**Software validation specification (SVS)**

**Version 1.6**

**Solar Orbiter Mission**

**RPW INVESTIGATION**

**MEB (Main Electronic Box) Instrument**

**LFR (Low Frequency Receiver) Sub-Instrument**

|  |  |  |  |
| --- | --- | --- | --- |
| **Prepared by:** | **Function:** | **Signature:** | **Date** (Dd/mm/yyyy) |
| V. BOUZID(LPP)  B. KATRA (LPP) | V&V Engineer | signVB.gif  signature3.jpg | 04/05/2017 |
| **Verified by:** | **Function:** | **Signature:** | **Date** |
|  |  |  |  |
| **Approved by:** | **Function:** | **Signature:** | **Date** |
| W.RECART (NEXEYA SERVICES) | Software Quality Engineer |  |  |
| **For application:** | **Function:** | **Signature:** | **Date** |
|  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Iss** | **Rev** | **Date** | **Author** | **Notes** |
| 0 | 0 | 15-Feb-13 | G.SAULE | First Issue according to the ECSS‐E‐ST‐40C\_6March2009 document (**Annex L** (normative) Software validation specification (SVS) -  DRD \ **L.2** Expected response \ **L.2.1** Scope and content) |
| 0 | 1 | 11-Jun-13 | G.SAULE | Taking into account:   * The RPW-SYS-SSS-00013-LES\_Issue1\_rev2 document. * The relecture\_SVS.xlsx sheet. |
| 0 | 2 | 30-Jul-13 | G.SAULE | Minor evolutions |
| 1 | 0 | 15-Nov-13 | G.SAULE | Evolutions for TRR. |
| 1 | 1 | 10-Apr-14 | G.SAULE | Evolutions for V2. |
| 1 | 2 | 25-jun-14 | B.KATRA  V.BOUZID | Evolutions for TRR2 |
| 1 | 3 | 15-aug-14 | B.KATRA  V.BOUZID | Update/corrections from JIRA issues for CDR/TRB R2 :  RPWSWR-433 |
| 1 | 4 | 20-oct-15 | B.KATRA  V.BOUZID | paragraphs 5.3, 5.6,8.1.3 and 9 updated  RD07 added  RD08 added  All TestCases updated  SVS-0004 SVS-0015 SVS-0016 SVS-0017 SVS-0023 SVS-0026 SVS-0037 SVS-0060 SVS-0061 SVS-0072 deleted  SVS-0068 SVS-0082 SVS-0083 SVS-0084 SVS-0086 SVS-0087 SVS-0088 SVS-0089 SVS-0091 SVS-0092 SVS-0093 SVS-0094 SVS-0096 SVS-0097 added |
| 1 | 5 | 16-Feb-16 | B.KATRA  V.BOUZID | Minor corrections  Shift bug in SVS numbering fixed  Updated SW (3.0.0.10 to 3.0.0.22)  SVS-0003 updated (TC with length > MAX\_LEN)  SVS-0005 SVS-0007 SVS-0008 SVS-0011 SVS-0012 SVS-0019 SVS-0020 SVS-0027 SVS-0029 SVS-0031 SVS-0032 SVS-0034 SVS-0055 SVS-0056 SVS-0057 SVS-0058 SVS-0059 SVS-0067 SVS-0068 SVS-0069 SVS-0070 SVS-0073 SVS-0074 SVS-0076 SVS-0080 SVS-0083 SVS-0089 SVS-0095 updated  SVS-0088 updated to comply with ICD >=3.1  SVS-0026 SVS-0037 SVS-0060 SVS-0061 added  SVS-0006 SVS-0085 SVS-0096 SVS-0097 deleted |
| 1 | 6 | 02-Apr-17 | B.KATRA  V.BOUZID | Major updated and tests added for R3++ compliancy  Updated SW (3.0.0.22 to 3.2.0.24) and HW (1.1.89 to 1.1.91)  Merged requirements  Updated ( New TC and improvements)  SVS-0003 SVS-0005 SVS-0007 SVS-0008 SVS-0012 SVS-0018 SVS-0019 SVS-0020 SVS-0021 SVS-0022 SVS-0026 SVS-0027 SVS-0028 SVS-0029 SVS-0030 SVS-0031 SVS-0032 SVS-0034 SVS-0035 SVS-0037 SVS-0040 SVS-0041 SVS-0042 SVS-0043 SVS-0044 SVS-0045 SVS-0053 SVS-0054 SVS-0055 SVS-0056 SVS-0057 SVS-0059 SVS-0060 SVS-0062 SVS-0064 SVS-0069 SVS-0077 SVS-0078 SVS-0080 updated  SVS-0024 SVS-0025 SVS-0039 SVS-0046 SVS-0047 SVS-0047 SVS-0048 SVS-0049 SVS-0050 SVS-0051 SVS-0052 SVS-0058 SVS-0068 SVS -0075 SVS-0084 SVS-0087 SVS-0092 SVS-0093 SVS-0094 deleted  SVS-0096 SVS-1100 SVS-1110 SVS-1102 SVS-1103 SVS-1109 SVS-1201 SVS-1202 SVS-1203 SVS-1204 Added |
| 1 | 7 | 7-nov-18 | VBO | Minor Updated tests due to improve GCOV analyze  Updated SW (3.2.0.24 to 3.2.0.24) and HW ( 1.1.91 to 1.1.91)  updated SVS-0003 SVS-0007 SVS-0013 SVS-0018 SVS-0019 SVS-0026 SVS-0027 SVS-0034 SVS-0037 SVS-0041 SVS-0042 SVS60043 SVS-0044 SVS-0045 SVS-0053 SVS-0057 SVS-0061 SVS-0065 SVS-0069 SVS-0077 SVS-0079 SVS-0081  SVS-0058 Added  SVS-0068 SVS-0080 Deleted |

**Table of Contents**

[1 Introduction 6](#_Toc482103709)

[2 Applicable and reference documents 6](#_Toc482103710)

[2.1 Applicable documents 6](#_Toc482103711)

[2.2 Reference documents 6](#_Toc482103712)

[3 Terms, definitions and abbreviated terms 7](#_Toc482103713)

[4 Software overview 8](#_Toc482103714)

[5 Software validation specification task identification 8](#_Toc482103715)

[5.1 Task and criteria 8](#_Toc482103716)

[5.2 Features to be tested 8](#_Toc482103717)

[5.3 Features not to be tested 8](#_Toc482103718)

[5.4 Test pass - fail criteria 8](#_Toc482103719)

[5.5 Items that cannot be validated by test 8](#_Toc482103720)

[5.6 Manually and automatically generated code 9](#_Toc482103721)

[6 Software validation testing specification design 9](#_Toc482103722)

[6.1 General 9](#_Toc482103723)

[6.2 Organization of each identified test design 9](#_Toc482103724)

[7 Software validation test case specification 9](#_Toc482103725)

[7.1 General 9](#_Toc482103726)

[7.2 Organization of each identified test case 9](#_Toc482103727)

[8 Software validation test procedures 11](#_Toc482103728)

[8.1 General 11](#_Toc482103729)

[8.2 Organization of test procedures 13](#_Toc482103730)

[Validation test platform requirements 196](#_Toc482103731)

[9 Software validation specification additional information 203](#_Toc482103732)

**List of Figures**

This section has been left blank intentionally.

Aucune entrée de table d'illustration n'a été trouvée.

**List of Table**

This section has been left blank intentionally.

**Aucune entrée de table d'illustration n'a été trouvée.**

# Introduction

This document provides the validation tests procedures to be run according to the validation test strategy describe in AD01.

# Applicable and reference documents

## Applicable documents

|  |  |  |  |
| --- | --- | --- | --- |
|  | Reference | Ver. | Title |
|  | RPW-MEB-LFR-SDP-00040-1-1\_LFR\_Software\_Development\_Plan | 2.0 | Software Validation and Verification Plan (included in referenced Software Development Plan) |
|  | RPW-MEB-LFR-PLN-00033 |  | SPAP |
|  | RPW-SYS-SSS-00013-LES | Issue4 rev3 | RPW Instrument - Software System Specification |
|  | RPW-MEB-LFR-PLN-00035-LPP-0-0\_SValP | 1.1 | SOFTWARE VALIDATION PLAN (SVALP) - Version 0.0 - Solar Orbiter Mission - RPW INVESTIGATION - MEB (Main Electronic Box) Instrument - LFR (Low Frequency Receiver) Sub-Instrument |
|  | SOL.S.ASTR.TN.00079\_03 | 3 | Solar Orbiter TM-TC and Packet Structure ICD |
|  | SOL-EST-RCD-0050 | i3 | Solar Orbiter Experiment Interface Document – Part A (EID-A) |
|  | RPW-MEB-LFR-SRS-00020 | 2.1 | Software Requirements Specification (SRS) |
|  | RPW FDIR |  | RPW-SYS-MEB-###-FMC-000207-LES |
|  | ECSS-E-ST-50-53C | C | Space engineering / SpaceWire – CCSDS packet transfer protocol |
|  | Solar Orbiter Operations Requirements Document | I1r6 | SO-ESC-RS-05001 |
|  | RPW-SYS---ICD-00067-LES | Issue4 rev3 | RPW IDB PARAMETER DEFINITION |
|  | RPW-MEB-LFR-SDD-00039 | 1.3 | Software Design Document |

## Reference documents

|  |  |  |  |
| --- | --- | --- | --- |
|  | Reference | Ver. | Title |
|  | ECSS-Q-ST-80 | C | Space product assurance Software product assurance |
|  | ECSS-E-ST-40 | C | Space engineering - Software |
|  | ECSS‐E‐ST‐50‐12 | C | Space engineering - SpaceWire – Links, nodes, routers  and networks |
|  | RPW-MEB-LFR-DRP-00028 |  | Design report |
|  |  | Version 1.1.0 - B4104, November 2010 | GRLIB IP Core User’s Manual - |
|  | RPW-SYS-MEB-LFR-ICD-00097 | Issue4\_Rev3 | RPW\_ICD\_LFR\_TC\_TM |
|  | RPW-MEB-LFR-NTT-00124 | 1.1 | SGSE Software User Manual |
|  | RPW-MEB-LFR-NTT-00125 | 1.0 | LFR GSE : system clock simulator |

# Terms, definitions and abbreviated terms

| **Abbreviation** | **Mean** |
| --- | --- |
| AD | Applicable document |
| APID | Application Process Identifier |
| ASM | averaged spectral matrices |
| BIAS | Biasing Unit |
| bit | Binary unit |
| CCSDS | Consultative Committee for Space Data Systems |
| CRC | Cyclic Redundancy Code |
| CTR | Central Time Reference |
| CWF | continuous waveforms |
| DAS | DPU Application Software |
| DMS | Data Management System |
| dop | degree of polarization |
| DPS | DPU Flight Software |
| DPU | Data Processing Unit |
| DR | Reference documents |
| DRD | Document Requirements Definition |
| ECSS | European Cooperation on Space Standardization |
| EID-A | Experiment Interface Document part A |
| EM | Engineering Model |
| FDIR | Failure Detection Isolation and Recovery |
| fn | Function |
| FSW | Flight Software |
| HK | House Keeping |
| HMI | Human machine interface |
| HW | Hardware |
| i.e. | Id est |
| ICD | Interface Control Document |
| ID | Identifier |
| IDB | Instrument DataBase |
| LA | logical address |
| LED |  |
| LF | Low Frequency |
| LFR | Low Frequency Receiver |
| LSB | Least Significant Bit |
| max | Maximum |
| MEB | Main Electronics Box |
| ms | millisecond |
| MSB | Most Significant Bit |
| NA | Not Applicable |
| NOK | Not OK |
| NT | Not tested |
| PB | PlayBack |
| PE | spectral power of E field |
| PLN | Plan |
| POK | Partially Ok |
| PUS | Packet Utilization Service |
| RB | requirements baseline |
| RD | Reference Document |
| REQ | requirement |
| Rev | Revision |
| RMAP | remote memory access protocol |
| RPW | Radio and Plasma Wave analyzer |
| s | second |
| SBM | Selected Burst Mode |
| SCM | Search Coil Magnetometer |
| SGICD | Space to Ground Interface Control Document |
| SPAP | Software Product Assurance Plan |
| SpW | SpaceWire |
| SRS | Software Requirement(s) Specification |
| SSS | System Software Specification |
| SVALP | Software Validation Plan |
| SVS | Software Verification Specification |
| SW | Software |
| SWF | snapshot waveform |
| tab |  |
| TBC | To Be Confirmed |
| TC | TeleCommand |
| TDS | Time Domain Sampler |
| THR | TNR-HFR |
| TM | Telemetry |
| Topcased | Toolkit in Open-Source for Criticical Application & Systems Development |
| TS | Time Series |
| Ver | Version |
| Vol | volume |

# 

# Software overview

The purpose of the LFR FSW is to drive the appropriate LFR scientific modes upon TC received from the RPW DPU.

The LFR FSW main functions are the following:

* Communication with the RPW DPU: telecommands interpretation, telemetry products packaging and transmission to the DPU.
* Configuration of the LFR sampling strategy and pre-processing abilities following the reception of telecommands and of appropriate parameter sets coming from the RPW DPU.
* Time keeping.
* Computation of a given set of scientific products: the Basic Parameters.
* Housekeeping data collection and housekeeping reporting to the DPU.

The LFR FSW is made of one only product that implements all of the requirements related to the LFR instrument scientific modes as described in the AD03.

# Software validation specification task identification

## Task and criteria

Each SRS requirement tagged as “TEST” is related to a test case in this document (see §8). Each test case can be divided into several steps(§8.1.3) which represent a sub-case intended to validate a part of the requirement.

## Features to be tested

Each SRS requirement (derived from SSS requirements) that are tagged “TEST” should be covered by a SVS test case e.g. related to one or more tests validate it.

## Features not to be tested

SRS requirement (derived from SSS requirements) that are tagged “DESIGN” are not tested and not described in this document (see §6). Also, some requirements tagged “TEST” that have test case associated could not be tested and are tagged “NA” e.g. “Not Applicable” because they may be related to yet not implemented functions.

## Test pass - fail criteria

A test is considered as “PASSED” or “OK” if : all steps of test case or “OK” e.g. tests results are matching expected results.

A test is considered as “FAILED” or “NOK” e.g. “Not OK” if : all steps of test case are are “NOK” e.g. tests results do not match expected results.

A test is considered as “POK” e.g. “Partially OK” if : 1 or more steps are “NOK” or “POK”.

## Items that cannot be validated by test

SRS requirement (derived from SSS requirements) that are tagged “DESIGN” are not tested and not described in this document. All requirements that cannot be tested by test are at least tested by design. Some test case hardly linked to SpaceWire communication protocol are also noted “NT” e.g. “Not Tested” because of a lack of advanced interfaces and tools to interact with SpaceWire registers and input/output.

