Global Navigation Satellite System (GNSS) Resilience
System Design Document
Version <2.0>
03/05/2025

# Document Control

## Distribution List

The following list of people will receive a copy of this document every time a new version of this document becomes available:

Teaching assistants:

Pate, Clay

Customer(s): Clifford, Jayson F
Neighbors, Jake C.

Project team members:

Arena, Cristina M.
Cappers, Haskell A.
Legro, Jonathan
Lofton, Tyler R.

## Change Summary

The following table details changes made between versions of this document:

| Version | Date | Modifier | Description |
|---|---|---|---|
| 1.0 | 10/24/2024 | Jonathan Legro | Initial Document Creation |
| 1.1 | 11/21/2024 | Tytler Lofton | Added Figures 1 & 2 |
| 1.2 | 11/21/2024 | Tytler Lofton | Updated Sections 3.1. |
| 2.0 | 02/02/2025 | Jonathan Legro | General updates and refinements |
| 2.1 | 02/02/2025 | Jonathan Legro | Revised 1.1 Purpose and Scope for clarity |
| 2.2 | 02/04/2025 | Jonathan Legro | Updated 1.2.1 System Overview |
| 2.3 | 02/04/2025 | Jonathan Legro | Improved 1.2.2 Design Constraints hardware and display details |
| 2.4 | 02/04/2025 | Jonathan Legro | Refined 1.2.3 Future Contingencies to address potential MSFS 2020 update issues |
| 2.5 | 02/13/2025 | Jonathan Legro | Added interface change considerations in 1.2.3 Future Contingencies |
| 2.6 | 02/13/2025 | Jonathan Legro | Included feature expansion possibilities in 1.2.3 Future Contingencies |
| 2.7 | 02/13/2025 | Jonathan Legro | Noted potential performance optimizations in 1.2.3 Future Contingencies |
| 2.8 | 02/13/2025 | Jonathan Legro | Updated Figure 1 System Context Diagram in 1.2.1 System Overview |
| 2.9 | 02/20/2025 | Jonathan Legro | Updated Figure 2 Plugin Output in 3.2 Outputs |
| 2.10 | 02/20/2025 | Jonathan Legro | Reformatted and updated 3.1 Inputs to align with 3.2 Outputs |

| | | | |
|---|---|---|---|
| 2.11 | 02/20/2025 | Jonathan Legro | Added Purpose of Inputs and Access Restrictions to 3.1 Inputs |
| 2.12 | 03/06/2025 | Cristina Arena | Added 5.1 Hardware Interface Architecture, noting the system is currently software-only. |
| 2.13 | 03/06/2025 | Cristina Arena | Added 5.2 Hardware Interface Detailed Design, outlining potential future hardware integration. |
| 2.14 | 03/06/2025 | Tytler Lofton | Created 2.2 System Software Architecture diagram to illustrate software components. |
| 2.15 | 03/06/2025 | Tytler Lofton | Added a description explaining the 2.2 System Software Architecture diagram. |
| 2.16 | 03/06/2025 | Jonathan Legro | Updated 1.3 Document Organization for clarity and alignment with the document structure. |
| 2.17 | 03/06/2025 | Jonathan Legro | Updated 1.5.3 Abbreviations to include missing terms for consistency. |

# Table of Contents

# 1. Introduction

## 1.1. Purpose and Scope

The System Design Document (SDD) outlines the architecture and implementation of the GNSS Resilience - GPS Display Plugin for Microsoft Flight Simulator 2020 (MSFS 2020). This document details the system's design, constraints, and integration within MSFS 2020, focusing on software development using TypeScript and HTML. The plugin simulates GPS navigation but does not modify core simulator functionality or incorporate real-world GPS data. Instead, it uses predefined data to improve flight simulation while establishing a foundation for future features like geo-spoofing simulations. The system is designed primarily for pilots and researchers, enabling navigation training, aviation vulnerability testing, and GPS resilience studies.

## 1.2. Project Executive Summary

### 1.2.1. System Overview

The GNSS Resilience - GPS Display Plugin is a GPS display for Microsoft Flight Simulator 2020 (MSFS 2020). It shows predefined GPS data but does not support real-time navigation or route planning. The plugin integrates into MSFS 2020's instrument panel and is built using TypeScript and HTML.

The system does not modify MSFS 2020's core functionality or use real-world GPS data. It provides predefined navigation data and lays the groundwork for future features, such as geo-spoofing simulations.

