Software Requirements Specification

for

AErial Routing & Operational LOGistics Integration Code (AEROLOGIC)

Version 0.1.0

Prepared by Carson Frankovich

**Student Unmanned Ariel Systems Team at
Embry-Riddle Aeronautical University**

December 20, 2023

Table of Contents

1. INTRODUCTION

1.1 Purpose and Scope

This Software Requirements Specification (SRS) document details the requirements for the "Aerial Routing and Operational Logistics Integration Code," hereafter referred to as "AEROLOGIC." Developed for the Student Unmanned Aerial Systems (SUAS) Competition [1.5.1]. It aims to define the software functionalities and specifications necessary for autonomous flight, navigation, obstacle avoidance, and precise package delivery tasks stipulated in the SUAS 2024 Competition rules.

1.2 Document Conventions

This SRS uses **bold** for section headings and *italic* for emphasis. Each requirement is uniquely identified with a tag, such as R 1.1, R 4.7, etc., for easy reference. Blue underlined text indicates hyperlinks or references to other sections or documents. Readers should refer to the glossary for definitions of technical terms.

Each version of this document is systematically numbered to track updates and changes over time. The versioning follows a Major.Minor.Patch scheme. A change in the "Major" number indicates significant revisions or overhauls, the "Minor" number reflects smaller updates or additions, and the "Patch" number denotes minor edits or corrections. Historical versions can be found on the Documentation GitHub Repository [1.5.4]. Versions starting with 0.X.X are considered unfinished.

1.3 Intended Audience and Reading Suggestions

This document is primarily intended for the current developers of the AEROLOGIC software, as well as future developers and contributors who wish to understand or enhance the software's functionalities. It is also suitable for project managers overseeing the development lifestyle and testers who will validate the software implementation against these requirements.

The SRS is structured to first present an introduction to AEROLOGIC, followed by a detailed overall description, and then delving into specific external interface requirements and system features. We recommend that all readers start with the introduction to understand the purpose, scope, and context of AEROLOGIC. Developers may then focus on Sections 2 and 4, which detail the product functions and system features.

1.4 Product Scope

AEROLOGIC is a software package designed for the SUAS 2024 Competition, aimed at empowering a UAS to fulfill its mission objectives effectively. The software has a focus on achieving autonomous flight, navigation, obstacle detection, and payload delivery.