## Manually and automatically generated code

We internally developed our own framework (SocExplorer) to interact with LFR and be able to write and execute python scripts. Those script are used to implement test case described in this document (1 script = 1 step of a test case). Around 70% of tests used are ran and automatically analyzed by scripts to determine pass – fail criteria. Also, some tests have been implemented as rules that are always and automatically checked for all tests even if it’s not the purpose of the test (for example: periodicity of HK is always checked).

# Software validation testing specification design

Everything that is relative to test by “DESIGN” is detailed in AD12.

## General

This section has been left blank intentionally.

## Organization of each identified test design

This section has been left blank intentionally.

### General

This section has been left blank intentionally.

### Features to be tested

This section has been left blank intentionally.

### Approach refinements

This section has been left blank intentionally.

# Software validation test case specification

## General

Most of the elements of this chapter are described in §5 and §8 of this document.

## Organization of each identified test case

### Test case identifier

See §8.1.1.

### Inputs specification

This section has been left blank intentionally.

### Outputs specification

This section has been left blank intentionally.

### Test pass - fail criteria

See §5.4.

### Environmental needs

This section has been left blank intentionally.

### Special procedural constraints (ECSS-Q-ST-80 clause 6.3.5.25)

This section has been left blank intentionally.

### Interfaces dependencies

This section has been left blank intentionally.

# Software validation test procedures

## General

The organization of each identified test procedure is done according the template below:



Note some fields use a dedicated style to be treated by TOPCASED/TRAMWAY tool: those styles are: Assessment\_Topcased, Date\_Topcased, TestOperator\_Topcased, TestProcedureIdentifier\_Topcased, Step\_Topcased, Comments\_Topcased, ExpectedResult\_Topcased, Status\_Topcased (the field <XXX> is in <XXX>\_Topcased style).

### Test procedure identifier

It’s an identifier “SVS-<X>\_Ed<i>” with the “TestProcedureIdentifier\_Topcased” style.

<X> is an integer.

<i> is the release of the SRS req. The goal of this index is to trace the changes of substance.

### Purpose

This section has been left blank intentionally.

### Procedure steps

The steps are cut. They preferentially are status results oriented. So, the steps -by in anticipation of eventual fail- correspond to the verifiable independent characteristics. Example when one action is sufficient to check x items: x steps will get have all the combination of verdicts. The potential faulty steps will isolate parts to fail. Some steps may be tagged not applicable (NA) if the expected result is not relevant compared to other steps. The reason should be explained in the “Comments” column (see for example SVS-0008/Step 1).

### Test script

If possible, tests are automated. This is the case of repetitive or long tests. Python 2.7 is used.

The program sends integer sequences to simulate TCs. It stores the TCs and the integers received (TMs).

At the end of progress (date=xxx=yyyy\_mm\_dd-hh\_mm\_ss, for example 2013\_10\_15-23\_46\_55):

-Raw data files are generated: for example, xxx-Tc.csv and xxx-Tm.csv.

-All the TCs/TMs are automatically analyzed. All fields of all TCs/TMs are then extracted. This is the file xxx-Detail.txt. Note a synthetic file is too generated: xxx-Synth.txt.

All displays (logs, info, warnings, errors) obtained during the execution of the script are recorded in xxx-Nb.txt.

For some tests, using xxx-Tc.csv, and xxx-Tm.csv:

-if it is more ergonomically to have the relevant fields to reach a verdict: an xxx-Extract.txt is generated.

-if analysis is automated, it generates an xxx-Analyze.txt file.

This section has been left blank intentionally.