Primary users include:

- Pilots and aviation enthusiasts for navigation training.
- Researchers for aviation testing and GNSS resilience studies.

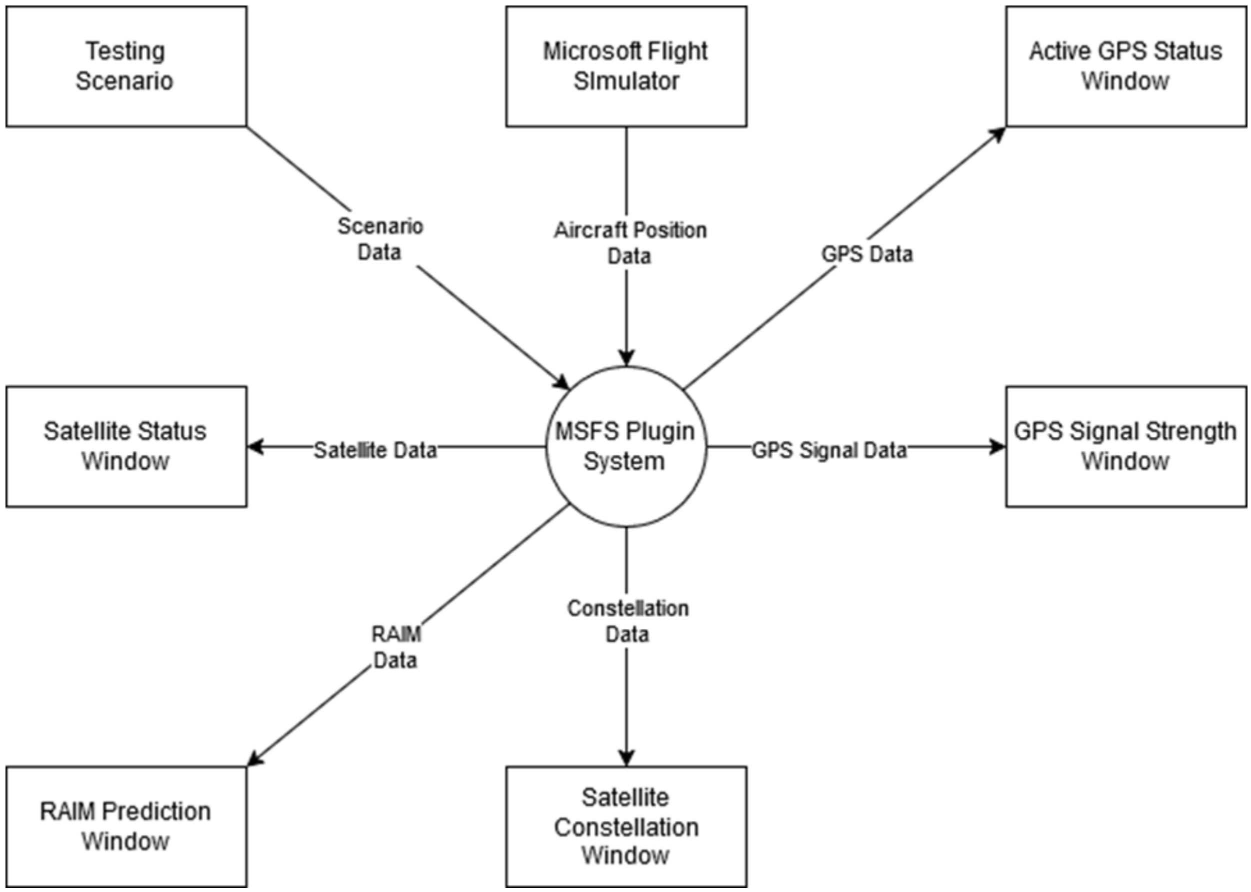


*Figure 1 System Context Diagram*

1.2.2. Design Constraints

- Platform: Runs on Windows 10 or higher, following MSFS 2020 system requirements.
- Hardware: Must meet MSFS 2020's minimum specifications, including 8GB RAM, a compatible GPU, and sufficient processing power.
- Display: Supports 800x480 (16:9) and 1024x768 (4:3) resolutions for cockpit integration.
- Software: Developed using TypeScript and HTML, ensuring compatibility with the MSFS 2020 SDK.
- Internet Requirement: No external communication is needed; however, MSFS 2020 may require an active internet connection for updates.
- Assumptions: Requires MSFS 2020 to be installed and properly configured for the plugin to function.

