|  |
| --- |
| Simon Crowther |
| SSRD |
| Contains Information about high level software requirements of the system being created |

# Contents

[Contents 1](#_Toc499581397)

[Change History 2](#_Toc499581398)

[Referenced Documents 2](#_Toc499581399)

[1 Introduction 3](#_Toc499581400)

[1.1 Purpose 3](#_Toc499581401)

[2 Main Systems\_Checks 3](#_Toc499581402)

# Change History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Issue | Author | Date | Comment | Authorised |
| 1 | S.C | 10/10/2017 | Initial Version | Yes |

# Referenced Documents

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Issue | Author | Date | Comment | Authorised |
| N/A |  |  |  |  |

# Abbreviations

|  |  |
| --- | --- |
| Abbreviation | Detail |
| FPT | Flight Phase Test |
| IPT | Initialised Phase Test |
| LPT | Launch Phase Test |
| SPT | Simulation Phase Test |
|  |  |
|  |  |

# 1 Introduction

## 1.1 Purpose

This Sub-system provides the software requirements for the validation and selection of the Main Missile System Checks parameters for use in Maintenance of the system. This document describes the overall architecture of the software and provided information on how that architecture has links to the Ada implementation. Traceability will be been provided for the generated code packages and operations.

## 1.2 Scope

This Software Requirement Specification captures the system design for the ADLAS. The System Checks software will integrate with the flight controller software. As such, it is required to interface with certain customer provided packages. The design identifies these packages and identifies the parts of those packages used, but does not describe them in detail as their design is not under the wing of this development arm.

## 1.3 Overview

The remainder of this document briefly gives an overview of the system to which the ADLASs function relates, and an overall description of how it works followed by the detailed requirements to be implemented.

# 2 Software Design

## 2.1 System Overview

The system intends to perform Common, IPT, SPT, LPT and FPT on the ADLAS Computer. The hardware provides a number of registers containing information about the system and the status of these registers is checked to determine the status of the system.

The tests are generally performed by reading the required bits of a number of hardware registers. Some registers react to values contained in other registers and so to test them it is necessary to write to a given register and then read the associated register to check its value has changed as expected.

### 2.1.1 Main System Checks

Common tests run as an uninterrupted sequence of tests, although prior to executing of the tests, a copy of the hardware registers is created and the test is performed on the register copies.

### 2.1.2 IPT

There is a single IPT test consisting of a number of tests and which is run when instructed by the operator. Again, IPT runs as a sequence of uninterrupted tests. Prior to executing the tests, a copy of the hardware registers is made and the test performed on the register copies.

### 2.1.3 SPT

There is a single SPT that consists of many simulator tests.

## 2.2 Design Approach

The approach used to represent the design will be based on Booch Object Oriented Design whereby the design is split into a number of high level ‘objects’ which can be subsequently decomposed individually into one or more further objects or one or more Ada packages with minimal reference to one another. The ‘objects’ will not be implemented but are used as a device to encapsulate the design. The overall layout of the objects is as follows:

ADLAS Schedulers

Static\_Initialise

IPT\_Scheduler

SPT\_Scheduler

LPT\_Scheduler

FPT\_Scheduler

ADLAS\_Checks

{set of ADLAS Tests Ops}

Failures

{set of Failure Ops}

Simulation

Simulation\_Coordinates

External\_Packages

{Temperatute\_Monitor ops}

Guidance

Registers

Read\_Register

Write Register

I/O\_Packages

Get\_Signal\_Value

Get\_Register\_Value

Get\_Signal\_Register\_Value

Put\_Signal\_Value

Static\_Initialise

Read\_And\_Buffer\_Input\_Signals

GPIO\_Read

GPIO\_Write

## 2.3 Object Descriptions

### 2.3.1 ADLAS Schedulers

This object provides the BIT controller packages. There will be a scheduler package for each of Phase Test Packages. The Scheduler packages will arrange the timing and execution of the procedures located in the phase packages.

### 2.3.2 ADLAS Checks

This object provides the tests that run on the algorithm set by the higher level procedures that are called by the Scheduler, the phases will contain tests that are also common to other phases.

### 2.3.3 External\_Packages

This object contains many external packages that have been included from previous ADLAS designs.

### 2.3.4 I/O\_Packages

This object provides types, data and procedures to allow ADLAS\_Tests to process the data and I/O on the processor. This package will create copies of the registers then also store them before starting through the phases.

### 2.3.5 Failures

This object provides types, data and procedures to record and monitor failures that occur on the ADLAS system.

### 2.3.6 Registers

This object provides an interface to the hardware registers and allows the register values to be read or written.

### 2.3.7 Simulation Package

This object provides an interface to the simulation procedures located from the simulation computer. Allows data values and precise points to be read back.

# 3 Object Bit\_Tests

This Object is further decomposed into objects due its size.

External\_Packages

I/O\_Packages

Failures



Common\_Checks

{Main\_System\_Checks}

## Objects

### IPT

This object holds all the packages that contain all the procedures to check the initialisation phase tests.

### SPT

This object holds all the packages that contain all the procedures to check the simulation phase tests.

### LPT

This object holds all the packages that contain all the procedures to check the launch phase tests.

### FPT

This object holds all the packages that contain all the procedures to check the flight phase tests.

### Common\_Checks

This object holds all the packages that contain all the procedures that are used by two or more phase test packages.