## Organization of test procedures

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE** | | | | | | | | |
| **Test procedure identifier** | | | **Date** | | **Assessment** | | | |
| SVS-\_Ed3 | | | DD/MM/YYYY | | NT / NA / NOK / POK / OK | | | |
| **TS Tested requirements** | | | **RB Tested requirements** | | | | | |
| N/A | | | REQ-LFR-SRS-5200\_Ed1 | | | | | |
| **Component:** | |  | **Version:** V1 | |  | | | |
| **Involved subsystems** | | | | | | | | |
| SW: 3.2.0.24 | |  | | | | | | |
| HW: 1.1.91 | | StarDundee | | | | | | |
| Start time: | | (DD/MM/YYYY) hh:mm | Test Operator: | | |  | | |
| End Time: | | (DD/MM/YYYY) hh:mm | Test Duration: | | |  | | |
| **Purpose** | | Check the LFR FSW processes the command packets transmitted by the DPU. | | | | | | |
|
| **General remarks about the test** | | | | | | | | |
| Requirement Title | | Command management. | | | | | | |
| dependencies | | SSS-CP-FS-020 | | | | | | |
| restrictions | | -DBS –used only in SAFE mode- does not communicate with LFR.  -There’s no Check memory feature for the LFR.  -There’s no self tests feature for the LFR (no req): see §5.5.1 (Requirements\Operational requirements\LFR mode management) of SRS.  -There’s no diagnostic feature for the LFR. | | | | | | |
| means | |  | | | | | | |
| input data | |  | | | | | | |
| prerequisite | |  | | | | | | |
| used test programs | |  | | | | | | |
| test script | | grspw\_registers.py commonIntensiveLoop.py, actionsTcLfr\*.py, | | | | | | |
| output data | |  | | | | | | |
| Notes | | This requirement is a very high level: its complete verification includes many TEST CASES.  Socexplorer can be used to check GRPSPW registers. | | | | | | |
| **Procedure steps** | | | | | | | | |
| **Step** | **Actions** | | | **Expected Results** | | | **Comments** | **Status** |
|  | Spw board is powered on. The FSW is not launched.  Read in the board register at 0x80000500+APB address offset:  the destination key for the RMAP protocol  the Node address.   * See in RD05 document, at the paragraph §"GRSPW - SpaceWire codec with AHB host Interface and GRSPW target"/"Registers" Currently, Destination key is 0x10. * The RPWSSS-202 Jira issue of the RPW-SSS project is "RMAP key values". Currently, "The analyzer FPGA and S/W shall be configured with the following RMAP key: LFR RMAP key = 2". * SY\_LFR\_DPU\_LA is defined in IBD. Currently SY\_LFR\_DPU\_LA=0xFE.   The grspw\_registers() script realize this verification, and produce a log file \*Synth.txt | | | Verify Node address  =SY\_LFR\_DPU\_LA,  and  Verify Destination key  = LFR RMAP key.  Verify the test result is success (“VERDICT = PASS”). See snapshots below. | | | Use Also Socexplorer |  |
|  | FSW LFR is loaded, and is launched.  Launch the /lfrverif/LFR\_SVS/SVS-0001 /Step03/actionsTcLfr\*.py scripts: send successively each TC with nominal values (PACKET\_LENGTH, CRC correct …).  For TC\_LFR\_LOAD\_NORMAL\_PAR, enter in STANDBY mode, else pass in NORMAL mode.  For the free parameters of each TC:  If possible, take the default values, otherwise set all allowable ranges for each discrete parameter.  Notice:  -Sends comply with timing constraints.  -The TC-chaining is not representative of use.  -The SEQUENCE\_CNT, CUC are not affected in a relevant way. | | | Verify:   * the FSW stays healthy. * the TM\_LFR\_TC\_EXE\_SUCCESS transmission. | | |  |  |
|  | FSW LFR is loaded, and is launched.  Successively, for each TC: repeatedly send it with nominal values (PACKET\_LENGTH, CRC correct …).  -For TC\_LFR\_LOAD\_NORMAL\_PAR, enter in STANDBY mode, else enter in NORMAL mode.  -Sends comply with timing constraints.  -The TC-chaining is not representative of use.  -The SEQUENCE\_CNT, CUC are not affected in a relevant way. | | | Verify the FSW stays healthy. | | |  |  |
|  | End | | |  | | |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE** | | | | | | | | | | | |
| **Test procedure identifier** | | | | | | **Date** | | | **Assessment** | | |
| SVS-\_Ed1 | | | | | | DD/MM/YYYY | | | NT / NA / NOK / POK / OK | | |
| **TS Tested requirements** | | | | | | **RB Tested requirements** | | | | | |
| N/A | | | | | | REQ-LFR-SRS-5233\_Ed1 | | | | | |
| **Component:** | | | |  | | **Version:** | V1 | | | | |
| **Involved subsystems** | | | | | | | | | | | |
| SW: 3.2.0.24 | | |  | | | | | | | | |
| HW: 1.1.91 | | | StarDundee | | | | | | | | |
| Start time: | | | (DD/MM/YYYY) hh:mm | | | Test Operator: | |  | | | |
| End Time: | | | (DD/MM/YYYY) hh:mm | | | Test Duration: | | 12 mn | | | |
| **Purpose:** | | Check the LFR FSW is able to receive, to process and to execute the command packets without affecting its other running independent processes. | | | | | | | | | |
| **General remarks about the test** | | | | | | | | | | | |
| Requirement Title | | | Command management. | | | | | | | | |
| dependencies | | | SSS-CP-FS-031, EIDA R-192 | | | | | | | | |
| restrictions | | | The verification of the complete RPW system is not made at this level | | | | | | | | |
| means | | |  | | | | | | | | |
| input data | | |  | | | | | | | | |
| prerequisite | | |  | | | | | | | | |
| used test programs | | |  | | | | | | | | |
| test script | | | Tc\_during\_matrix\_transmission\_asm4s.py and tc\_during\_matrix\_transmission.py | | | | | | | | |
| output data | | |  | | | | | | | | |
| notes | | | For the LFR validation, LFR is isolated; so it’s impossible to ensure that the on-board reception, processing and execution of telecommand do not affect other instruments processes.  For the LFR, the sending of TM\_LFR\_SCIENCE\_\* is the task independent. During this task, a TC is sent. | | | | | | | | |
| **Procedure steps** | | | | | | | | | | | |
| **Step** | **Actions** | | | | **Expected Results** | | | | | **Comments** | **Status** |
|  | LFR FSW is in NORMAL mode. It receipts erroneous TC during matrix transmission:  Send TC\_LFR\_DUMP\_PAR with erroneous CRC, during TM reception:  TM\_LFR\_SCIENCE\_NORMAL\_SWF\_F0  TM\_LFR\_SCIENCE\_NORMAL\_ASM\_F0. | | | | Verify:  -matrix transmission continues normally until its expected end (not stopped immediately by the action).  -Acceptance failure is sent.  -HK transmission is not affected by the action. | | | | |  |  |
|  | LFR FSW is in NORMAL mode. Activate burst mode during matrix transmission (TM\_LFR\_SCIENCE\_NORMAL\_SWF\_F0). | | | | Verify:  -matrix transmission continues normally until its expected end (not stopped immediately by the action).  -Acceptance is sent.  -HK transmission is not affected by the action. | | | | |  |  |
|  | LFR FSW is in SBM1 mode. It receipts nominal TC during matrix transmission:  Send TC\_LFR\_DUMP\_PAR, during TM\_LFR\_SCIENCE\_SBM1\_CWF\_F1 reception. | | | | Verify:  -matrix transmission continues normally until its expected end (not stopped immediately by the action).  -Acceptance success is sent.  -HK transmission is not affected by the action. | | | | |  |  |
|  | End | | | |  | | | | |  |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE** | | | | | | | | |
| **Test procedure identifier** | | | | | **Date** | **Assessment** | | |
| SVS-\_Ed3 | | | | | DD/MM/YYYY | NT / NA / NOK / POK / OK | | |
| **TS Tested requirements** | | | | | **RB Tested requirements** | | | |
| N/A | | | | | REQ-LFR-SRS-5234\_Ed1 REQ-LFR-SRS-5236\_Ed1  REQ-LFR-SRS-5237\_Ed1 REQ-LFR-SRS-5206\_Ed1  REQ-LFR-SRS-5220\_Ed1 REQ-LFR-SRS-5564\_Ed1  REQ-LFR-SRS-5407\_Ed1 REQ-LFR-SRS-5209\_Ed1  REQ-LFR-SRS-5546\_Ed1 | | | |
| **Component:** | | |  | | **Version:** V2 V1 |  | | |
| **Involved subsystems** | | | | | | | | |
| SW: 3.2.0.24 | | |  | | | | | |
| HW:1.1.91 | | | StarDundee | | | | | |
| Start time: | | | (DD/MM/YYYY) hh:mm | | Test Operator: |  | | |
| End Time: | | | (DD/MM/YYYY) hh:mm | | Test Duration: 3mn |  | | |
| **Purpose:** | | | Check the execution of each TC is verifiable through a resulting change in the value of a TM parameter.  The LFR FSW shall compute the CRC on the CCSDS Telecommand Source Packet part of the received packet | | | | | |
| **General remarks about the test** | | | | | | | | |
| Requirement Title | | Command management. And DPU / RPW Analyzer SW communication protocol | | | | | | |
| dependencies | | SSS-CP-FS-032 and SSS-CP-FS-055 and SSS-CP-FS-376 and SSS-CP-FS-075 and SSS-CP-EQS-354 and  SSS-IF-DPS-EQ-160 and SSS-IF-DPS-EQ-180 SSS-CP-EQS-170  RPW-SYS-MEB-LFR-ICD-00097, SSS-CP-FS-032, EIDA R-402 | | | | | | |
| restrictions | | -Verification is only on the “Telecommand Verification Packets”. Functional results (by TM transmission -Housekeeping Reporting, Science, LFR Configuration-) are not tested.  -Exhaustive coverage of all cases is infeasible; the steps below are a subjective compromise to perform a suitable verification. | | | | | | |
| means | |  | | | | | | |
| input data | |  | | | | | | |
| prerequisite | |  | | | | | | |
| used test programs | |  | | | | | | |
| test script | | loop\_tm\_lfr\_tc\_exe(): Nominal TC+TC with error detected during the acceptance stage verifications (wrong CRC) | | | | | | |
| output data | |  | | | | | | |
| notes | | For this TEST CASE, in the "ack" field in each telecommand packet:  ACK\_ACCEPTANCE=”report required” (1).  ACK\_EXECUTION\_COMPLETION=”report required” (1).  TC\_LFR\_RESET is only sent in with error in acceptance stage | | | | | | |
| **Procedure steps** | | | | | | | | |
| **Step** | **Actions** | | | **Expected Results** | | | **Comments** | **Status** |
|  | Nominal TC\_LFR\_ENTER\_MODE.  For mode=0..4; loop all the steps below. | | | Verify the LFR FSW: -sends TM\_LFR\_TC\_EXE\_SUCCESS in case of change LFR mode (TM\_LFR\_TC\_EXE\_NOT\_EXECUTABLE if LFR is already in the target mode). -transmits consistent PA\_RPW\_TELECOMMAND\_PKT\_ID and PA\_RPW\_PKT\_SEQ\_CONTROL parameters. | | |  |  |
|  | Nominal TC\_LFR\_RESET. | | | Verify the LFR FSW: -sends TM\_LFR\_TC\_EXE\_SUCCESS, with consistent PA\_RPW\_TELECOMMAND\_PKT\_ID and PA\_RPW\_PKT\_SEQ\_CONTROL parameters. | | | TC\_LFR\_RESET performs exit() |  |
|  | TC\_LFR\_RESET with error detected during the acceptance stage verifications (wrong CRC). | | | Verify the LFR FSW: -sends TM\_LFR\_TC\_EXE\_CORRUPTED.  -transmits consistent PA\_RPW\_COMPUTED\_CRC parameter. | | |  |  |
|  | Nominal TC\_LFR\_LOAD\_COMMON\_PAR. | | | Verify the LFR FSW: -sends TM\_LFR\_TC\_EXE\_SUCCESS. -transmits consistent PA\_RPW\_TELECOMMAND\_PKT\_ID and PA\_RPW\_PKT\_SEQ\_CONTROL parameters. | | |  |  |
|  | TC\_LFR\_LOAD\_COMMON\_PAR with error detected during the acceptance stage verifications (wrong CRC). | | | Verify the LFR FSW: -sends TM\_LFR\_TC\_EXE\_CORRUPTED.  -transmits consistent PA\_RPW\_COMPUTED\_CRC parameter. | | |  |  |
|  | Nominal TC\_LFR\_LOAD\_NORMAL\_PAR. | | | Verify the LFR FSW: -sends TM\_LFR\_TC\_EXE\_SUCCESS (except in NORMAL, SBM1 and SBM2: NOT\_EXECUTABLE). -transmits consistent PA\_RPW\_TELECOMMAND\_PKT\_ID and PA\_RPW\_PKT\_SEQ\_CONTROL parameters. | | |  |  |
|  | TC\_LFR\_LOAD\_NORMAL\_PAR with error detected during the acceptance stage verifications (wrong CRC). | | | Verify the LFR FSW: -sends TM\_LFR\_TC\_EXE\_CORRUPTED.  -transmits consistent PA\_RPW\_COMPUTED\_CRC parameter. | | |  |  |
|  | Nominal TC\_LFR\_LOAD\_BURST\_PAR. | | | Verify the LFR FSW: -sends TM\_LFR\_TC\_EXE\_SUCCESS(except in BURST: NOT\_EXECUTABLE). -transmits consistent PA\_RPW\_TELECOMMAND\_PKT\_ID and PA\_RPW\_PKT\_SEQ\_CONTROL parameters. | | |  |  |
|  | TC\_LFR\_LOAD\_BURST\_PAR with error detected during the acceptance stage verifications (wrong CRC). | | | Verify the LFR FSW: -sends TM\_LFR\_TC\_EXE\_CORRUPTED.  -transmits consistent PA\_RPW\_COMPUTED\_CRC parameter. | | |  |  |
|  | Nominal TC\_LFR\_LOAD\_SBM1\_PAR. | | | Verify the LFR FSW: -sends TM\_LFR\_TC\_EXE\_SUCCESS (except in SBM1: NOT\_EXECUTABLE). -transmits consistent PA\_RPW\_TELECOMMAND\_PKT\_ID and PA\_RPW\_PKT\_SEQ\_CONTROL parameters. | | |  |  |
|  | TC\_LFR\_LOAD\_SBM1\_PAR with error detected during the acceptance stage verifications (wrong CRC). | | | Verify the LFR FSW: -sends TM\_LFR\_TC\_EXE\_CORRUPTED.  -transmits consistent PA\_RPW\_COMPUTED\_CRC parameter. | | |  |  |
|  | Nominal TC\_LFR\_LOAD\_SBM2\_PAR. | | | Verify the LFR FSW: -sends TM\_LFR\_TC\_EXE\_SUCCESS (except in SBM2: NOT\_EXECUTABLE). -transmits consistent PA\_RPW\_TELECOMMAND\_PKT\_ID and PA\_RPW\_PKT\_SEQ\_CONTROL parameters. | | |  |  |
|  | TC\_LFR\_LOAD\_SBM2\_PAR with error detected during the acceptance stage verifications (wrong CRC). | | | Verify the LFR FSW: -sends TM\_LFR\_TC\_EXE\_CORRUPTED.  -transmits consistent PA\_RPW\_COMPUTED\_CRC parameter. | | |  |  |
|  | Nominal TC\_LFR\_DUMP\_PAR. | | | Verify the LFR FSW: -sends TM\_LFR\_TC\_EXE\_SUCCESS. -transmits consistent PA\_RPW\_TELECOMMAND\_PKT\_ID and PA\_RPW\_PKT\_SEQ\_CONTROL parameters. | | |  |  |
|  | TC\_LFR\_DUMP\_PAR with error detected during the acceptance stage verifications (wrong CRC). | | | Verify the LFR FSW: -sends TM\_LFR\_TC\_EXE\_CORRUPTED.  -transmits consistent PA\_RPW\_COMPUTED\_CRC parameter. | | |  |  |