1.5 References

1.5.1 Student Unmanned Aerial Systems 2024 Competition Rulebook

https://suas-competition.org/s/suas-2024-02-rules

1.5.2 Core Avionics GitHub Repository

https://github.com/ERAU-SUAS/core-avionics

1.5.3 Vision Processing GitHub Repository

https://github.com/ERAU-SUAS/vision-processing

1.5.4 Documentation GitHub Repository

https://github.com/ERAU-SUAS/documentation

1.5.5 AEROLOGIC Software Design Document

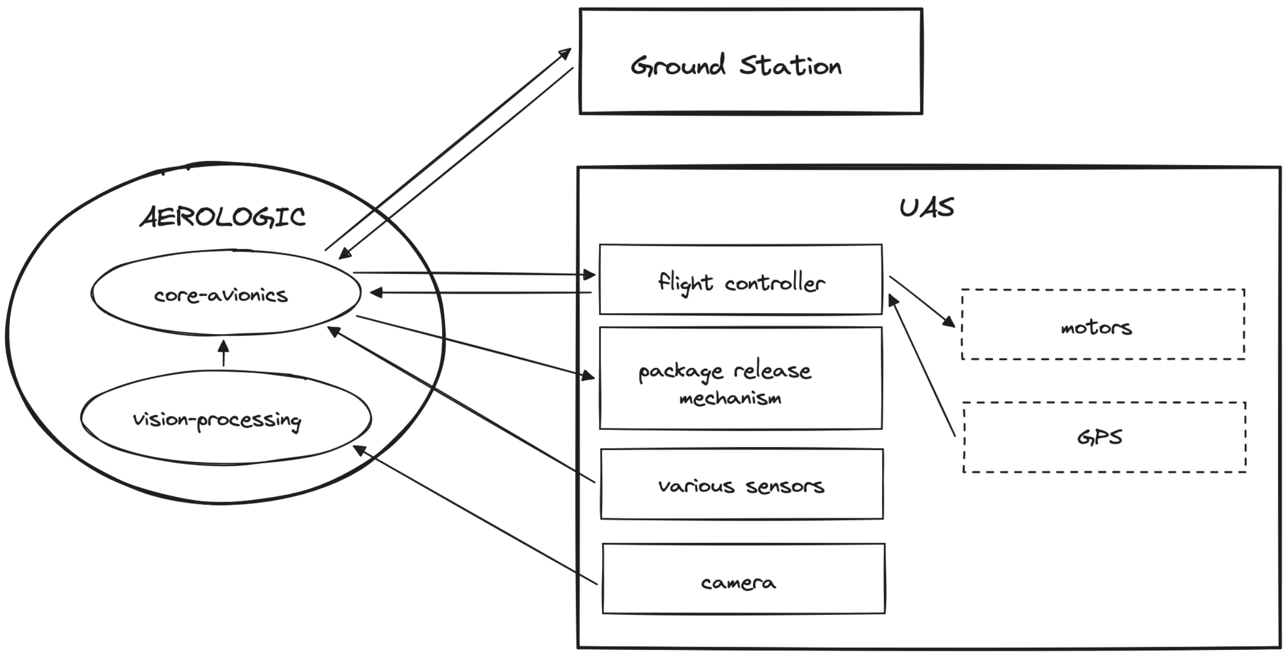
<TBD>

2. OVERALL DESCRIPTION

2.1 Product Perspective

AEROLOGIC is an independent product, conceptualized as a solution for a UAS. It is not a successor to, or a replacement for, any existing systems. As a crucial subcomponent of the larger UAS, AEROLOGIC acts as the central software hub, orchestrating various functionalities such as autonomous flight, navigation, obstacle avoidance, and precise package delivery.

AEROLOGIC integrates seamlessly with the UAS, serving as a middleware that facilitates critical interactions between the hardware components (like sensors, motors, and flight controllers) and external interfaces (such as the ground control station). This software ensures coordinated operations, enhancing overall efficiency and effectiveness of the UAS.



The accompanying sketch visually represents the role of AEROLOGIC within the broader UAS framework. It illustrates AEROLOGIC as the middleware that manages core avionic systems and vision processing, highlighting the importance of coordination with the UAS's hardware components. This high-level overview provides a map of how AEROLOGIC services as the central command unit that bridges the gap between hardware integration and the execution of software tasks.

2.2 Product Functions

The following are essential functionalities of AEROLOGIC, categorized primarily under core avionics and vision processing. These modules are the bedrock upon which AEROLOGIC operates, allowing for a blend of hardware interaction, data analysis, and mission execution. The core avionics module [1.5.2] ensures the UAS responds adeptly to dynamic flight conditions and mission objectives, while the vision processing module [1.5.3] empowers the UAS with environmental comprehension, critical for obstacle avoidance and pinpointing precise locations for package delivery. Each bullet point below encapsulates a function that contributes to the overall autonomy of the UAS.

2.2.1 Core Avionics Module

- Path planning and real-time navigation adjustments
- Mission state management and progression tracking
- Payload release timing calculations
- Sensor data integration for environmental awareness
- Fail-safe mechanisms for critical operation handling
- Energy management for optimized power consumption
- Communication protocols with the ground control station

2.2.2 Vision Processing Module

- Real-time image and video processing for object detection
- Machine learning algorithms for target identification
- Obstacle detection and avoidance strategies
- Georeferencing for accurate location pinpointing
- Payload drop zone recognition and verification

2.3 Operating Environment TBD

<Describe the environment in which the software will operate, including the hardware platform, operating system and versions, and any other software components or applications with which it must peacefully coexist.>

2.4 Design and Implementation Constraints

The following constraints listed are driven by the aim to create a compliant system within the scope of the SUAS 2024 competition.

- *Compliance with SUAS 2024 Rules*: All features must adhere to the competition regulations.
- *Hardware Limitations*: The UAS's limited processing power, memory, and particularly energy capacity necessitate efficient power management and optimization in software design.
- *Programming Standard*: Use of industry-standard programming languages and development environments to ensure ease of future development and maintenance.
- *Maintenance Standards*: Adherence to coding and documentation standard for future upkeep.

2.5 User Documentation TBD

<List the user documentation components (such as user manuals, on-line help, and tutorials) that will be delivered along with the software. Identify any known user documentation delivery formats or standards.> <GitHub readmes, this document, possible docusaurus site??>

2.6 Assumptions and Dependencies

- *Hardware Compatibility*: Assuming the software will be compatible with various UAS sensors used in the competition. <GO INTO MORE DETAIL WHICH SENSORS – TBD>
- *Third-Party Component Integration*: Dependence on third-party components or libraries stated with the software design document [1.5.5] with the assumption these will remain available and supported.

- *External Data Sources*: Dependence on external data sources (like GPS signals) assumed to be accurate and continuously available.

