

Software for Luna

Luna Project Team

October 2025

Abstract

This document details the complete software architecture for the Luna project, which is divided into two core components. The first is a suite of high-fidelity physical models designed for ground-based simulation and analysis on powerful computers. These models will allow for exhaustive computational experiments to predict Luna's behavior and refine its design. The second component is the lightweight, robust safety system designed to run on Luna's onboard hardware. This system uses a set of critical parameters derived from the ground-based models to ensure autonomous, real-time emergency response, independent of any external connection.

A Two-Tiered Software Framework

The software strategy for Luna is built on a two-tiered architecture that separates computationally intensive analysis from instantaneous, onboard safety logic. This approach ensures both maximum predictive accuracy during development and maximum reliability during operation.

- **Ground-Based Simulation & Analysis:** The comprehensive physical models, which form the first chapter of this report, are designed to run on high-performance machines. Their purpose is to create a digital twin of Luna, allowing us to conduct virtual experiments that would be costly or impractical to perform physically. We can simulate fluid dynamics, thermal transfer, and optical properties to understand everything there is to know about Luna’s behavior. The key output of these simulations is a refined set of simplified “critical parameters” that define the boundaries of safe operation.
- **Onboard Safety System:** The second chapter details the lightweight and robust program to be implemented on Luna’s onboard hardware (e.g., an Arduino). This local program will *not* run the full, complex models. Instead, it will continuously monitor live sensor data against the pre-defined critical parameters. This architecture ensures that Luna can autonomously identify an emergency scenario and execute the appropriate safety protocols instantly, without depending on a constant connection to the ground station.

Methodology and Realism

A core principle of this work is to ground our software in physical laws. Each model in the first chapter begins with an analysis of the relevant physics, which is then translated into a computational model. These models are designed to be as realistic as possible, incorporating variables that describe Luna’s internal structure and material properties, and even accounting for the known error margins of the onboard sensors. While we will conduct physical experiments to validate our findings, the goal is to have a simulation environment so precise that it can reliably predict the results.

Contents

- 1 Physical Models for Ground-Based Simulation 3**
 - 1.1 External Fluid Dynamics Model 3
 - 1.2 Internal Fluid Dynamics Model 3
 - 1.3 Optical Model 3
 - 1.4 Thermodynamic Model 3
 - 1.5 Barometric Pressure Model 3
- 2 Onboard Software & Safety Systems 4**
 - 2.1 Emergency Case Definitions 4
 - 2.2 Sensor Discrepancy Protocol 4
 - 2.3 Communication Protocol 4
 - 2.4 Hardware Specifications 4

Chapter 1

Physical Models for Ground-Based Simulation

The models in this chapter form the analytical core of the Luna project. They are designed to be implemented in Python, leveraging scientific libraries to achieve the highest possible fidelity. By running these complex simulations on ground-based computers, we can thoroughly explore Luna's performance envelope and derive the critical parameters needed for its onboard safety system.

1.1 External Fluid Dynamics Model

1.2 Internal Fluid Dynamics Model

1.3 Optical Model

1.4 Thermodynamic Model

1.5 Barometric Pressure Model

Chapter 2

Onboard Software & Safety Systems

This chapter details the architecture of the lightweight software that will reside on Luna's onboard computer. This system is designed for reliability and speed, focused exclusively on monitoring sensor data and executing safety protocols.

2.1 Emergency Case Definitions

2.2 Sensor Discrepancy Protocol

2.3 Communication Protocol

2.4 Hardware Specifications