|  | Nominal TC\_LFR\_ENABLE\_CALIBRATION. | | | Verify the LFR FSW: -sends TM\_LFR\_TC\_EXE\_SUCCESS. -transmits consistent PA\_RPW\_TELECOMMAND\_PKT\_ID and PA\_RPW\_PKT\_SEQ\_CONTROL parameters. | | |  |  |
|  | TC\_LFR\_ENABLE\_CALIBRATION with error detected during the acceptance stage verifications (wrong CRC). | | | Verify the LFR FSW: -sends TM\_LFR\_TC\_EXE\_CORRUPTED.  -transmits consistent PA\_RPW\_COMPUTED\_CRC parameter. | | |  |  |
|  | Nominal TC\_LFR\_TC\_LFR\_DISABLE\_CALIBRATION. | | | Verify the LFR FSW: -sends TM\_LFR\_TC\_EXE\_SUCCESS. -transmits consistent PA\_RPW\_TELECOMMAND\_PKT\_ID and PA\_RPW\_PKT\_SEQ\_CONTROL parameters. | | |  |  |
|  | TC\_LFR\_TC\_LFR\_DISABLE\_CALIBRATION with error detected during the acceptance stage verifications (wrong CRC). | | | Verify the LFR FSW: -sends TM\_LFR\_TC\_EXE\_CORRUPTED.  -transmits consistent PA\_RPW\_COMPUTED\_CRC parameter. | | |  |  |
|  | Nominal TC\_LFR\_UPDATE\_TIME.  (Wait 1.1\*SY\_RPW\_ACK\_RESPONSE\_TIME) | | | Verify the LFR FSW:  -increases HK\_LFR\_DPU\_SPW\_PKT\_RCV\_CNT and HK\_LFR\_UPDATE\_TIME\_TC\_CNT. -starts the execution of the commands: verify the TIME field in the TM\_LFR\_HK (No Timecode sent) | | | REQ-LFR-SRS-5237\_Ed1 |  |
|  | TC\_LFR\_UPDATE\_TIME with error detected during the acceptance stage verifications (wrong CRC).  (Wait 1.1\*SY\_RPW\_ACK\_RESPONSE\_TIME) | | | Verify the LFR FSW:  -sends TM\_LFR\_HK with HK\_LFR\_DPU\_SPW\_PKT\_RCV\_CNT increased. | | | REQ-LFR-SRS-5237\_Ed1 |  |
|  | Nominal TC\_LFR\_UPDATE\_INFO.  (Wait 1.1\*SY\_RPW\_ACK\_RESPONSE\_TIME) | | | Verify the LFR FSW sends TM\_LFR\_HK with HK\_LFR\_DPU\_SPW\_PKT\_RCV\_CNT and  HK\_LFR\_UPDATE\_INFO\_TC\_CNT increased. | | |  |  |
|  | TC\_LFR\_UPDATE\_INFO with error detected during the acceptance stage verifications (wrong CRC).  (Wait 1.1\*SY\_RPW\_ACK\_RESPONSE\_TIME) | | | Verify the LFR FSW sends TM\_LFR\_HK with HK\_LFR\_DPU\_SPW\_PKT\_RCV\_CNT increased. | | |  |  |
|  | Nominal TC\_LFR\_LOAD\_KCOEFFICIENTS. | | | Verify the LFR FSW. -sends TM\_LFR\_TC\_EXE\_SUCCESS. -transmits consistent PA\_RPW\_TELECOMMAND\_PKT\_ID and PA\_RPW\_PKT\_SEQ\_CONTROL parameters. | | |  |  |
|  | TC\_LFR\_LOAD\_KCOEFFICIENTS with error detected during the acceptance stage verifications (wrong CRC).  (Wait 1.1\*SY\_RPW\_ACK\_RESPONSE\_TIME) | | | Verify the LFR FSW: -sends TM\_LFR\_TC\_EXE\_CORRUPTED.  -transmits consistent PA\_RPW\_COMPUTED\_CRC parameter. | | |  |  |
|  | Nominal TC\_LFR\_DUMP \_KCOEFFICIENTS. | | | Verify the LFR FSW  - sends 2 TM\_LFR\_KCOEFFICIENTS\_DUMP. -sends TM\_LFR\_TC\_EXE\_SUCCESS. -transmits consistent PA\_RPW\_TELECOMMAND\_PKT\_ID and PA\_RPW\_PKT\_SEQ\_CONTROL parameters. | | |  |  |
|  | TC\_LFR\_DUMP\_KCOEFFICIENTS with error detected during the acceptance stage verifications (wrong CRC).  (Wait 1.1\*SY\_RPW\_ACK\_RESPONSE\_TIME) | | | Verify the LFR FSW: -sends TM\_LFR\_TC\_EXE\_CORRUPTED.  -transmits consistent PA\_RPW\_COMPUTED\_CRC parameter. | | |  |  |
|  | Nominal TC\_LFR\_LOAD\_FBINS\_MASK | | | Verify the LFR FSW. -sends TM\_LFR\_TC\_EXE\_SUCCESS. -transmits consistent PA\_RPW\_TELECOMMAND\_PKT\_ID and PA\_RPW\_PKT\_SEQ\_CONTROL parameters. | | |  |  |
|  | TC\_LFR\_LOAD\_FBINS\_MASK with error detected during the acceptance stage verifications (wrong CRC).  (Wait 1.1\*SY\_RPW\_ACK\_RESPONSE\_TIME) | | | Verify the LFR FSW: -sends TM\_LFR\_TC\_EXE\_CORRUPTED.  -transmits consistent PA\_RPW\_COMPUTED\_CRC parameter. | | |  |  |
|  | Send a TC with length >  SY\_LFR\_TC\_MAX\_LEN  (228 bytes) | | | Spacewire rejects the TC too long  In TM\_LFR\_HK; only these Two fields are updated:  HK\_LFR\_DPU\_SPW\_RX\_TOO\_BIG is incremented  HK\_LFR\_ME\_CNT is incremented | | |  |  |
|  | Send a TC with length >  SY\_LFR\_TC\_MAX\_LEN  (228 bytes) with error detected during the acceptance stage verifications (wrong CRC).  (Wait 1.1\*SY\_RPW\_ACK\_RESPONSE\_TIME) | | | WARNING:  NO TM\_LFR\_TC\_EXE\_CORRUPTED because the spacewire rejects the TC too long first.  In TM\_LFR\_HK; only these Two fields are updated:  HK\_LFR\_DPU\_SPW\_RX\_TOO\_BIG is incremented  HK\_LFR\_ME\_CNT is incremented | | |  |  |
|  | Nominal TC\_LFR\_LOAD\_ FILTER\_PAR | | | Verify the LFR FSW. -sends TM\_LFR\_TC\_EXE\_SUCCESS. -transmits consistent PA\_RPW\_TELECOMMAND\_PKT\_ID and PA\_RPW\_PKT\_SEQ\_CONTROL parameters. | | |  |  |
|  | TC\_LFR\_LOAD\_FILTER\_PAR with error detected during the acceptance stage verifications (wrong CRC).  (Wait 1.1\*SY\_RPW\_ACK\_RESPONSE\_TIME) | | | Verify the LFR FSW: -sends TM\_LFR\_TC\_EXE\_CORRUPTED.  -transmits consistent PA\_RPW\_COMPUTED\_CRC parameter. | | |  |  |
|  | For mode=0..4; loop the steps 34-35  Send a TC with length <  CCSDS\_TC\_PKT\_MIN\_SIZE  (12 bytes)  (Wait 1.1\*SY\_RPW\_ACK\_RESPONSE\_TIME) | | | Verify the LFR\_FSW don’t acknowledge this TC  In TM\_LFR\_HK, only HK\_LFR\_DPU\_SPW\_PKT\_RCV\_CNT field is incremented. | | |  |  |
|  | Send a TC with length <  CCSDS\_TC\_PKT\_MIN\_SIZE  (12 bytes)  with error detected during the acceptance stage verifications (wrong CRC).  (Wait 1.1\*SY\_RPW\_ACK\_RESPONSE\_TIME) | | | Verify the LFR\_FSW don’t acknowledge this TC  In TM\_LFR\_HK, only HK\_LFR\_DPU\_SPW\_PKT\_RCV\_CNT field is incremented. | | |  |  |
|  | For mode=0..4  TC\_LFR\_ENTER\_MODE with error detected during the acceptance stage verifications (wrong CRC). | | | Verify the LFR FSW: -sends TM\_LFR\_TC\_EXE\_CORRUPTED.  -transmits consistent PA\_RPW\_COMPUTED\_CRC parameter. | | |  |  |
|  | End | | |  | | |  |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE** | | | | | | | |
| **Test procedure identifier** | | | **Date** | **Assessment** | | | |
| SVS-0005\_Ed2 | | | DD/MM/YYYY | NT / NA / NOK / POK / OK | | | |
| **TS Tested requirements** | | | **RB Tested requirements** | | | | |
| N/A | | | REQ-LFR-SRS-5203\_Ed2 REQ-LFR-SRS-5206\_Ed1 | | | | |
| **Component:** | |  | **Version:** V2 |  | | | |
| **Involved subsystems** | | | | | | | |
| SW: 3.2.0.24 | |  | | | | | |
| HW:1.1.91 | | Stardundee | | | | | |
| Start time: | | (DD/MM/YYYY) hh:mm | Test Operator: |  | | | |
| End Time: | | (DD/MM/YYYY) hh:mm | Test Duration: 30sec |  | | | |
| **Purpose:** | | Check the LFR FSW validates the received commands prior to their execution (acceptance stage). | | | | | |
| **General remarks about the test** | | | | | | | |
| Requirement Title | | Command management. | | | | | |
| dependencies | | SSS-CP-FS-040, AD06: EIDA R-187/EIDA-2809, AD10:TCV-1 and SSS-CP-FS-075 | | | | | |
| restrictions | |  | | | | | |
| means | |  | | | | | |
| input data | |  | | | | | |
| prerequisite | |  | | | | | |
| used test programs | |  | | | | | |
| test script | | acceptation\_execution\_failures.py | | | | | |
| output data | |  | | | | | |
| notes | |  | | | | | |
| **Procedure steps** | | | | | | | |
| **Step** | **Actions** | | | | **Expected Results** | **Comments** | **Status** |
|  | For each TC in TC\_LFR\_LOAD\_COMMON\_PAR, TC\_LFR\_LOAD\_NORMAL\_PAR, TC\_LFR\_LOAD\_BURST\_PAR, TC\_LFR\_LOAD\_SBM1\_PAR, TC\_LFR\_LOAD\_SBM2\_PAR, TC\_LFR\_DUMP\_PAR, TC\_LFR\_ENABLE\_CALIBRATION, TC\_LFR\_DISABLE\_CALIBRATION,TC\_LFR\_LOAD\_KCOEFFICIENTS,TC\_LFR\_KCOEFFICIENTS\_DUMP,TC\_LFR\_LOAD\_FBINS\_MASK,TC\_LFR\_LOAD\_FILTER\_PAR:  -Send that nominal TC; ensure the LFR returns TM\_LFR\_TC\_EXE\_SUCCESS  -Send that TC with forbidden source ID (and CRC updated). | | | | Check the LFR FSW validates the acceptance stage: TM\_LFR\_TC\_EXE\_CORRUPTED. |  |  |
|  | For each <mode> in NORMAL, BURST, SBM1, SBM2:  -Enter in that mode.  -Send nominal TC\_LFR\_LOAD\_<mode>\_LOAD; ensure the LFR returns TM\_LFR\_TC\_EXE\_NOT\_EXECUTABLE.  -Send that TC\_LFR\_LOAD with forbidden source ID (and CRC updated). | | | | Check the LFR FSW validates the acceptance stage: TM\_LFR\_TC\_EXE\_CORRUPTED. |  |  |
|  | For each <mode> in STANDBY, NORMAL, BURST, SBM1, SBM2:  -Enter in that mode. Ensure the expected mode is reached.  -Send again that nominal TC\_LFR\_ENTER\_MODE; ensure the LFR returns TM\_LFR\_TC\_EXE\_NOT\_EXECUTABLE.  -Send that TC\_LFR\_ENTER\_MODE with forbidden source ID (and CRC updated). | | | | Check the LFR FSW validates the acceptance stage: TM\_LFR\_TC\_EXE\_CORRUPTED. |  |  |
|  | End | | | |  |  |  |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE** | | | | | | | | | | |
| **Test procedure identifier** | | | | | | **Date** | | **Assessment** | | |
| SVS-0007\_Ed2 | | | | | | DD/MM/YYYY | | NT / NA / NOK / POK / OK | | |
| **TS Tested requirements** | | | | | | **RB Tested requirements** | | | | |
| N/A | | | | | | REQ-LFR-SRS-5205\_Ed2 REQ-LFR-SRS-5204\_Ed1  REQ-LFR-SRS-5212\_Ed1 REQ-LFR-SRS-5206\_Ed1 | | | | |
| **Component:** | | | | |  | **Version:** V2 | |  | | |
| **Involved subsystems** | | | | | | | | | | |
| SW: 3.2.0.24 | | | | |  | | | | | |
| HW:1.1.91 | | | | | Stardundee | | | | | |
| Start time: | | | | | (DD/MM/YYYY) hh:mm | Test Operator: | |  | | |
| End Time: | | | | | (DD/MM/YYYY) hh:mm | Test Duration :3mn40sec | |  | | |
| **Purpose:** | | Check if the acceptance of the command fails, the LFR FSW systematically generates a Telecommand Execution Completed Failure Report (TM\_LFR\_TC\_EXE\_CORRUPTED or TM\_LFR\_TC\_EXE\_NOT\_EXECUTABLE) conforming to the PUS telecommand verification service, whatever the value of the acknowledgment flag in the command packet header.  This requirement does not apply to the following packets: TC\_LFR\_UPDATE\_INFO, TC\_LFR\_UPDATE\_TIME.  Criterias are defined by REQ-LFR-SRS-5204\_Ed1  Upon the reception of any command packet, the LFR FSW shall verify if the packet can be accepted by:   * Checking the following items in this specific order :   + APID (PID, CAT)   + Length of the received packet should match packet\_length field contained in packet header   + service type and service subtype   + source ID   + Length\_packet field value is relevant with expected length considering definition of the sub-type.   + packet error control (CRC). The algorithm to compute the packet error control is specified in the [AD10], appendix 6.   If at least one of this criteria fails, TM\_LFR\_EXE\_CORRUPTED packet is emitted with failure code 42005   * Checking that the command is allowed in the current mode (see tab 5.2) and/or state of the instrument.   No check of the packet sequence counter shall be made | | | | | | | | |
| **General remarks about the test** | | | | | | | | | | |
| Requirement Title | | | | Command acceptance stage / Content of the verification report | | | | | | |
| dependencies | | | | AD05, SSS-CP-FS-065 SSS-CP-FS-050 SSS-CP-FS-120 SSS-CP-FS-075 | | | | | | |
| restrictions | | | |  | | | | | | |
| means | | | |  | | | | | | |
| input data | | | |  | | | | | | |
| prerequisite | | | | Acceptance of the command fails. | | | | | | |
| used test programs | | | |  | | | | | | |
| test script | | | | loop\_tm\_lfr\_tc\_exe\_no\_ack.py | | | | | | |
| output data | | | |  | | | | | | |
| notes | | | -criterias of acceptance failed are listed in REQ-LFR-SRS-5204  -CRC test : each byte will be tested  TC too long or too short don’t reach this acceptance stage. | | | | | | | |
| **Procedure steps** | | | | | | | | | | |
| **Step** | **Actions** | | | | | | **Expected Results** | | **Comments** | **Status** |
|  | For each combinaison of ack\_execution and ack\_acceptance send a TC with  Failed CRC first byte  Failed CRC second byte  Invalid PID for LFR (PID=9)  Valid LFR PID but not the right one for the TC  Invalid LFR CAT (CAT different to 1)  Invalid LFR Type (Type=13)  Valid LFR Type but not the right one for the TC  Invalid LFR SUBTYPE  Bad packet-length (65535)  Invalid source\_id (255)  Send a  TC\_LFR\_DISABLE\_CALIBRATION with error  detected during the acceptance stage verifications. | | | | | | Ensure FSW replies TM\_LFR\_TC\_EXE\_CORRUPTED. | |  |  |
|  | TC\_LFR\_DUMP\_PAR with error detected during the acceptance stage verifications. | | | | | | Ensure FSW replies TM\_LFR\_TC\_EXE\_CORRUPTED. | |  |  |
|  | TC\_LFR\_ENABLE\_CALIBRATION with error detected during the acceptance stage verifications verifications | | | | | | Ensure FSW replies TM\_LFR\_TC\_EXE\_CORRUPTED. | |  |  |
|  | TC\_LFR\_ENTER\_MODE with error detected during the acceptance stage verifications. | | | | | | Ensure FSW replies TM\_LFR\_TC\_EXE\_CORRUPTED. | |  |  |
|  | TC\_LFR\_LOAD\_BURST\_PAR with error detected during the acceptance stage verifications. | | | | | | Ensure FSW replies TM\_LFR\_TC\_EXE\_CORRUPTED. | |  |  |
|  | TC\_LFR\_LOAD\_COMMON\_PAR with error detected during the acceptance stage verifications. | | | | | | Ensure FSW replies TM\_LFR\_TC\_EXE\_CORRUPTED. | |  |  |
|  | TC\_LFR\_LOAD\_NORMAL\_PAR with error detected during the acceptance stage verifications. | | | | | | Ensure FSW replies TM\_LFR\_TC\_EXE\_CORRUPTED. | |  |  |
|  | TC\_LFR\_LOAD\_SBM1\_PAR with error detected during the acceptance stage verifications. | | | | | | Ensure FSW replies TM\_LFR\_TC\_EXE\_CORRUPTED. | |  |  |