3. EXTERNAL INTERFACE REQUIREMENTS TBD

3.1 User Interfaces TBD

<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.> mission planner? Custom ground station software for images? Write a custom ground station software to combine the two? Why not?

3.2 Hardware Interfaces TBD

<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.>

3.3 Software Interfaces TBD

<Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial

components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.> this would be best to write after software is “complete”

3.4 Communications Interfaces TBD

<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.> <mavlink, serial ports, communication with ground station, rf??>

4. SYSTEM FEATURES

4.1 Automated Flight Plan Execution

Functions include automated takeoff, landing, adherence to predefined flight paths, and dynamic obstacle avoidance [4.2]. Priority: High

4.1.1 Functional Requirements

R 1.1 The system shall autonomously initiate takeoff when a mission is started.

R 1.2 The system shall navigate through mission flight path waypoints with a max error of 25ft only deviating during obstacle avoidance.

R 1.3 The system shall autonomously execute landing procedures at the designated landing zone.

4.2 Dynamic Obstacle Avoidance

Enables the UAS to autonomous identify and avoid obstacles during flight using onboard sensors, ensuring safety and operational reliability. Priority: High

4.2.1 Functional Requirements

R 2.1 The system shall detect obstacles within a specified range.

R 2.2 Upon obstacle detection, the system shall calculate an alternate path to avoid collision.

R 2.3 The system shall execute avoidance maneuvers while maintaining the overall mission path.

R 2.4 The system shall log all obstacle encounters and avoidance actions taken.

4.3 Real-Time Image Processing

Empowers the UAS to detect and interpret visual data in the drop zone for accurate payload delivery calculations. Priority: High

4.3.1 Functional Requirements

R 3.1 The system shall identify and classify standard objects [REF] and emergent objects [REF].

R 3.2 The system shall process images in real-time to determine precise drop locations.

R 3.3 Image analysis shall include object localization within a defined accuracy.

R 3.4 The system shall log image processing activities for post mission analysis.

4.4 Payload Management

Facilitates accurate and timely payload delivery based on the vision system and mission requirements. Priority: High

4.4.1 Functional Requirements

R 4.1 The system must process image data to identify and confirm the correct drop zone.

R 4.2 Payload release must be precisely timed to ensure delivery within 20ft from the target.

R 4.3 The system must account for UAS altitude and speed to calculate optimal payload release.

R 4.4 The system must log all payload delivery attempts for post-mission analysis.

4.5 Emergency Response Handling

Ensures the UAS can autonomously handle in-flight emergencies, complying with safety protocols outlined in section 4.11 in the 2024 SUAS competition rules. Priority: High

4.5.1 Functional Requirements

R 5.1 The system shall autonomously execute a return-to-home (RTH) or return-to-land (RTL) procedure upon loss of communication after 30 seconds.

R 5.2 In case of prolonged communication loss of 3 minutes, the system shall automatically initiate a flight termination process.

R 5.3 The emergency response system shall be operable without dependency on the internet.

R 5.4 The system shall be configured to return to GPS coordinates (38.315339, -76.548108) in the event of lost communication, serving as the designated RTH/RTL and flight termination point.

4.6 Telemetry and Communication

Facilitates robust data exchange between the UAS and ground control for effective mission management. Priority: Medium

4.6.1 Functional Requirements

R 6.1 The system shall maintain a continuous telemetry link to transmit real-time flight and image processing data to the ground station.

R 6.2 The system must provide a mechanism for manual override from the ground station in case of autonomous flight system failure.

R 6.3 The system shall allow manual override for target identification by operators of the ground station.

R 6.4 The system shall alert ground control operators of critical incidents or status changes.

4.7 Maintenance and Diagnostic Tools

Provides software utilities for routine system checks, troubleshooting, and post-mission data evaluation. Priority: Medium

4.7.1 Functional Requirements

R 7.1 The system shall include diagnostic tools for pre-flight checks and real-time monitoring.

R 7.2 The system shall have automated error detection and reporting for prompt maintenance actions.

4.8 Customizable Alerts

Enables operators to set up alerts for various flight conditions without needing to modify the software code. Priority: Low

4.8.1 Functional Requirements

R 8.1 The system shall provide a configuration panel where operators can set up custom alerts for specific flight conditions such as low batter, loss of signal, or deviations from flight paths.

5. OTHER REQUIREMENTS

AEROLOGIC will prioritize maintainability and testability to ensure consistent development throughout future missions.

Glossary

SRS – Software Requirements Specification

AEROLOGIC – Aerial Routing and Operational Logistics Integration Code

SUAS – Student Unmanned Aerial Systems

UAS – Unmanned Aerial System

GPS – Global Positioning System

RTH – Return to Home

RTL – Return to Land