**Issue/Rev No: 01/00**

**Date of Release: 8 Feb 2025**

**Software Design Standard**

**of <LRU/System Name>**

**for**

**<Platform Name>**

**Template No.**

CEMILAC\_SYSGP\_SDD\_07

|  |
| --- |
|  |
| C:\Users\jitendra Kumar\AppData\Local\Packages\Microsoft.Windows.Photos_8wekyb3d8bbwe\TempState\ShareServiceTempFolder\logo.jpeg**<DESIGN AGENCY LOGO>** | **Document No.:** |  |
| **Issue No. :** | <00X> | **Issue Date :** | <DD/MM/YYYY> |
| **Copy No. :**  | 01 of N | **No. of Pages :**  | < Total No. of Pages > |
| **Document Classification :** | 🞎 Secret 🞎 Confidential 🞎 Restricted 🞎 Unrestricted  |
| **Title:** | **Project/System :** |
| **Software Design Standard****of****<LRU/SYSTEM Name>for <Platform name>** | **<System/Project Name>** |
| **LRU/System Part No.:**  |
| <No.> |
| **Software Criticality Level :** |
| DO-178C Level <A/ B/ C/ D> |
|  | **Name & Designation** | **Signature** |
| Prepared By | <Design Rep Name>, < Designation> |  |
| Reviewed By | <Project Leader Name>, <Designation><AWG/QA HOD Name>, <Designation> |  |
| Approved By | <RCMA Officer>, <Designation><RCMA Name> |  |
| **<Design Agency Name & Address>** |

**Disclaimer:**

This document is a guidance document. Applicable section / table rows may be considered. Any additional details may be added. Any not applicable section/ table rows may be deleted. The template is very general and vary with process to process followed by Development Agency. The document may be fine-tuned with the TAA for finalization.

**Distribution List**

|  |  |  |
| --- | --- | --- |
| **Copy No.** | **Designation of the Copy Holder** | **Organisation** |
| 01 | Head of Design Agency | Design Agency Name |
| 02 |  |  |
| 03 |  |  |
| 04 |  |  |

**Amendment History**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Issue,****Rev. No.** | **Issue, Rev Date** | **Change Request Ref.** | **Brief Description of Amendment** | **Affected****Pages/ Section** | **Changed By** | **Change Effective****Date** |
| 01, 00 |  | NA | Initial Issue | NA | NA | Initial |
|  |  |  |  |  |  |  |

**Table of Contents**

[1. Introduction 6](#_Toc184112932)

[1.1 Purpose and Scope 6](#_Toc184112933)

[1.2 Applicable Documents 6](#_Toc184112934)

[1.2.1 External Documents 6](#_Toc184112935)

[1.2.2 Internal Documents 6](#_Toc184112936)

[1.3 Part Number and Nomenclature 6](#_Toc184112937)

[1.4 Acronyms and Abbreviations 6](#_Toc184112938)

[2. Design Methods 7](#_Toc184112939)

[3. Naming Convections 7](#_Toc184112940)

[4. Conditions for Design Methods 7](#_Toc184112941)

[4.1 Module Design 7](#_Toc184112942)

[4.2 Interface Design 8](#_Toc184112943)

[4.3 Data Design 8](#_Toc184112944)

[4.4 Control Design 8](#_Toc184112945)

[4.5 Error Handling 8](#_Toc184112946)

[4.6 Resource Management 8](#_Toc184112947)

[4.7 Timing and Scheduling 8](#_Toc184112948)

[4.8 Interrupts and others 8](#_Toc184112949)

[5. Tools for Design 8](#_Toc184112950)

[5.1 Tools to be used 9](#_Toc184112951)

[5.2 Constraints on Design Tools 9](#_Toc184112952)

[6 Constraints on Design 9](#_Toc184112953)

[7 Complexity Restrictions 9](#_Toc184112954)

# Introduction

Software Design Standards defines the methods, rules, and tools to be used to develop the software architecture and low-level requirements

# Purpose and Scope

This document is used to provide guidelines to develop low-level requirements and software architecture for <LRU\_NAME>. This is one of the major lifecycle data to be generated during the planning phase of the project in accordance with Section 4.5 and 11.10 of RTCA/DO-178C.