|  | TC\_LFR\_LOAD\_SBM2\_PAR with error detected during the acceptance stage verifications. | | | | | | Ensure FSW replies TM\_LFR\_TC\_EXE\_CORRUPTED. | |  |  |
|  | TC\_LFR\_LOAD\_KCOEFFICIENTS with error detected during the acceptance stage verifications. | | | | | | Ensure FSW replies TM\_LFR\_TC\_EXE\_CORRUPTED | |  |  |
|  | TC\_LFR\_ DUMP\_KCOEFFICIENTS with error detected during the acceptance stage verifications. | | | | | | Ensure FSW replies TM\_LFR\_TC\_EXE\_CORRUPTED | |  |  |
|  | TC\_LFR\_LOAD\_FBINS\_MASK with error detected during the acceptance stage verifications. | | | | | | Ensure FSW replies TM\_LFR\_TC\_EXE\_CORRUPTED | |  |  |
|  | TC\_LFR\_LOAD\_FILTER\_PAR with error detected during the acceptance stage verifications. | | | | | | Ensure FSW replies TM\_LFR\_TC\_EXE\_CORRUPTED | |  |  |
|  | TC\_LFR\_RESET with error detected during the acceptance stage verifications. | | | | | | Ensure FSW replies TM\_LFR\_TC\_EXE\_CORRUPTED. | |  |  |
|  | In STANDBY, send all TC\_LFR\_xx, excepted TC\_LFR\_UPDATE\_xxx and TC\_LFR\_RESET. | | | | | | Verify if the acceptance of the command fails, the LFR FSW systematically generates a Telecommand Execution Completed Failure Report FSW replies.  Only TM\_LFR\_TC\_EXE\_NOT\_EXCUTABLE will be seen. | |  |  |
|  | In NORMAL, send all TC\_LFR\_xx, excepted TC\_LFR\_UPDATE\_xxx and TC\_LFR\_RESET. | | | | | | Verify if the acceptance of the command fails, the LFR FSW systematically generates a Telecommand Execution Completed Failure Report FSW replies. | |  |  |
|  | In BURST, send all TC\_LFR\_xx, excepted TC\_LFR\_UPDATE\_xxx and TC\_LFR\_RESET. | | | | | | Verify if the acceptance of the command fails, the LFR FSW systematically generates a Telecommand Execution Completed Failure Report FSW replies. | |  |  |
|  | In SBM1, send all TC\_LFR\_xx, excepted TC\_LFR\_UPDATE\_xxx and TC\_LFR\_RESET. | | | | | | Verify if the acceptance of the command fails, the LFR FSW systematically generates a Telecommand Execution Completed Failure Report FSW replies. | |  |  |
|  | In SBM2, send all TC\_LFR\_xx, excepted TC\_LFR\_UPDATE\_xxx and TC\_LFR\_RESET. | | | | | | Verify if the acceptance of the command fails, the LFR FSW systematically generates a Telecommand Execution Completed Failure Report FSW replies. | |  |  |
|  | For each combinaison of ack\_execution and ack\_acceptance  send all TC with  a wrong packet-length vs Type/subtype | | | | | | Ensure FSW replies TM\_LFR\_TC\_EXE\_CORRUPTED  except for TC\_LFR\_UPDATE\_xxx. | |  |  |
|  | End | | | | | |  | |  |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE** | | | | | | | | |
| **Test procedure identifier** | | | | | **Date** | | **Assessment** | |
| SVS-0008\_Ed2 | | | | | DD/MM/YYYY | | NT / NA / NOK / POK / OK | |
| **TS Tested requirements** | | | | | **RB Tested requirements** | | | |
| N/A | | | | | REQ-LFR-SRS-5207\_Ed2 REQ-LFR-SRS-5212\_Ed1  REQ-LFR-SRS-5206\_Ed1 REQ-LFR-SRS-5202\_Ed1  REQ-LFR-SRS-5207\_Ed2 | | | |
| **Component:** | | | |  | **Version:** V2 | |  | |
| **Involved subsystems** | | | | | | | | |
| SW: 3.2.0.24 | | | |  | | | | |
| HW:1.1.91 | | | | Stardundee | | | | |
| Start time: | | | | (DD/MM/YYYY) hh:mm | Test Operator: | |  | |
| End Time: | | | | (DD/MM/YYYY) hh:mm | Test Duration:22sec | |  | |
| **Purpose:** | | Check, if the execution of the command fails, the LFR FSW generates a Telecommand Execution Completed Failure Report conforming to the PUS telecommand verification service, even if this has not been requested in the execution acknowledgement flag in the telecommand packet header.  This requirement does not apply to the following packets: TC\_LFR\_UPDATE\_INFO, TC\_LFR\_UPDATE\_TIME. | | | | | | |
| **General remarks about the test** | | | | | | | | |
| Requirement Title | | | Command execution stage | | | | | |
| dependencies | | | SSS-CP-FS-080, “Solar Orbiter Operations Requirements Document”-TCV-5 B and SSS-CP-FS-120 and  SSS-CP-FS-075 and SSS-CP-FS-100 and SSS-CP-EQS-200 | | | | | |
| restrictions | | |  | | | | | |
| means | | |  | | | | | |
| input data | | |  | | | | | |
| prerequisite | | | Acceptance of the command does not fail. | | | | | |
| used test programs | | |  | | | | | |
| test script | | | tc\_execution\_failure\_report.py | | | | | |
| output data | | |  | | | | | |
| notes | | | ACK\_EXECUTION\_COMPLETION=0 («no report required»).  Note case bit=1 is seen in REQ-LFR-SRS-5234 (SVS-0003).  TC\_LFR\_DISABLE\_CALIBRATION TC\_LFR\_ENABLE\_CALIBRATION TC\_LFR\_DUMP\_PAR TC\_LFR\_DUMP\_KCOEFFICIENTS TC\_LFR\_LOAD\_FBINS\_MASK TC\_LFR\_LOAD\_COMMON\_PAR never generated a failure report. | | | | | |
| **Procedure steps** | | | | | | | | |
| **Step** | **Actions** | | | | | **Expected Results** | **Comments** | **Status** |
|  | TC\_LFR\_ENTER\_MODE with wrong or inconsistent data field. | | | | | Verify the LFR FSW sends TM\_LFR\_TC\_EXE\_INCONSISTENT with  PA\_RPW\_BYTE\_POSITION=11 |  |  |
|  | TC\_LFR\_ENTER\_MODE when it cannot be executed at this time either when enter\_mode = current\_mode | | | | | Verify the LFR FSW sends TM\_LFR\_TC\_EXE\_NOT\_EXECUTABLE. | REQ-LFR-SRS-5202\_Ed1 |  |
|  | TC\_LFR\_LOAD\_BURST\_PAR with wrong / inconsistent data field. | | | | | Verify the LFR FSW sends TM\_LFR\_TC\_EXE\_INCONSISTENT. | See SVS-0065  all parameters are tested |  |
|  | TC\_LFR\_LOAD\_BURST\_PAR when it cannot be executed at this time either in BURST mode. | | | | | Verify the LFR FSW sends TM\_LFR\_TC\_EXE\_NOT\_EXECUTABLE. |  |  |
|  | TC\_LFR\_LOAD\_NORMAL\_PAR with wrong /inconsistent data field. | | | | | Verify the LFR FSW sends TM\_LFR\_TC\_EXE\_INCONSISTENT. | See SVS-0065  all parameters are tested |  |
|  | TC\_LFR\_LOAD\_NORMAL\_PAR when it cannot be executed at this time either in NORMAL, SBM1, SBM2 modes. | | | | | Verify the LFR FSW sends TM\_LFR\_TC\_EXE\_NOT\_EXECUTABLE. |  |  |
|  | TC\_LFR\_LOAD\_SBM1\_PAR with wrong or inconsistent data field. | | | | | Verify the LFR FSW sends TM\_LFR\_TC\_EXE\_INCONSISTENT. | See SVS-0065  all parameters are tested |  |
|  | TC\_LFR\_LOAD\_SBM1\_PAR when it cannot be executed at this time either in SBM1 mode. | | | | | Verify the LFR FSW sends TM\_LFR\_TC\_EXE\_NOT\_EXECUTABLE. |  |  |
|  | TC\_LFR\_LOAD\_SBM2\_PAR with wrong or inconsistent data field. | | | | | Verify the LFR FSW sends TM\_LFR\_TC\_EXE\_INCONSISTENT. | See SVS-0065  all parameters are tested |  |
|  | TC\_LFR\_LOAD\_SBM2\_PAR when it cannot be executed at this time. | | | | | Verify the LFR FSW sends TM\_LFR\_TC\_EXE\_NOT\_EXECUTABLE. |  |  |
|  | TC\_LFR\_LOAD\_KCOEFFICIENTS with wrong / inconsistent data field. | | | | | Verify the LFR FSW sends TM\_LFR\_TC\_EXE\_INCONSISTENT with  PA\_RPW\_BYTE\_POSITION=10  Only one inconsistent parameter. |  |  |
|  | TC\_LFR\_LOAD\_FILTER\_PAR with wrong/inconsistent data field | | | | | Verify the LFR FSW sends TM\_LFR\_TC\_EXE\_INCONSISTENT | See SVS-0065  all parameters are tested |  |
|  | End | | | | |  |  |  |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE** | | | | | | | | | |
| **Test procedure identifier** | | | | | **Date** | | **Assessment** | | |
| SVS-0009\_Ed2 | | | | | DD/MM/YYYY | | NT / NA / NOK / POK / OK | | |
| **TS Tested requirements** | | | | | **RB Tested requirements** | | | | |
| N/A | | | | | REQ-LFR-SRS-5209\_Ed2 REQ-LFR-SRS-5206\_Ed1  REQ-LFR-SRS-5202\_Ed1 REQ-LFR-SRS-5210\_Ed1 | | | | |
| **Component:** | | | |  | **Version:** V2 | |  | | |
| **Involved subsystems** | | | | | | | | | |
| SW: 3.2.0.24 | | | |  | | | | | |
| HW: 1.1.91 | | | | Stardundee | | | | | |
| Start time: | | | | (DD/MM/YYYY) hh:mm | Test Operator: | |  | | |
| End Time: | | | | (DD/MM/YYYY) hh:mm | Test Duration: | | 6mn 30sec | | |
| **Purpose:** | | Check, when a command has been properly executed, the LFR FSW systematically generates a report of successful completion for the execution stage, whatever the value of the acknowledgement flag in the command packet header. This requirement does not apply to the following packets: TC\_LFR\_UPDATE\_TIME and TC\_LFR\_UPDATE\_INFO. | | | | | | | |
| **General remarks about the test** | | | | | | | | | |
| Requirement Title | | | Command execution stage | | | | | | |
| dependencies | | | SSS-CP-FS-090, AD10: TCV-4 + TCV-6 and SSS-CP-FS-075  and SSS-CP-FS-100 and SSS-CP-FS-095 | | | | | | |
| restrictions | | |  | | | | | | |
| means | | |  | | | | | | |
| input data | | |  | | | | | | |
| prerequisite | | | A command has been properly executed | | | | | | |
| used test programs | | | completionExecution.py | | | | | | |
| test script | | |  | | | | | | |
| output data | | |  | | | | | | |
| notes | | | Python script surveys the acknowledges; if a TC with expected ack would have no report after the SY\_RPW\_ACK\_RESPONSE\_TIME delay: Timeout will be displayed, and noticed in NB\*.txt file. | | | | | | |
| **Procedure steps** | | | | | | | | | |
| **Step** | **Actions** | | | | | **Expected Results** | | **Comments** | **Status** |
|  | For each combination of ACK\_EXECUTION\_COMPLETION, and ACK\_ACCEPTANCE, reach each mode.  For each mode, send all TC (except the TC\_LFR\_UPDATE\_\*), | | | | | Verify the LFR FSW sends TM\_LFR\_TC\_EXE\_SUCCESS when TC are ok (see tab 5.2 in SRS). | |  |  |
|  | For each combination of ACK\_EXECUTION\_COMPLETION, and ACK\_ACCEPTANCE, make all possible transitions (except when expected mode=current mode ) | | | | | Verify the LFR FSW sends TM\_LFR\_TC\_EXE\_SUCCESS . | |  |  |
|  | End | | | | |  | |  |  |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE** | | | | | | | | | | |
| **Test procedure identifier** | | | | | | **Date** | **Assessment** | | | |
| SVS-0010\_Ed2 | | | | | | DD/MM/YYYY | NT / NA / NOK / POK / OK | | | |
| **TS Tested requirements** | | | | | | **RB Tested requirements** | | | | |
| N/A | | | | | | REQ-LFR-SRS-5211\_Ed1 | | | | |
| **Component:** | | | |  | | **Version:** V2 |  | | | |
| **Involved subsystems** | | | | | | | | | | |
| SW: 3.2.0.24 | | | |  | | | | | | |
| HW:1.1.91 | | | | Stardundee | | | | | | |
| Start time: | | | | (DD/MM/YYYY) hh:mm | | Test Operator: | |  | | |
| End Time: | | | | (DD/MM/YYYY) hh:mm | | Test Duration:50s |  | | | |
| **Purpose:** | | | Check the TC acceptance/execution success report contains:  -copy of the TC Packet ID field.  -copy of the TC Packet Sequence Control field. | | | | | | | |
| **General remarks about the test** | | | | | | | | | | |
| Requirement Title | | Content of the verification reports | | | | | | | | |
| dependencies | | SSS-CP-FS-110, AD10: TCV-7 | | | | | | | | |
| restrictions | |  | | | | | | | | |
| means | |  | | | | | | | | |
| input data | |  | | | | | | | | |
| prerequisite | |  | | | | | | | | |
| used test programs | |  | | | | | | | | |
| test script | | successful\_content\_verification\_reports.py | | | | | | | | |
| output data | |  | | | | | | | | |
| notes | | No feedback for TC\_LFR\_UPDATE\_\* | | | | | | | | |
| **Procedure steps** | | | | | | | | | | |
| **Step** | **Actions** | | | | **Expected Results** | | | | **Comments** | **Status** |
|  | For each mode, send all TC. | | | | Verify the TM for TC acceptance/execution success report contains: -a copy of the TC Packet ID field. -a copy of the TC Packet Sequence Control field. | | | |  |  |
|  | End | | | |  | | | |  |  |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE** | | | | | | | | | |
| **Test procedure identifier** | | | | | | **Date** | **Assessment** | | |
| SVS-0011\_Ed2 | | | | | | DD/MM/YYYY | NT / NA / NOK / POK / OK | | |
| **TS Tested requirements** | | | | | | **RB Tested requirements** | | | |
| N/A | | | | | | REQ-LFR-SRS-5218\_Ed1 | | | |
| **Component:** | | | |  | | **Version:** V2 |  | | |
| **Involved subsystems** | | | | | | | | | |
| SW: 3.2.0.24 | | |  | | | | | | |
| HW:1.1.91 | | | StarDundee | | | | | | |
| Start time: | | | (DD/MM/YYYY) hh:mm | | | Test Operator: |  | | |
| End Time: | | | (DD/MM/YYYY) hh:mm | | | Test Duration:38s |  | | |
| **Purpose:** | | Check after initialization, the LFR FSW starts its local time set the most significant bit of its local time to 1 and all other bits to 0. | | | | | | | |
| **General remarks about the test** | | | | | | | | | |
| Requirement Title | | | Time management | | | | | | |
| dependencies | | | SSS-CP-FS-360,AD06 R-214; EIDA-2931 | | | | | | |
| restrictions | | |  | | | | | | |
| means | | |  | | | | | | |
| input data | | |  | | | | | | |
| prerequisite | | | The CTR SpaceWire time codes are properly transmitted (SY\_RPW\_CTR\_FREQUENCY)  TC\_LFR\_RESET can’t be used because it performs exit(0) | | | | | | |
| used test programs | | |  | | | | | | |
| test script | | | starting\_time.py | | | | | | |
| output data | | |  | | | | | | |
| notes | | |  | | | | | | |
| **Procedure steps** | | | | | | | | | |
| **Step** | **Actions** | | | | **Expected Results** | | | **Comments** | **Status** |