1.2.3. Future Contingencies

Potential changes may include:

- MSFS 2020 Updates: Compatibility issues may arise with future MSFS 2020 updates, requiring adjustments to maintain functionality.
- Interface Changes: If new display or interaction interfaces are introduced, modifications may be needed for seamless integration.
- Feature Expansions: Future updates may include geo-spoofing simulations or enhanced GNSS resilience testing.
- Performance Optimizations: Improvements may be made to enhance rendering efficiency and reduce system impact.

These factors may influence design updates or adjustments to ensure plugin functionality.

### 1.3. Document Organization

The System Design Document (SDD) includes:

- Introduction: Defines the purpose, scope, and key references.
- System Architecture: Describes hardware, software, and communication structures.
- Human-Machine Interface: Details system inputs and outputs.
- Detailed Design: Covers hardware, software, and internal communication design.
- External Interfaces: Explains hardware and software connections.
- System Integrity Controls: Addresses system reliability and future considerations.

This structure provides a clear and organized view of the system's design.

### 1.4. Project References

1) Microsoft Flight Simulator

- Title: *Microsoft Flight Simulator*
- Author: Wikipedia Contributors
- Source: Wikipedia
- Date: Accessed 2024
- Location: [Microsoft Flight Simulator Wiki]
- *Description*: Provides general information on Microsoft Flight Simulator, relevant for understanding the simulator's framework and user environment.

2) TypeScript Documentation

- Title: *TypeScript: JavaScript With Syntax for Types*
- Author: Microsoft
- Version: Latest

- Date: Accessed 2024
- Location: [TypeScript is JavaScript with syntax for types.]
- *Description*: The official TypeScript documentation, useful for developers implementing the plugin within MSFS 2020.

3) Federal Aviation Administration (FAA) - GNSS and GPS

- Title: *Global Navigation Satellite System (GNSS) and GPS*
- Author: Federal Aviation Administration (FAA)
- Date: Accessed 2024
- Location: Satellite Navigation - Global Positioning System (GPS)
- *Description*: An FAA resource detailing GNSS and GPS functionality, supporting the project's relevance to aviation resilience.

4) Advanced Navigation - GNSS and Satellite Navigation Explained

- Title: *Global Navigation Satellite System (GNSS) and Satellite Navigation Explained*
- Author: Advanced Navigation
- Date: Accessed 2024
- Location: Global Navigation Satellite System (GNSS) and Satellite Navigation Explained
- *Description*: An article on GNSS concepts, providing background information on satellite navigation relevant to GPS functionality in the plugin.

5) Microsoft Flight Simulator SDK Documentation

- Title: *Microsoft Flight Simulator SDK Documentation*
- Author: Microsoft
- Date: Accessed 2024
- Location: SDK Documentation
- *Description*: The official SDK documentation for MSFS 2020, essential for understanding development standards and integration guidelines.

6) Microsoft Flight Simulator FAQ

- Title: *Minimum, Recommended, and Ideal PC requirements for Microsoft Flight Simulator*
- Author: Microsoft
- Date: April 23, 2020
- Location: General & Announcements
- *Description*: The official SDK Support Update for MSFS 2020, essential for understanding hardware constraints.

7) Microsoft Flight Simulator Developer Support

- Title: *Official Developer Support Platform for MSFS 2020*
- Author: Microsoft
- Date: Ongoing
- Location: [MSFS DevSupport]
- *Description*: Comprehensive platform offering SDK documentation, best practices, community forums, and troubleshooting resources for developing add-ons and features for Microsoft Flight Simulator 2020.

## 1.5. Definitions, Acronyms, and Abbreviations

### 1.5.1. Definitions

This section lists terms used in this document and their associated definitions.

Table 1: Terms

| Term | Definition |
|---|---|
| Spoofing | Mimicking signals to deceive a system, often used to mislead GPS navigation. |
| Display Plugin | Software component added to extend MSFS 2020 with GPS display functionality. |
| TypeScript | Typed JavaScript used for developing the GPS plugin in MSFS 2020. |
| Instrument Panel | Displays essential flight info in MSFS 2020, where the GPS plugin is integrated. |
| End User | Uses the GPS plugin in MSFS 2020 for navigation. |
| Researcher | Tests GNSS resilience using the plugin. |
| Simulator | Imitative representation of a process or system. |
| Actors | All users of the system. |
| Spoofing | Mimicking signals to deceive a system, often used to mislead GPS navigation. |

### 1.5.2. Acronyms

This section lists the acronyms used in this document and their associated definitions.