This document describes design description methods, naming conventions, design elements & complexity restrictions.

#  Applicable Documents

Define the list of all applicable documents in following sections:

## External Documents

Define the list of all applicable documents of external origin, relevant for this project with Document Name, Document No., Issue and Date.

## Internal Documents

Define the list of all applicable documents of internal origin, relevant for this project with Document Name, Document No., Issue and Date.

##  Part Number and Nomenclature

Define the details of all software components having unique part number and nomenclature to identify them through the software development life cycle.

#  Acronyms and Abbreviations

Define all the abbreviations and acronyms are listed with their expanded names.

# Design Methods

Define the methods and techniques to be used to describe the design of software design.

Software is driven from functional requirements, with low-level requirements derived from high-level requirements. Each low-level requirement is then mapped to a design. The following approach is used:

* Each high-level requirement is considered as a use-case, and broken up into low-level requirements comprising of interface (or boundary) functions, process functions and data.
* Interface functions (primarily input and output functions) are generalized and specified for device-level access functions. The calling convention and the return type are formalized.
* Process functions are defined with greater detail on how, when and under what conditions they are invoked, which other functions (generally interface functions or lower-level functions) are called and what type of data is accessed or updated.
* Software architecture is finalized to provide a robust framework to implement the design.

# Naming Convections

Define all the Naming Convections to be used to explain design details that enhance code readability, maintainability, such as data flow diagrams and formal specification languages.

Example All low level requirements will be named as LLRX.X, and Derived low level requirement will be named as DLLRX.X, where LLR stands for low-level requirements, DLLR stands for derived low-level requirements and X.X stands for the numbered hierarchy.

Provide Unique Id to each low level requirement so that it may be traced throughout the software life cycle.

# Conditions for Design Methods

Define all rules, conditions that are to be used in design methods as

## Module Design

Each module should have a detailed design specification, including its purpose, functionality, interfaces, data structures, and algorithms. The design should ensure module independence and ease of testing, focusing on the unique requirements.

## Interface Design

Interface specifications should include details on how modules interact, including data formats, protocols, and error handling mechanisms. Interfaces should be designed to be as simple and robust as possible, ensuring reliable communication between hardware and software components of the system.

## Data Design

Data design should ensure that data structures are well defined and optimized for performance and memory usage. Data integrity and consistency must be maintained across the system, particularly in managing antenna positioning data and control signals.

## Control Design

Control design should outline the control flow within and between modules. It should ensure that the system behaves correctly under all operational conditions, given in system specification.

## Error Handling

The design must include robust error handling mechanisms to detect, report, and recover from errors. Error handling should be consistent and cover all possible error conditions, especially those that could impact performance and safety of system.

## Resource Management

Resource management should ensure efficient use of system resources such as memory, processing power, and I/O. The design should avoid resource leaks and ensure timely release of resources, considering the real-time constraints of the software.

## Timing and Scheduling

The design should address timing and scheduling requirements, ensuring that all tasks meet their deadlines. Real-time constraints specific to antenna control operations should be clearly specified and adhered to.

## Interrupts and others

The design should address all requirements like use of interrupts, event-driven architectures, dynamic tasking, re-entry, global data, and exception handling by ensuring that all tasks meet as per time scheduling and priority.

# Tools for Design

Tools are used in design process, to maintain the consistency, compatibility and efficiency thought the design process. Define tools to be used for developing SDD document.

#  Tools to be used

Define list of tool(s) to be used for development of SDD document and UML diagrams.

#  Constraints on Design Tools

Define any limitation(s) or any constraint(s) on the use of the tools used.

# Constraints on Design

Define any constraint(s) on design. For Example, exclusion of recursion, dynamic objects, data aliases, and compacted expressions.

# Complexity Restrictions

Define the guidelines to manage and limit the complexity restriction in software design to ensure that it remains understandable, implementable, maintainable, testable etc. For Example maximum level of nested calls or conditional structures, use of unconditional branches, and number of entry/exit points of code components.