|  | Restart the FSW by an HW reset.  Launch the Python script.  No automatic timecode sent | | | | TIME field (CUC format): verify the LFR FSW sets the most significant bit of its local time to 1.  LFR status shown in TM\_LFR\_HK is:  TIME=0x8000000248ba, HK\_LFR\_MODE: STANDBY = 0, HK\_LFR\_DPU\_SPW\_ENABLED: ENABLED = 1, HK\_LFR\_DPU\_SPW\_LINK\_STATE: RUN = 5HK\_LFR\_SC\_POTENTIEL\_FLAG: ON = 1  SY\_LFR\_WATCHDOG\_ENABLED: ENABLED = 1, HK\_LFR\_CALIB\_ENABLED: DISABLED = 0,  HK\_LFR\_RESET\_CAUSE: POWER\_ON = 1,  One error  HK\_LFR\_LE\_CNT=1  HK\_LFR\_LAST\_ER\_RID: LE\_LFR\_TIMEC = 42129,  HK\_LFR\_LAST\_ER\_CODE: MISSING = 21, HK\_LFR\_LAST\_ER\_TIME=0x800000022f12  HK\_LFR\_TIMECODE\_ERRONEOUS=0, HK\_LFR\_TIMECODE\_MISSING=1, HK\_LFR\_TIMECODE\_INVALID=0, HK\_LFR\_TIME\_TIMECODE\_IT=0, HK\_LFR\_TIME\_NOT\_SYNCHRO=0, HK\_LFR\_TIME\_TIMECODE\_CTR=0 | | |  |  |
|  | Synchronize LFR  Send a TC\_LFR\_UPDATE\_TIME with CP\_RPW\_TIME = 1 and  a valid timecode (1)  wait 5s without timecode | | | | TIME field = 0x000000004c7c on the TM\_LFR\_HK received  status LFR in TM\_LFR\_HK shows the synchronization  HK\_LFR\_DPU\_SPW\_TICK\_OUT\_CNT=1,  HK\_LFR\_DPU\_SPW\_LAST\_TIMC=1  No timing errors detected  After TM\_LFR\_HK shows the timecode absence in the suitable error counter only one time.  HK\_LFR\_TIMECODE\_MISSING=2.  and HK\_LFR\_LE\_CNT=2 | | |  |  |
|  | Reload LFR software and wait the boot sequence. | | | | verify the LFR FSW sets the most significant bit of its local time to 1 and all bits to 0  Time is 0x8000000231ba .  Boot sequence around 2sec  All suitable fields in TM\_HK\_LFR are set to 0.  HK\_LFR\_UPDATE\_TIME\_TC\_CNT=0, HK\_LFR\_LE\_CNT=0, HK\_LFR\_ME\_CNT=0, HK\_LFR\_LAST\_ER\_RID: NO\_ERROR = 0, HK\_LFR\_LAST\_ER\_CODE: NO\_ERROR = 0, HK\_LFR\_LAST\_ER\_TIME=0x000000000000, HK\_LFR\_DPU\_SPW\_TICK\_OUT\_CNT=0, HK\_LFR\_DPU\_SPW\_LAST\_TIMC=0, HK\_LFR\_DPU\_SPW\_PARITY=0, HK\_LFR\_DPU\_SPW\_DISCONNECT=0, HK\_LFR\_DPU\_SPW\_ESCAPE=0, HK\_LFR\_DPU\_SPW\_CREDIT=0, HK\_LFR\_DPU\_SPW\_WRITE\_SYNC=0, HK\_LFR\_DPU\_SPW\_RX\_AHB=0, HK\_LFR\_DPU\_SPW\_TX\_AHB=0, HK\_LFR\_DPU\_SPW\_EARLY\_EOP=0, HK\_LFR\_DPU\_SPW\_INVALID\_ADDR=0, HK\_LFR\_DPU\_SPW\_EEP=0, , HK\_LFR\_TIMECODE\_ERRONEOUS=0, HK\_LFR\_TIMECODE\_MISSING=0, HK\_LFR\_TIMECODE\_INVALID=0, HK\_LFR\_TIME\_TIMECODE\_IT=0, HK\_LFR\_TIME\_NOT\_SYNCHRO=0, HK\_LFR\_TIME\_TIMECODE\_CTR=0  This operation is not equivalent to HW reset. | | |  |  |
|  | End | | | |  | | |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE** | | | | | | | | | | | | | |
| **Test procedure identifier** | | | | | | | | | **Date** | | **Assessment** | | |
| SVS-0012\_Ed2 | | | | | | | | | DD/MM/YYYY | | NT / NA / NOK / POK / OK | | |
| **TS Tested requirements** | | | | | | | | | **RB Tested requirements** | | | | |
| N/A | | | | | | | | | REQ-LFR-SRS-5219\_Ed1 | | | | |
| **Component:** | | | | |  | | | | **Version:** V2 | |  | | |
| **Involved subsystems** | | | | | | | | | | | | | |
| SW: 3.2.0.24 | | | |  | | | | | | | | | |
| HW:1.1.91 | | | | StarDundee | | | | | | | | | |
| Start time: | | (DD/MM/YYYY) hh:mm | | | | | Test Operator: |  | | | | | |
| End Time: | | (DD/MM/YYYY) hh:mm | | | | | Test Duration: | 25mn | | | | | |
| **Purpose:** | | | Check the LFR FSW synchronizes its local time with the CTR distributed as a SpW command packet coupled to a SpW time code:   * The CTR SpaceWire time code is transmitted at SY\_RPW\_CTR\_FREQUENCY. * The CTR SpaceWire command packet containing the CTR is not necessarily sent. * When it is generated, the CTR SpaceWire command packet is transmitted > SY\_RPW\_CTR\_MIN\_DELAY prior to the time code itself. * The CTR SpaceWire command packet is distributed thanks to the “Accept Time Update” command. * The SpaceWire time code contains the least significant bits of the CTR coarse time part. | | | | | | | | | | |
| **General remarks about the test** | | | | | | | | | | | | | |
| Requirement Title | | | | | | Time management | | | | | | | |
| dependencies | | | | | | §3.2.64 Time-Code of DR03. SSS-CP-FS-370, AD06OBTM-1 | | | | | | | |
| restrictions | | | | | |  | | | | | | | |
| means | | | | | |  | | | | | | | |
| input data | | | | | |  | | | | | | | |
| prerequisite | | | | | |  | | | | | | | |
| used test programs | | | | | |  | | | | | | | |
| test script | | | | | | ccsds\_time\_code\_format.py + ccsds\_time\_code\_format\_xxx.py (burst,normal,sbm1,sbm2)  each mode performs step from 1 to 5  send\_timecode\_not\_nominal.py 🡪 step 6 | | | | | | | |
| output data | | | | | |  | | | | | | | |
| notes | | | | | | AD03: 1/SY\_RPW\_CTR\_FREQUENCY=1s, SY\_RPW\_CTR\_MIN\_DELAY=300ms | | | | | | | |
| **Procedure steps** | | | | | | | | | | | | | |
| **Step** | **Actions** | | | | | | | | | **Expected Results** | | **Comments** | **Status** |
|  | Restart the FSW by an HW reset and waitt the end of boot sequence. No automatic timecode is sent.  LFR mode is standby and no synchronized.  For each mode, send the nominal sequence At t0: TC\_LFR\_UPDATE\_TIME with any CP\_RPW\_TIME.  At t0+SY\_RPW\_CTR\_MIN\_DELAY: valid time code  A timecode is valid:   1. For spacewire, timecode = previous + 1 2. For CTR timecode = 6 bits of the least coarse time   In this case, the spacewire constraint is mandatory | | | | | | | | | Verify the LFR FSW synchronizes its local time with the CTR.  TM are consistent with the CP\_RPW\_TIME  MSB=0  TM\_LFR\_HK:  Field HK\_LFR\_DPU\_SPW\_TICK\_OUT incremented of 1  HK\_LFR\_DPU\_SPW\_LAST\_TMEC = time code | |  |  |
|  | For each mode, send the time-code at SY\_RPW\_CTR\_FREQUENCY during 1mn: | | | | | | | | | LFR FSW local time is incremented.  TM\_LFR\_HK:  Field HK\_LFR\_DPU\_SPW\_TICK\_OUT incremented of 1  HK\_LFR\_DPU\_SPW\_LAST\_TMEC = time code  HK\_LFR\_TIME\_TIMECODE\_CTR incremented by 1  and traced into the last error fields.  last error detected is:  HK\_LFR\_LAST\_ER\_RID: LE\_LFR\_TIME = 42119, HK\_LFR\_LAST\_ER\_CODE: TIMECODE\_CTR = 26, HK\_LFR\_LAST\_ER\_TIME  error counter  HK\_LFR\_LE\_CNT is incremented. | |  |  |
|  | For each mode, no time code of SY\_RPW\_CTR\_FREQUENCY during 1mn. | | | | | | | | | .during the one minute delay:  **LFR stays synchronize**  **MSB = 0**  TM\_LFR\_HK:  Field HK\_LFR\_DPU\_SPW\_TICK\_OUT not updated  HK\_LFR\_DPU\_SPW\_LAST\_TMEC not updated  HK\_LFR\_TIME\_TIMECODE\_CTR not updated  HK\_LFR\_TIMECODE\_MISSING is incremented and  traced into the last error fields.  last error detected: HK\_LFR\_LAST\_ER\_RID: LE\_LFR\_TIMEC = 42129, HK\_LFR\_LAST\_ER\_CODE: MISSING = 21, HK\_LFR\_LAST\_ER\_TIME  error counter HK\_LFR\_LE\_CNT is incremented  **After 1 mn LFR lost the synchronization.**  **MSB = 1**  LFR is desynchronized for the first time  HK\_LFR\_TIME\_NOT\_SYNCHRO=1  and traced into the last error fields  HK\_LFR\_LAST\_ER\_RID: LE\_LFR\_TIME = 42119, HK\_LFR\_LAST\_ER\_CODE: NOT\_SYNCHRO = 25, HK\_LFR\_LAST\_ER\_TIME  HK\_LE\_CNT is incremented. | |  |  |
|  | For each mode, send the time-code at SY\_RPW\_CTR\_FREQUENCY during 1mn | | | | | | | | | **LFR increments its local time but stays desynchronized**  **MSB = 1**  TM\_LFR\_HK:  Field HK\_LFR\_DPU\_SPW\_TICK\_OUT incremented of 1  HK\_LFR\_DPU\_SPW\_LAST\_TMEC = time code  HK\_LFR\_TIME\_TIMECODE\_IT is incremented.  2 possible errors:  HK\_LFR\_TIME\_TIMECODE\_CTR incremented by 1  and  traced into the last error fields.  HK\_LFR\_LAST\_ER\_RID: LE\_LFR\_TIME = 42119, HK\_LFR\_LAST\_ER\_CODE: TIMECODE\_CTR = 26, HK\_LFR\_LAST\_ER\_TIME=0x1xxxx  OR  HK\_LFR\_TIME\_TIMECODE\_IT is incremented by 1  and  traced into the last error fields:  HK\_LFR\_LAST\_ER\_RID: LE\_LFR\_TIME = 42119, HK\_LFR\_LAST\_ER\_CODE: TIMECODE\_IT = 24  HK\_LE\_CNT is incremented.  After 1mn, LFR always is desynchronized  LFR TIME has MSB = 1  **LFR needs a CTR couple to be synchronized** | |  |  |
|  | For each mode:  At t0: TC\_LFR\_UPDATE\_TIME with any CP\_RPW\_TIME.  At t0+SY\_RPW\_CTR\_MIN\_DELAY: Send time code at SY\_RPW\_CTR\_FREQUENCY | | | | | | | | | Verify the LFR FSW synchronizes its local time with the CTR.  MSB=0  TM\_LFR\_HK:  No more error detected  Field HK\_LFR\_DPU\_SPW\_TICK\_OUT incremented of 1  HK\_LFR\_DPU\_SPW\_LAST\_TMEC = time code  HK\_LFR\_TIME\_TIMECODE\_CTR not updated | |  |  |
|  | This step uses another the script **send\_timecode\_not\_nominal.py**  **For each mode send a not nominal sequence (timecode is sent at t0+2s)**  At t0: TC\_LFR\_UPDATE\_TIME with any CP\_RPW\_TIME.  At t0+2s: valid time code  Wait 3s (no timecode is sent)  **For each mode send a not nominal sequence (invalid timecode)**  At t0: TC\_LFR\_UPDATE\_TIME with any CP\_RPW\_TIME.  At t0+0.3s invalid time code | | | | | | | | | Verify the LFR FSW synchronizes its local time with the CTR.  TM are consistent with the CP\_RPW\_TIME.  HK\_LFR\_LFR\_TIMECODE\_MISSING is incremented of only 1.  HK\_LFR\_LE\_CNT is incremented of 1  HK\_LFR\_LAST\_ER\_CODE: MISSING = 21  Verify the LFR FSW doesn’t synchronize its local time with the CTR.    TM\_LFR\_HK:  Field HK\_LFR\_DPU\_SPW\_TICK\_OUT and  HK\_LFR\_DPU\_SPW\_LAST\_TMEC are not updated  (=5)  HK\_LFR\_LE\_CNT not updated  No error counters updated  (HK\_LFR\_TIMECODE\_MISSING=6) and the others =0 | | LFR is tolerant with SY\_RPW\_CTR\_MIN\_DELAY delay  --------------- |  |
|  | End | | | | | | | | |  | |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE** | | | | | | | | | | | | |
| **Test procedure identifier** | | | | | | | **Date** | | **Assessment** | | | |
| SVS-0013\_Ed2 | | | | | | | DD/MM/YYYY | | NT / NA / NOK / POK / OK | | | |
| **TS Tested requirements** | | | | | | | **RB Tested requirements** | | | | | |
| N/A | | | | | | | REQ-LFR-SRS-5220\_Ed2 | | | | | |
| **Component:** | | | | |  | | **Version**: V2 | |  | | | |
| **Involved subsystems** | | | | | | | | | | | | |
| SW: | 3.2.0.24 | | | | | | | | | | | |
| HW: | 1.1.91 StarDundee | | | | | | | | | | | |
| Start time: | | | (DD/MM/YYYY) hh:mm | | | | Test Operator: | | | |  | |
| End Time: | | | (DD/MM/YYYY) hh:mm | | | | Test Duration:20mn | | | |  | |
| **Purpose:** | | | | Check , upon reception of the CTR SpW time code:  -If a CTR command packet coupled has been received prior to the time code: LFR updates the coarse time part of the local time with the coarse time value previously transmitted in the CTR command packet.  -If a CTR command packet has not been received prior to the time code: LFR doesn’t ignores the time code.  -LFR sets the fine time part of its local time to 0. | | | | | | | | |
| **General remarks about the test** | | | | | | | | | | | | |
| Requirement Title | | | | | | Time management | | | | | | |
| dependencies | | | | | | SSS-CP-FS-380 | | | | | | |
| restrictions | | | | | |  | | | | | | |
| means | | | | | |  | | | | | | |
| input data | | | | | |  | | | | | | |
| prerequisite | | | | | |  | | | | | | |
| used test programs | | | | | |  | | | | | | |
| test script | | | | | | 4 scripts nominal\_tc\_lfr\_update\_time\_Step0i.py (i=1,2,3,4,5,6)  Reset before each step | | | | | | |
| output data | | | | | |  | | | | | | |
| notes | | | | | | SY\_RPW\_CTR\_FREQUENCY=1s, SY\_RPW\_CTR\_MIN\_DELAY=300ms | | | | | | |
| **Procedure steps** | | | | | | | | | | | | |
| **Step** | | **Actions** | | | | | | **Expected Results** | | **Comments** | | **Status** |
|  | | Run the LFR FSW.  For each mode:  At t0, send TC\_LFR\_UPDATE\_TIME At t1=t0+1s, send an other TC\_LFR\_UPDATE\_TIME.  At t2=t1+SY\_RPW\_CTR\_MIN\_DELAY, send a time code. | | | | | | Verify the LFR FSW, upon reception of the time code:  -updates the coarse time part of the local time with the coarse time value previously transmitted in the t2 emitted TC\_LFR\_UPDATE\_TIME.  -sets the fine time part of its local time to 0. | | **Only the last** TC\_LFR\_UPDATE\_TIME is used. | |  |
|  | | Run the LFR FSW.  For each mode, for several delays:  At t0, send TC\_LFR\_UPDATE\_TIME At t1=t0+delay, send a time code. | | | | | | -At each time code, the FSW LFR sets the fine time part of its local time to 0.  Reception time code > SY\_RPW\_CTR\_FREQUENCY  Coarse time incremented of 1. | | LFR is tolerant with SY\_RPW\_CTR\_MIN\_DELAY delay | |  |
|  | | Run the LFR FSW.  For each mode, for several periods T:  At ti, send TC\_LFR\_UPDATE\_TIME.  At t0=ti+ SY\_RPW\_CTR\_MIN\_DELAY, send a time code.  At tn=t0+ n\*T, send a time code during 20s (n is a non null integer). | | | | | | Verify the LFR FSW, at each reception of time code:  -At t0 only, LFR updates the coarse time part of the local time with the coarse time value previously transmitted in the TC\_LFR\_UPDATE\_TIME.  -LFR sets the fine time part of its local time to 0. | |  | |  |
|  | | Run the LFR FSW.  For each mode, for several periods T:  At t0=n\*T send TC\_LFR\_UPDATE\_TIME  At t1=t0+ SY\_RPW\_CTR\_MIN\_DELAY, send a time code.  At tn=t0+ n\*T, send a time code during 1mn (n is an non null integer). | | | | | | Verify the LFR FSW, at each reception of time code:  -LFR updates the coarse time part of the local time with the coarse time value previously transmitted in the TC\_LFR\_UPDATE\_TIME.  -LFR sets the fine time part of its local time to 0. | |  | |  |
|  | | Run the LFR FSW. | | | | | |  | |  | |  |
