bg="blue.800"
      color="white"
      _hover={{ bg: "blue.700" }}
      height="2.5rem"
      width="11.5rem"
      fontSize="sm"
      fontWeight="semibold"
      py={4}
      px={6}
      borderRadius="md"
      boxShadow="md"
      type="submit"
      mt={5}
    >
      Login
    </Button>
  </VStack>
 </form>
 </Box>
 </Flex>
 </>
 );
}
```

[10] DashboardPage.tsx -This file defines the Dashboard page, which displays a simple heading and renders the Map component to visualize data on the main dashboard.

```
import Map from "../components/MapComponent";


export default function Dashboard() {
 return (
  <>
   <h1>Main Dashbaord</h1>
   <Map />
  </>
 );
}
```

[11] MapComponent.tsx -This file defines the Map component using React Leaflet, rendering an interactive map centered near Key West with OpenSeaMap and OpenStreetMap layers, along with a marker and customizable popup.

```
import { MapContainer, TileLayer, Marker, Popup } from "react-leaflet";
import "leaflet/dist/leaflet.css";


export default function Map() {
 return (
  <MapContainer
   center={[24.5551, -81.78]}
   zoom={7}
   scrollWheelZoom={false}
   style={{ height: "600px", width: "100%" }}
  >
   <TileLayer
    attribution='Map data: &copy; <a href="http://www.openseamap.org">OpenSeaMap</a> contributors'
    url="https://tiles.openseamap.org/seamark/{z}/{x}/{y}.png"
   />
   <TileLayer
    attribution='&copy; <a href="https://www.openstreetmap.org/copyright">OpenStreetMap</a> contributors'
```

```
        url="https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png"
      />
      <Marker position={[24.5551, -81.78]}>
        <Popup>
          A pretty CSS3 popup. <br /> Easily customizable.
        </Popup>
      </Marker>
    </MapContainer>
  );
}
```

Appendix E - ChatGPT prompts

[1] Boat Image Generated Using Chat GPT

Prompt: Please generate a picture of a small ship overseas.

[2] Frontend help using Claude

Prompt: How should I set up my React components for a dashboard application that displays live radar data and vessel overlays?

[3] Material Selection Using Google Gemini

Prompt1: What structural building materials (give me a list of 5) would be best for designing a maritime structure that is corrosion resistant and can uphold and encapsulate the system in the divide and conquer paper.

Prompt2:My structure contains 4 components, 1. The structural frame that upholds the radar and encases a vertical wind turbine within it. 2. The top plate on which the radar pedestal is mounted on top. 3. The bottom plate which gets mounted onto the boat. 4. The electronics enclosure which holds the technology to process data and the starlink module. Please provide me with a list of possible material selections for each component.

[4] Radar Features/ Selection Case Study Using Google Gemini

Prompt1: I am researching radar comparison, including features and specifications needed for a senior design project involving a maritime radar. This radar will be connected to a NUC/ raspberry pi computer and will collect vessel data overseas. I need to look into what factors are important for radar overseas, what kind of radars work best, and then compare say 5 different radars ultimately choosing the best option for this project. Attached is a document encasing our senior design plan thus far.

Prompt2: Do a much deeper analysis with more in depth information. Please pick the best radar option.

[5] Optical Case Study using ChatGPT

Prompt 1: Write an outline for comparing Keplerian and Galilean telescopes and include a chart with comparison features.

Prompt 2: Would a CCD or CMOS sensor be better for a camera system mounted on a boat? Which type of CCD or CMOS sensor would be best for this scenario?

Prompt 3: Create a table comparing a front-illuminated CMOS, back-illuminated CMOS, a stacked CMOS sensor, and global shutter CMOS system.

[6] Embedded Case Study with Chat GPT

Prompt 1: I am looking into the NMEA 2000 protocol for a senior design project. Talk to me about its hardware and software benefits/ drawbacks.

[7] Turbine Perspective with Deep Seek

Prompt 1: Discuss vertical versus horizontally mounted wind turbines, and how they would be used.

Prompt2: How durable are they in a maritime environment? What about mounting requirements?

[8] Research Ideas Related to Standards using ChatGPT

Prompt 1: I am working on a senior design project involving a marine X-band radar and want to write a standard on electromagnetic safety or things closely related. What 3 standards are the most applicable to this project?

Prompt 2: For each of the three standards identified, explain how it applies to the use of marine X-band radar systems in terms of compliance, deployment, and safety. Highlight any specific technical thresholds like safe exposure limits, frequency restrictions, or emission power levels that would be important.

[9] Interfacing a Server Connection Without Port Forwarding using ChatGPT

Prompt1: What are some possible ways like VPNS and tunneling that I can utilize a starlink router terminal to achieve a connection to a server hosted by another internet connection?

Prompt2: Can you please recommend me specific software and guide me through establishing a good handshake connection between two systems, they should both be on separate networks but still be able to communicate with each other without opening Ports.