Table 2: Acronyms

| Term | Definition |
|---|---|
| SDD | System Design Document |
| SRS | System Requirements Specification |
| GNSS | Global Navigation Satellite System |
| MSFS | Microsoft Flight Simulator |
| GPS | Global Positioning System |
| FAA | Federal Aviation Administration |
| CARS | Center for Aerospace Resilient Systems |
| HTML | HyperText Markup Language |

### 1.5.3. Abbreviations

This section lists the abbreviations used in this document and their associated definitions.

Table 3: Abbreviations

| Term | Definition |
|---|---|
| N/A | Not Applicable |

## 2. System Architecture

### 2.1. System Hardware Architecture

N/A

### 2.2. System Software Architecture

The system consists of a modular, dynamic design built using HTML, JavaScript, and TypeScript. The key components of software architecture are as follows:

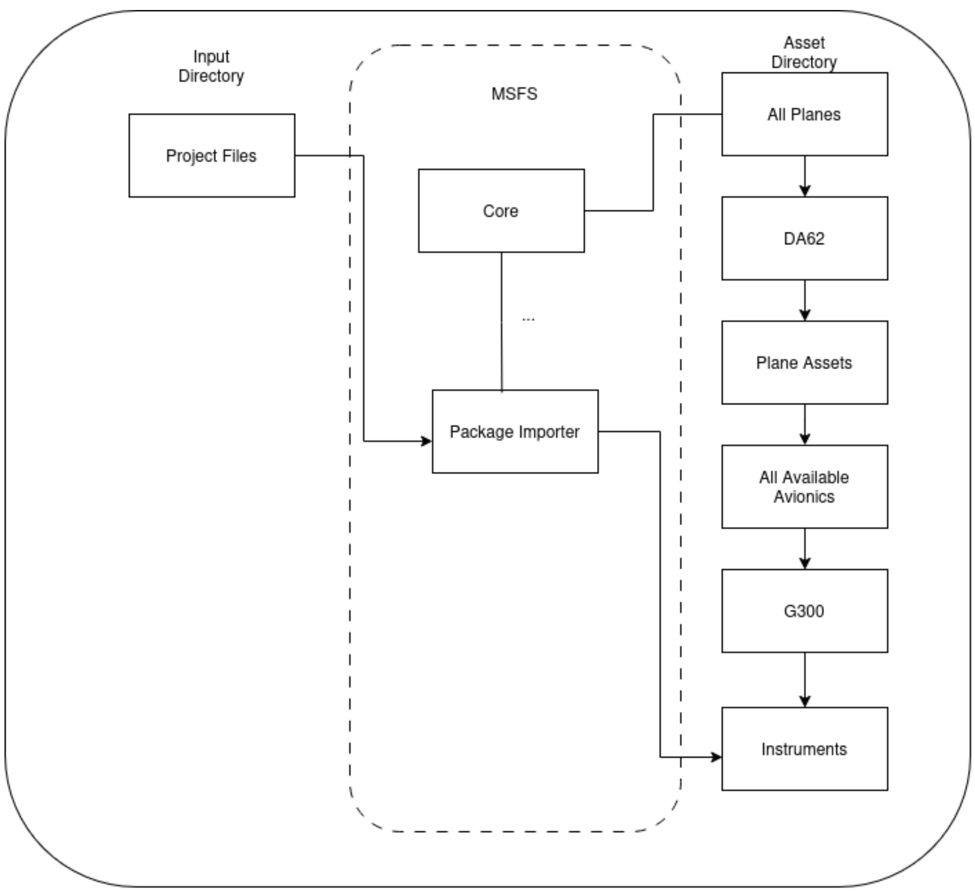


*Figure 2 System Software Architecture Diagram*

In the software architecture, the user will have all the project files in one location on their machine. In Microsoft Flight Simulator 2020, using the package importer module, they will select the directory containing the plugins files as the Input Directory. The user will then select the directory for the aircraft and then the avionics system that will be used. In this case we are using the Diamond DA62 as the aircraft and the Garming G3000 for the avionics system.

#### 2.2.1. Canvas Initialization Module

- Purpose: Sets up a single canvas element to replace the earlier multi-canvas approach, ensuring scalability and simplicity.
- Functions: Includes methods for initializing the canvas dimensions, obtaining context, and preparing it for graphical operations.