|  | | Run the LFR FSW.  automatic timecode is launched  For each mode send  four TC\_UPDATE\_TIME | | | | | |  | |  | |  |
|  | | End | | | | | |  | |  | |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE** | | | | | | | | | | | | |
| **Test procedure identifier** | | | | | **Date** | | | | | **Assessment** | | |
| SVS-0014\_Ed1 | | | | | DD/MM/YYYY | | | | | NT / NA / NOK / POK / OK | | |
| **TS Tested requirements** | | | | | **RB Tested requirements** | | | | | | | |
| N/A | | | | | REQ-LFR-SRS-5238\_Ed1 | | | | | | | |
| **Component:** | | | |  | **Version:** V2 | |  | | | | | |
| **Involved subsystems** | | | | | | | | | | | | |
| SW: | 3.2.0.24 | | | | | | | | | | | |
| HW: | 1.1.91 StarDundee | | | | | | | | | | | |
| Start time: | | | (DD/MM/YYYY) hh:mm | | Test Operator: | | | |  | | | |
| End Time: | | | (DD/MM/YYYY) hh:mm | | Test Duration:1mn14s | | |  | | | | |
| **Purpose:** | | | Check, on successful time synchronization, the LFR FSW sets the most significant bit of its local time to 0. | | | | | | | | | |
| **General remarks about the test** | | | | | | | | | | | | |
| Requirement Title | | | | Time management | | | | | | | | |
| dependencies | | | | SSS-CP-FS-405, AD06 R-788 | | | | | | | | |
| restrictions | | | |  | | | | | | | | |
| means | | | |  | | | | | | | | |
| input data | | | |  | | | | | | | | |
| prerequisite | | | |  | | | | | | | | |
| used test programs | | | |  | | | | | | | | |
| test script | | | | time\_msb.py | | | | | | | | |
| output data | | | |  | | | | | | | | |
| notes | | | |  | | | | | | | | |
| **Procedure steps** | | | | | | | | | | | | |
| **Step** | | **Actions** | | | | **Expected Results** | | | | | **Comments** | **Status** |
|  | | For each mode  Send nominal TC\_LFR\_UPDATE\_TIME Packet (CP\_RPW\_TIME with MSB=0  + send time code | | | | Verify the LFR FSW sets the most significant bit of its local time to 0. | | | | |  |  |
|  | | Send TC\_LFR\_UPDATE\_TIME Packet with CP\_RPW\_TIME= 0x876543210000 + send time code  Wait at least 3s. | | | | Verify the LFR FSW sets the most significant bit of its local time to 0. | | | | |  |  |
|  | | Send TC\_LFR\_UPDATE\_TIME Packet with CP\_RPW\_TIME=0x7FFFFFFF0000 + send time code  Wait at least 3s. | | | | Verify the LFR FSW sets the most significant bit of its local time to 0. | | | | |  |  |
|  | | End | | | |  | | | | |  |  |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE** | | | | | | | | | | |
| **Test procedure identifier** | | | | | **Date** | | | **Assessment** | | |
| SVS-0018\_Ed1 | | | | | DD/MM/YYYY | | | NT / NA / NOK / POK / OK | | |
| **TS Tested requirements** | | | | | **RB Tested requirements** | | | | | |
| N/A | | | | | REQ-LFR-SRS-5239\_Ed2 | | | | | |
| **Component:** | | | |  | **Version:** | V1 | | | | |
| **Involved subsystems** | | | | | | | | | | |
| SW: | 3.2.0.24 | | | | | | | | | |
| HW: | 1.1.91 Stardundee | | | | | | | | | |
| Start time: | | | (DD/MM/YYYY) hh:mm | | Test Operator: | | |  | | |
| End Time: | | | (DD/MM/YYYY) hh:mm | | Test Duration: | | | 15Mn + 6h15 | | |
| **Purpose:** | | | Check the LFR FSW sets the Destination ID field by complying with the rules when building a TM packet. | | | | | | | |
| **General remarks about the test** | | | | | | | | | | |
| Requirement Title | | | | Telemetry management | | | | | | |
| dependencies | | | | SSS-CP-FS-580 | | | | | | |
| restrictions | | | |  | | | | | | |
| means | | | |  | | | | | | |
| input data | | | |  | | | | | | |
| prerequisite | | | |  | | | | | | |
| used test programs | | | |  | | | | | | |
| test script | | | | source\_id\_loop\_step1.py ( step1 to step 3) and source\_id\_loop\_step2\_xxx.py (step 4 to 7)  one script by mode (normal,burst,sbm1,sbm2)  ASM=4s | | | | | | |
| output data | | | |  | | | | | | |
| notes | | | | SOURCE\_ID values defined are the enum of  No ack for TC\_LFR\_UPDATE\_INFO/TC\_LFR\_UPDATE\_TIME | | | | | | |
| **Procedure steps** | | | | | | | | | | |
| **Step** | | **Actions** | | | | | **Expected Results** | | **Comments** | **Status** |
|  | | LFR FSW builds TM packet out exception: TM packet having the packet category <> 2 (HK essential), <> 3 (Table), <> 4 (HK routine), <> 8 (diagnostic) and <> 9 (dump).  Send TC\_LFR\_LOAD\_COMMON\_PAR, TC\_LFR\_LOAD\_NORMAL\_PAR, TC\_LFR\_LOAD\_BURST\_PAR, TC\_LFR\_LOAD\_SBM1\_PAR, TC\_LFR\_LOAD\_SBM2\_PAR, TC\_LFR\_ENABLE\_CALIBRATION, TC\_LFR\_DISABLE\_CALIBRATION,  TC\_LFR\_LOAD\_KCOEFFICIENTS,  TC\_LFR\_FBINS\_MASK,  TC\_LFR\_LOAD\_FILTER\_PAR,  TC\_LFR\_ENTER\_MODE (with mode transition).  For each TC, send all the SOURCE\_ID values allowed (0, …, 255). | | | | | Verify for telemetry generated as an answer to a command, the Destination ID is:  -the copy of the command Source ID field for defined cases (using RD06)  -0 for undefined Id. | |  |  |
|  | | LFR FSW builds a TM packet having the packet category = 4. Wait a TM\_LFR\_HK. | | | | | Verify the Destination ID always is set to zero. | |  |  |
|  | | LFR FSW builds a TM packet having the packet category = 6  Send TC\_LFR\_DUMP\_PAR,  TC\_LFR\_DUMP\_KCOEFF,  For each TC, send all the SOURCE\_ID values allowed (0, …, 255). | | | | | Verify for telemetry generated as an answer to a command, the Destination ID is:  -the copy of the command Source ID field for defined cases (using RD06)  -0 for undefined Id. | |  |  |
|  | | For each TC, send all the SOURCE\_ID values defined+ one undefined value:  -Enter in NORMAL mode. Wait the all TM\_LFR\_SCIENCE\_NORMAL\_\*.  ( CWF\_F3 and CWF\_LONG\_F3) | | | | | Verify the Destination ID always is set to zero. | |  |  |
|  | | For each TC, send all the SOURCE\_ID values defined+ one undefined value:  -Enter in BURST mode. Wait all TM\_LFR\_SCIENCE\_BURST\_\*. | | | | | Verify the Destination ID always is set to zero. | |  |  |
|  | | For each TC, send all the SOURCE\_ID values defined+ one undefined value:  -Enter in SBM1 mode. Wait all  TM\_LFR\_SCIENCE\_SBM1\_\*.  ( NORMAL\_CWF\_F3 and NORMAL\_CWF\_LONG\_F3) | | | | | Verify the Destination ID always is set to zero. | |  |  |
|  | | For each TC, send all the SOURCE\_ID values defined+ one undefined value:  -Enter in SBM2 mode. Wait all  TM\_LFR\_SCIENCE\_SBM2\_\*.  ( NORMAL\_CWF\_F3 and NORMAL\_CWF\_LONG\_F3) | | | | | Verify the Destination ID always is set to zero. | |  |  |
|  | | End | | | | |  | |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE** | | | | | | | | | | | |
| **Test procedure identifier** | | | | | **Date** | | | | **Assessment** | | |
| SVS-0019\_Ed1 | | | | | DD/MM/YYYY | | | | NT / NA / NOK / POK / OK | | |
| **TS Tested requirements** | | | | | **RB Tested requirements** | | | | | | |
| N/A | | | | | REQ-LFR-SRS-5240\_Ed2 REQ-LFR-SRS-5241\_Ed1 | | | | | | |
| **Component:** | | | |  | **Version:** | | V2 | | | | |
| **Involved subsystems** | | | | | | | | | | | |
| SW: | 3.2.0.24 | | | | | | | | | | |
| HW: | 1.1.91 StarDundee | | | | | | | | | | |
| Start time: | | | (DD/MM/YYYY) hh:mm | | Test Operator: | | |  | | | |
| End Time: | | | (DD/MM/YYYY) hh:mm | | Test Duration: 9h | | |  | | | |
| **Purpose:** | | | Check the LFR FSW maintains, for each couple of APID and Destination ID, a TM sequence counter incremented when a packet is released. | | | | | | | | |
| **General remarks about the test** | | | | | | | | | | | |
| Requirement Title | | | | Telemetry management | | | | | | | |
| dependencies | | | | SSS-CP-FS-590 and SSS-CP-FS-201 | | | | | | | |
| restrictions | | | |  | | | | | | | |
| means | | | |  | | | | | | | |
| input data | | | |  | | | | | | | |
| prerequisite | | | |  | | | | | | | |
| used test programs | | | |  | | | | | | | |
| test script | | | | tm\_sequence\_counter\_loop.py (step 1 -8)  tm\_sequence\_counter\_hk.py (step 9) tm\_sequence\_counter\_dump\_loop.py (step 10) tm\_sequence\_science\_loop.py (step 11)  tm\_sequence\_counter\_loop\_acceptance\_failed.py (step12) | | | | | | | |
| output data | | | |  | | | | | | | |
| notes | | | |  | | | | | | | |
| **Procedure steps** | | | | | | | | | | | |
| **Step** | | **Actions** | | | | **Expected Results** | | | | **Comments** | **Status** |
|  | | For the TM\_LFR\_TC\_EXE\_\*:  For each SOURCE\_ID defined, send (to receive a TM\_LFR\_TC\_EXE\_SUCCESS):  TC\_LFR\_LOAD\_COMMON\_PAR, TC\_LFR\_LOAD\_NORMAL\_PAR, TC\_LFR\_LOAD\_BURST\_PAR, TC\_LFR\_LOAD\_SBM1\_PAR, TC\_LFR\_LOAD\_SBM2\_PAR, TC\_LFR\_ENTER\_MODE, TC\_LFR\_RESET, TC\_LFR\_ENABLE\_CALIBRATION, TC\_LFR\_DISABLE\_CALIBRATION,  TC\_LFR\_LOAD\_KCOEFFICIENTS,  TC\_LFR\_DUMP\_KCOEFICIENTS,  TC\_LFR\_FBINS\_MASK,  TC\_LFR\_DUMP\_PAR,  TC\_LFR\_LOAD\_FILTER\_PAR  Alter each TC to cause -if possible- the answer TM\_LFR\_TC\_EXE\_SUCCESS, TM\_LFR\_TC\_EXE\_INCONSISTENT, TM\_LFR\_TC\_EXE\_NOT\_EXECUTABLE, TM\_LFR\_TC\_EXE\_CORRUPTED.  Ensure the verification packets: PID=76, CAT=1.  For TC\_LFR\_DUMP\_KCOEFICIENTS and  TC\_LFR\_DUMP\_PAR  we wait  TM\_LFR\_PARAMETER\_DUMP.  TM\_LFR\_KCOEFFICIENTS\_DUMP  Ensure the verification packets: PID=76, CAT=6. | | | | Verify the LFR FSW maintains, for each couple of APID and Destination ID, a TM sequence counter incremented by 1 when a packet is released and the  FIRST SEQUENCE\_CNT=0 | | | |  |  |
|  | | For each SOURCE\_ID defined, send:  TC\_LFR\_RESET case CORRUPTED,  TC\_LFR\_LOAD\_COMMON\_PAR case CORRUPTED,  TC\_LFR\_LOAD\_NORMAL\_PAR cases CORRUPTED, NOT\_EXECUTABLE,  TC\_LFR\_LOAD\_BURST\_PAR cases CORRUPTED, NOT\_EXECUTABLE,  TC\_LFR\_LOAD\_SBM1\_PAR cases CORRUPTED, NOT\_EXECUTABLE,  TC\_LFR\_LOAD\_SBM2\_PAR cases NOT\_EXECUTABLE,  TC\_LFR\_ENTER\_MODE cases CORRUPTED, INCONSISTENT and NOT\_EXECUTABLE,  TC\_LFR\_ENABLE\_CALIBRATION case CORRUPTED,  TC\_LFR\_DISABLE\_CALIBRATION case CORRUPTED.  TC\_LFR\_LOAD\_KCOEFFICIENTS case CORRUPTED  TC\_LFR\_DUMP\_KCOEFICIENTS case CORRUPTED  TC\_LFR\_FBINS\_MASK case CORRUPTED  TC\_LFR\_DUMP\_PAR case CORRUPTED  TC\_LFR\_LOAD\_FILTER\_PAR case CORRUPTED  Ensure the verification packets: PID=76, CAT=1. | | | | Verify the LFR FSW maintains, for each couple of APID and Destination ID, a TM sequence counter incremented by 1 when a packet is released and the  FIRST SEQUENCE\_CNT=0. | | | |  |  |
|  | | In TM\_LFR\_HK, ensure the verification packets: PID=76, CAT=4. | | | | Verify the LFR FSW maintains, for each couple of APID and Destination ID, a TM sequence counter incremented by 1 when a packet is released and the  FIRST SEQUENCE\_CNT=0. | | | |  |  |
|  | | SWF\_P =22s  Enter in NORMAL mode. Wait TM\_LFR\_SCIENCE\_NORMAL\_SWF\_F0, TM\_LFR\_SCIENCE\_NORMAL\_SWF\_F1, TM\_LFR\_SCIENCE\_NORMAL\_SWF\_F2, TM\_LFR\_SCIENCE\_NORMAL\_CWF\_F3, TM\_LFR\_SCIENCE\_NORMAL\_ASM\_F0, TM\_LFR\_SCIENCE\_NORMAL\_ASM\_F1, TM\_LFR\_SCIENCE\_NORMAL\_ASM\_F2, TM\_LFR\_SCIENCE\_NORMAL\_BP1\_F0, TM\_LFR\_SCIENCE\_NORMAL\_BP1\_F1, TM\_LFR\_SCIENCE\_NORMAL\_BP1\_F2, TM\_LFR\_SCIENCE\_NORMAL\_BP2\_F0, TM\_LFR\_SCIENCE\_NORMAL\_BP2\_F1, TM\_LFR\_SCIENCE\_NORMAL\_BP2\_F2 (note durations are in TC\_LFR\_LOAD\_NORMAL\_PAR).  Ensure the verification packets: PID=76, CAT=12. | | | | Verify the LFR FSW maintains, for each couple of APID and Destination ID, a TM sequence counter incremented by 1 when a packet is released and the  FIRST SEQUENCE\_CNT=0. | | | |  |  |
|  | | Enter in BURST mode. Wait TM\_LFR\_SCIENCE\_BURST\_CWF\_F2, TM\_LFR\_SCIENCE\_BURST\_BP1\_F0, TM\_LFR\_SCIENCE\_BURST\_BP2\_F0, TM\_LFR\_SCIENCE\_BURST\_BP1\_F1, TM\_LFR\_SCIENCE\_BURST\_BP2\_F1 (note durations are in TC\_LFR\_LOAD\_BURST\_PAR).  Ensure the verification packets: PID=76, CAT=12. | | | | Verify the LFR FSW maintains, for each couple of APID and Destination ID, a TM sequence counter incremented by 1 when a packet is released and the  FIRST SEQUENCE\_CNT=0. | | | |  |  |
|  | | Enter in SBM1 mode. Wait TM\_LFR\_SCIENCE\_SBM1\_CWF\_F1, TM\_LFR\_SCIENCE\_SBM1\_BP1\_F0, TM\_LFR\_SCIENCE\_SBM1\_BP2\_F0 (note durations are in TC\_LFR\_LOAD\_SBM1\_PAR).  Ensure the verification packets: PID=79, CAT=12. | | | | Verify the LFR FSW maintains, for each couple of APID and Destination ID, a TM sequence counter incremented by 1 when a packet is released and the  FIRST SEQUENCE\_CNT=0. | | | |  |  |
|  | | Enter in SBM2 mode. Wait TM\_LFR\_SCIENCE\_SBM2\_CWF\_F2, TM\_LFR\_SCIENCE\_SBM2\_BP1\_F0, TM\_LFR\_SCIENCE\_SBM2\_BP2\_F0, TM\_LFR\_SCIENCE\_SBM2\_BP1\_F1, TM\_LFR\_SCIENCE\_SBM2\_BP2\_F1 (note durations are in TC\_LFR\_LOAD\_SBM2\_PAR).  Ensure the verification packets: PID=79, CAT=12. | | | | Verify the LFR FSW maintains, for each couple of APID and Destination ID, a TM sequence counter incremented by 1 when a packet is released and the  FIRST SEQUENCE\_CNT=0. | | | |  |  |
|  | | Packet 0xcfc (SBM1 and SBM2) valid the SEQUENCE\_CNTcounter reaches (2^14)-1. | | | | Verify the LFR stays healthy, and the counter wraps around from 2^14-1 to zero. | | | | REQ-LFR-SRS-5241\_Ed1 |  |
|  | | Packet 0xcc4 (HK) valid the SEQUENCE\_CNTcounter reaches (2^14)-1. | | | | Verify the LFR stays healthy, and the counter wraps around from 2^14-1 to zero. | | | | REQ-LFR-SRS-5241\_Ed1 |  |