#### 2.2.2. Drawing Utilities Module

- Purpose: Provides reusable functions for drawing elements such as boxes, rings, bar graphs, and text.
- Functions:
  - drawBox(): Dynamically creates boxes with specific sizes and positions.
  - drawRing(): Generates satellite constellation rings.
  - addText(): Places labels and information dynamically on the canvas.

#### 2.2.3. Dynamic Resizing Module

- Purpose: Ensures all canvas elements adjust dynamically to varying monitor dimensions and resolutions, making the system future-proof.
- Functions: Calculates proportional sizes and positions based on the canvas size rather than hard-coded values.

#### 2.2.4. Graph Visualization Module

- Purpose: Renders bar graphs to display GPS signal strength and RAIM predictions.
- Functions:
  - drawGraph(): Plots dynamic data bars based on provided input percentages.
  - drawGrid(): Creates background grids for better data readability.

### 2.3. Internal Communications Architecture

The system employs modularized functions to facilitate seamless communication between components. Key communication flows include:

- Canvas Initialization to Drawing Utilities: Ensures the canvas is prepared before any graphical operations are performed.

- Drawing Utilities to Dynamic Resizing: Relays updated canvas dimensions for proportionate scaling of all visual elements.
- Dynamic Resizing to Graph Visualization: Provides calculated dimensions for the bar graph and grids.

# 3. Human-Machine Interface

## 3.1. Inputs

The system will display GPS navigation data within Microsoft Flight Simulator 2020 (MSFS 2020). The user will interact with the flight simulator as if using a real-world GPS display, but the system will not process real-time flight data or user input at this stage.

The system will use hardcoded navigation data to simulate GPS functionality, including:

- Ground Speed – Displays the aircraft's velocity relative to the ground.
- Longitude & Latitude – Tracks the aircraft's position on the world map.
- Altitude & Heading – Indicates current flight level and directional heading.

Purpose of Inputs:

- To provide GPS navigation data for a simulated flight experience.
- To support GNSS resilience research by presenting static satellite and signal conditions.

Access Restrictions:

- The system does not accept real-time user inputs or external control data.
- Only authorized developers can modify hardcoded navigation data for testing purposes.

## 3.2. Outputs

The system provides outputs to the user in the form of dynamic visualizations rendered on the canvas:

- GPS Status Display: Displays current GPS data, such as satellite status and RAIM predictions.
- Satellite Constellation Rings: Visual representation of satellite connections.
- Signal Strength Bar Graphs: Graphs showing relative signal strength, dynamically updated based on input data.

Purpose of Outputs:

- To present real-time GPS data and navigation-related information.
- To allow researchers to analyze GNSS resilience using visual aids.

Access Restrictions:

- The system allows anyone to view outputs.
- Only authorized users can modify settings or input critical data.

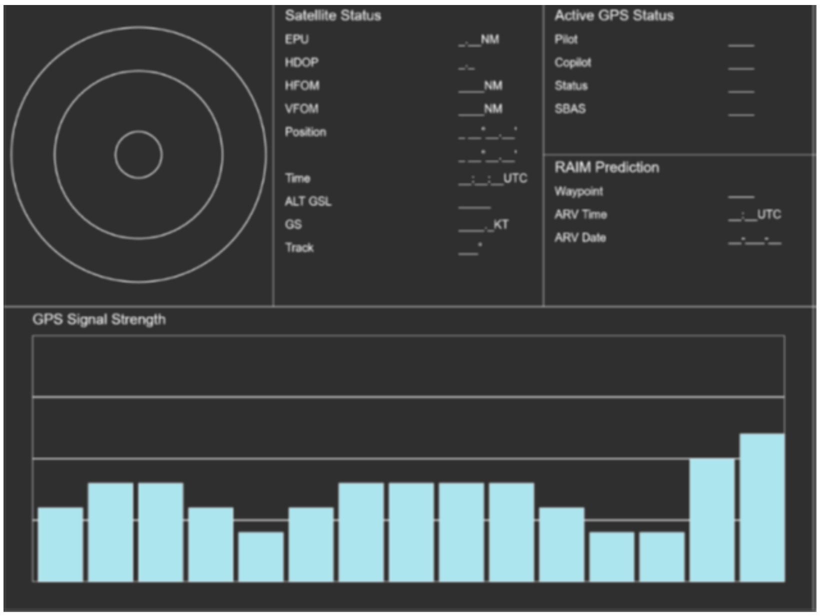


*Figure 3 GPS Plugin Display*

## 4. Detailed Design

### 4.1. Hardware Detailed Design

N/A

### 4.2. Software Detailed Design

The software design is modular, with each module performing specific roles. Below are the key modules and their descriptions:

#### 4.2.1. Canvas Initialization

- Purpose: Prepares the HTML canvas element for drawing.
- Details:
  - Initializes dimensions and context.
  - Ensures proper scaling for various screen sizes

#### 4.2.2. Drawing Utilities

- Purpose: Provides reusable functions for drawing elements on the canvas.
- Details:
  - drawBox: Creates boxes to represent windows or data areas.
  - drawRing: Generates satellite constellation rings.
  - addText: Adds labels or descriptions dynamically.

#### 4.2.3. Dynamic Resizing

- Purpose: Ensures all graphical elements scale proportionally with the canvas size.
- Details:
  - Calculates element dimensions relative to the canvas size.
  - Adjusts positions and sizes dynamically during runtime.

#### 4.2.4. Graph Visualization

- Purpose: Renders bar graphs and grids for visualizing GPS signal strength and RAIM predictions.
- Details:
  - drawGraph: Plots dynamic data bars.
  - drawGrid: Creates a structured grid for readability.

Code Reference: Specific implementations can be found in script.ts.

### 4.3. Internal Communications Detailed Design

The modules interact as follows:

1. Canvas Initialization:
   - Provides canvas dimensions and context to Drawing Utilities.
2. Dynamic Resizing:
   - Communicates updated dimensions to all drawing functions for proportional adjustments.
3. Graph Visualization:
   - Uses Drawing Utilities to render graph elements like bars and grids.

## 5. External Interfaces

### 5.1. Hardware Interface Architecture

The hardware components of the system interact with MSFS 2020 through a USB microcontroller, which serves as the interface between physical cockpit controls and the simulator software.

## 5.2. Hardware Interface Detailed Design

- Microcontroller Interface
  - USB microcontroller board to process inputs from physical components such as switches, encoders, and potentiometers. Communicates with MSFS 202 using SimConnect to register input actions and provide feedback to the user

- Landing Gear and Flaps Interface
  - Mechanical controls for landing gear, flaps, and parking break switch, which uses rotary encoders, limit switches, and potentiometers to send positional data to MSFS 2020.

- Control Panel and Switches Interface
  - Dedicated control panel houses various cockpit functions, including labeled toggle switches and push buttons for essential aircraft controls such as battery and alternator controls, avionics master switch and rotary encoders.
- Indicator Lights and LED Feedback Interface
  - Warning and caution lights provide visual status updates based on real-time flight conditions received from MSFS 2020. Includes LED indicators, control logic, and power source.
- Performance and Input Handling
  - The hardware system is designed to ensure real-time synchronization of physical inputs with MSFS 2020, including input processing time, self-test function, and allocations for future expansions.

## 5.3. Software Interface Architecture

The system integrates with Microsoft Flight Simulator 2020 using the MSFS SDK. Key architectural details include:

- TypeScript/JavaScript: The plugin is developed in TypeScript and JavaScript, aligning with MSFS SDK requirements for custom instruments.
- Integration Points:
  - Canvas rendering for visualizations like graphs, rings, and GPS data.
  - Data processing and dynamic resizing, ensuring compatibility across different hardware configurations.

## 5.4. Software Interface Detailed Design

The system communicates with MSFS 2020 via the following detailed software interfaces:

- Canvas Element: Acts as the primary visual interface, dynamically rendering GPS data and visual elements.

- Drawing Utilities: Modular functions (drawBox, drawRing, drawGraph) generate elements that reflect the current GPS status.
- Dynamic Resizing: Ensures the visual interface adjusts to various screen sizes, making it adaptable to different monitor dimensions.

## 6. System Integrity Controls

Currently, no system integrity controls are implemented, as the GPS display plugin operates independently within Microsoft Flight Simulator 2020 (MSFS 2020) and does not process live or external data.

Potential future integrity controls may include:

- Error Handling Mechanisms – Ensuring the system detects and logs display failures.
- Data Validation Checks – Verifying correct rendering of GPS data.
- Access Controls – Restricting modifications to authorized developers only.

These measures may be introduced in future updates to enhance system reliability.