|  | | Packet 0xcc1 (TM\_LFR\_TC\_EXE\_SUCCESS) and  Packet 0xcc6 (TM\_LFR\_PARAMETER\_DUMP TM\_LFR\_KCOEFFICIENTS\_DUMP)  valid the SEQUENCE\_CNTcounter reaches (2^14)-1. | | | | Verify the LFR stays healthy, and the counter wraps around from 2^14-1 to zero. | | | | REQ-LFR-SRS-5241\_Ed1 |  |
|  | | Packet 0xccc valid the SEQUENCE\_CNTcounter reaches (2^14)-1. | | | | Verify the LFR stays healthy, and the counter wraps around from 2^14-1 to zero. | | | | REQ-LFR-SRS-5241\_Ed1 |  |
|  | | Packet 0xcc1 ((TM\_LFR\_TC\_EXE \_CORRUPTED NOT\_EXECUTABLE) valid the SEQUENCE\_CNTcounter reaches (2^14)-1. | | | | Verify the LFR stays healthy, and the counter wraps around from 2^14-1 to zero. | | | | REQ-LFR-SRS-5241\_Ed1 |  |
|  | | End | | | |  | | | |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE** | | | | | | | | | | | | |
| **Test procedure identifier** | | | | | | **Date** | | | | **Assessment** | | |
| SVS-0020\_Ed2 | | | | | | DD/MM/YYYY | | | | NT / NA / NOK / POK / OK | | |
| **TS Tested requirements** | | | | | | **RB Tested requirements** | | | | | | |
| N/A | | | | | | REQ-LFR-SRS-5300\_Ed2 | | | | | | |
| **Component:** | | | | |  | **Version:** | | V2 | | | | |
| **Involved subsystems** | | | | | | | | | | | | |
| SW: | | 3.2.0.24 | | | | | | | | | | |
| HW: | | 1.1.91 StarDundee | | | | | | | | | | |
| Start time: | | | (DD/MM/YYYY) hh:mm | | | Test Operator: | | |  | | | |
| End Time: | | | (DD/MM/YYYY) hh:mm | | | Test Duration: | | | 34mn + 3610s | | | |
| **Purpose:** | | | Check the LFR FSW maintains a local time, with:  - a resolution of at least SY\_RPW\_TIME\_RESOLUTION.  -a relative accuracy of SY\_RPW\_TIME\_ACCURACY. | | | | | | | | | |
| **General remarks about the test** | | | | | | | | | | | | |
| Requirement Title | | | | Time management | | | | | | | | |
| dependencies | | | | SSS-CP-FS-340 | | | | | | | | |
| restrictions | | | | This verification could be thorough: -Rigorous verification is possible only with calibrated external equipments. -Functional verification is possible with all instruments (validation at system level).  -A long delay avoids distortion measurement (the CPU load variation are not visible), but PC resynchronizations via the network, and delays due to treatments by the brick are possible. | | | | | | | | |
| means | | | |  | | | | | | | | |
| input data | | | |  | | | | | | | | |
| prerequisite | | | |  | | | | | | | | |
| used test programs | | | |  | | | | | | | | |
| test script | | | | time\_resolution.py + set\_time\_wrap.py | | | | | | | | |
| output data | | | |  | | | | | | | | |
| notes | | | | The PC clock is the reference for this test (a log delay avoid to fail the mesure).  Numerical values are in AD03 (currently:SY\_RPW\_TIME\_RESOLUTION = 1 ms, SY\_RPW\_TIME\_ACCURACY = 500 us).  The verif\_internal\_time\_consistence tool (cf verif\_fields.py) feature surveys the consistence between the pc time, and the lfr time. | | | | | | | | |
| **Procedure steps** | | | | | | | | | | | | |
| **Step** | **Actions** | | | | | | **Expected Results** | | | | **Comments** | **Status** |
|  | Change some NORMAL parameters  NORMAL MODE  SWF=22s, ASM=4s,  Wait at least 500s in each mode without TC\_LFR\_UPDATE\_TIME and without time code reception by LFR. | | | | | | Using TIME field of TM\_LFR\_\* packets, verify the LFR FSW maintains a local time with a resolution relative accuracy of SY\_RPW\_TIME\_ACCURACY (compare the delays: from the PC clock, and from the internal time. Using the accumulated deviation, calculate the average shift per second). | | | |  |  |
|  | Launch LFR FSW  Reach NORMAL mode  Send a TC\_LFR\_UPDATE\_TIME with  CP\_RPW\_TIME=0x7FFFFE01 + valid time code  Wait 3600s | | | | | | the TIME field wraps around 0xFFFFFFFF to 0x80000000  MSB= 1 because we lost the synchronization (no timecode sent during the test)  See on TM\_LFR\_HK, all TM\_LFR\_SCIENCE\_NORMAL products are consistent. | | | |  |  |
|  | End | | | | | |  | | | |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE** | | | | | | | | | | | | | | | |
| **Test procedure identifier** | | | | | | | **Date** | | | | **Assessment** | | | | |
| SVS-0021\_Ed2 | | | | | | | DD/MM/YYYY | | | | NT / NA / NOK / POK / OK | | | | |
| **TS Tested requirements** | | | | | | | **RB Tested requirements** | | | | | | | | |
| N/A | | | | | | | REQ-LFR-SRS-5302\_Ed1 | | | | | | | | |
| **Component:** | | | | | | | | |  | **Version:** V2 | | |  | | |
| **Involved subsystems** | | | | | | | | | | | | | | | |
| SW: 3.2.0.24 | | | | |  | | | | | | | | | | |
| HW:1.1.91 StarDundee | | | | |  | | | | | | | | | | |
| Start time: | | | | (DD/MM/YYYY) hh:mm | | | | Test Operator: | | | | | |  | |
| End Time: | | | | (DD/MM/YYYY) hh:mm | | | | Test Duration:3mn10 | | | | |  | | |
| **Purpose:** | | Check the LFR FSW generates the TC execution report not longer than SY\_RPW\_ACK\_RESPONSE\_TIME from the completion of the TC execution. | | | | | | | | | | | | | |
| **General remarks about the test** | | | | | | | | | | | | | | | |
| Requirement Title | | | Response time | | | | | | | | | | | | |
| dependencies | | | SSS-CP-FS-131, AD10: 5-TCV-8 | | | | | | | | | | | | |
| restrictions | | |  | | | | | | | | | | | | |
| means | | |  | | | | | | | | | | | | |
| input data | | |  | | | | | | | | | | | | |
| prerequisite | | | SVS-0009 | | | | | | | | | | | | |
| used test programs | | |  | | | | | | | | | | | | |
| test script | | | tm\_delay.py + just\_tc\_lfr\_reset.py | | | | | | | | | | | | |
| output data | | |  | | | | | | | | | | | | |
| notes | | | No ack for TC\_LFR\_UPDATE\_INFO/TC\_LFR\_UPDATE\_TIME  NB\*.txt synthesizes whenever the timeout is not reached. | | | | | | | | | | | | |
| **Procedure steps** | | | | | | | | | | | | | | | |
| **Step** | **Actions** | | | | | **Expected Results** | | | | | | **Comments** | | | **Status** |
|  | Use just\_tc\_lfr\_reset.py  Generate nominal TC\_LFR\_RESET. | | | | | Exit of LFR FSW.  NO TM\_LFR\_TC\_EXE\* after time-out (4s) | | | | | |  | | |  |
|  | Generate nominal TC\_LFR\_LOAD\_COMMON\_PAR. | | | | | Verify the LFR FSW generates one TC execution report (success) not longer than SY\_RPW\_ACK\_RESPONSE\_TIME from the completion of the TC execution. | | | | | |  | | |  |
|  | Generate nominal TC\_LFR\_LOAD\_NORMAL\_PAR. | | | | | Verify the LFR FSW generates one TC execution report (success) not longer than SY\_RPW\_ACK\_RESPONSE\_TIME from the completion of the TC execution. | | | | | |  | | |  |
|  | Generate nominal TC\_LFR\_LOAD\_BURST\_PAR. | | | | | Verify the LFR FSW generates one TC execution report (success) not longer than SY\_RPW\_ACK\_RESPONSE\_TIME from the completion of the TC execution. | | | | | |  | | |  |
|  | Generate nominal TC\_LFR\_LOAD\_SBM1\_PAR. | | | | | Verify the LFR FSW generates one TC execution report (success) not longer than SY\_RPW\_ACK\_RESPONSE\_TIME from the completion of the TC execution. | | | | | |  | | |  |
|  | Generate nominal TC\_LFR\_LOAD\_SBM2\_PAR. | | | | | Verify the LFR FSW generates one TC execution report (success) not longer than SY\_RPW\_ACK\_RESPONSE\_TIME from the completion of the TC execution. | | | | | |  | | |  |
|  | Generate nominal TC\_LFR\_DUMP\_PAR. | | | | | Verify the LFR FSW generates one TC execution report (success) not longer than SY\_RPW\_ACK\_RESPONSE\_TIME from the completion of the TC execution. | | | | | |  | | |  |
|  | Generate nominal TC\_LFR\_LOAD\_KCOEFFICIENTS | | | | | Verify the LFR FSW generates one TC execution report (success) not longer than SY\_RPW\_ACK\_RESPONSE\_TIME from the completion of the TC execution. | | | | | |  | | |  |
|  | Generate nominal TC\_LFR\_DUMP\_KCOEFFICIENTS | | | | | Verify the LFR FSW generates one TC execution report (success) not longer than SY\_RPW\_ACK\_RESPONSE\_TIME from the completion of the TC execution | | | | | |  | | |  |
|  | Generate nominal TC\_LFR\_LOAD\_FBINS\_MASK | | | | | Verify the LFR FSW generates one TC execution report (success) not longer than SY\_RPW\_ACK\_RESPONSE\_TIME from the completion of the TC execution. | | | | | |  | | |  |
|  | Generate nominal TC\_LFR\_LOAD\_FILTER\_PAR | | | | | Verify the LFR FSW generates one TC execution report (success) not longer than SY\_RPW\_ACK\_RESPONSE\_TIME from the completion of the TC execution. | | | | | |  | | |  |
|  | Generate nominal TC\_LFR\_ENTER\_MODE. | | | | | Verify the LFR FSW generates one TC execution report (success) not longer than SY\_RPW\_ACK\_RESPONSE\_TIME from the completion of the TC execution. | | | | | |  | | |  |
|  | Generate nominal TC\_LFR\_ENABLE\_CALIBRATION. | | | | | Verify the LFR FSW generates one TC execution report (success) not longer than SY\_RPW\_ACK\_RESPONSE\_TIME from the completion of the TC execution. | | | | | |  | | |  |
|  | Generate nominal TC\_LFR\_DISABLE\_CALIBRATION. | | | | | Verify the LFR FSW generates one TC execution report (success) not longer than SY\_RPW\_ACK\_RESPONSE\_TIME from the completion of the TC execution. | | | | | |  | | |  |
|  | Generate failed  TC\_LFR\_RESET. | | | | | Verify the LFR FSW generates one TC execution report (failure) not longer than SY\_RPW\_ACK\_RESPONSE\_TIME from the completion of the TC execution. | | | | | |  | | |  |
|  | Generate failed TC\_LFR\_LOAD\_COMMON\_PAR. | | | | | Verify the LFR FSW generates one TC execution report (failure) not longer than SY\_RPW\_ACK\_RESPONSE\_TIME from the completion of the TC execution. | | | | | |  | | |  |
|  | Generate failed TC\_LFR\_LOAD\_NORMAL\_PAR. | | | | | Verify the LFR FSW generates one TC execution report (failure) not longer than SY\_RPW\_ACK\_RESPONSE\_TIME from the completion of the TC execution. | | | | | |  | | |  |
|  | Generate failed TC\_LFR\_LOAD\_BURST\_PAR. | | | | | Verify the LFR FSW generates one TC execution report (failure) not longer than SY\_RPW\_ACK\_RESPONSE\_TIME from the completion of the TC execution. | | | | | |  | | |  |
|  | Generate failed TC\_LFR\_LOAD\_SBM1\_PAR. | | | | | Verify the LFR FSW generates one TC execution report (failure) not longer than SY\_RPW\_ACK\_RESPONSE\_TIME from the completion of the TC execution. | | | | | |  | | |  |
|  | Generate failed TC\_LFR\_LOAD\_SBM2\_PAR. | | | | | Verify the LFR FSW generates one TC execution report (failure) not longer than SY\_RPW\_ACK\_RESPONSE\_TIME from the completion of the TC execution. | | | | | |  | | |  |
|  | Generate failed  TC\_LFR\_DUMP\_PAR. | | | | | Verify the LFR FSW generates one TC execution report (failure) not longer than SY\_RPW\_ACK\_RESPONSE\_TIME from the completion of the TC execution. | | | | | |  | | |  |
|  | Generate failed TC\_LFR\_LOAD\_KCOEFFICIENTS | | | | | Verify the LFR FSW generates one TC execution report (failure) not longer than SY\_RPW\_ACK\_RESPONSE\_TIME from the completion of the TC execution. | | | | | |  | | |  |
|  | Generate failed TC\_LFR\_DUMP\_KCOEFFICIENTS | | | | | Verify the LFR FSW generates one TC execution report (failure) not longer than SY\_RPW\_ACK\_RESPONSE\_TIME from the completion of the TC execution. | | | | | |  | | |  |
|  | Generate failed  TC\_LFR\_LOAD\_FBINS\_MASK | | | | | Verify the LFR FSW generates one TC execution report (failure) not longer than SY\_RPW\_ACK\_RESPONSE\_TIME from the completion of the TC execution. | | | | | |  | | |  |
|  | Generate failed  TC\_LFR\_LOAD\_FILTER\_PAR | | | | | Verify the LFR FSW generates one TC execution report (failure) not longer than SY\_RPW\_ACK\_RESPONSE\_TIME from the completion of the TC execution. | | | | | |  | | |  |
|  | Generate failed TC\_LFR\_ENTER\_MODE. | | | | | Verify the LFR FSW generates one TC execution report (failure) not longer than SY\_RPW\_ACK\_RESPONSE\_TIME from the completion of the TC execution. | | | | | |  | | |  |
|  | Generate failed TC\_LFR\_ENABLE\_CALIBRATION. | | | | | Verify the LFR FSW generates one TC execution report (failure) not longer than SY\_RPW\_ACK\_RESPONSE\_TIME from the completion of the TC execution. | | | | | |  | | |  |
|  | Generate failed TC\_LFR\_DISABLE\_CALIBRATION. | | | | | Verify the LFR FSW generates one TC execution report (failure) not longer than SY\_RPW\_ACK\_RESPONSE\_TIME from the completion of the TC execution. | | | | | |  | | |  |
|  | Ensure each ack has been provided; so provoke the TM\_LFR\_TC\_EXE\_\* not seen at the previous step. | | | | | Verify all the TM\_LFR\_TC\_EXE\_\* are seen for this TEST CASE. | | | | | |  | | |  |
|  | End | | | | |  | | | | | |  | | |  |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE** | | | | | | | | | | | |
| **Test procedure identifier** | | | | | **Date** | | | **Assessment** | | | |
| SVS-0022\_Ed2 | | | | | DD/MM/YYYY | | | NT / NA / NOK / POK / OK | | | |
| **TS Tested requirements** | | | | | **RB Tested requirements** | | | | | | |
| N/A | | | | | REQ-LFR-SRS-5303\_Ed1 | | | | | | |
| **Component:** | | | |  | **Version:** | | V2 | | | | |
| **Involved subsystems** | | | | | | | | | | | |
| SW: | | 3.2.0.24 | | | | | | | | | |
| HW: | | 1.1.91 StarDundee | | | | | | | | | |
| Start time: | | | (DD/MM/YYYY) hh:mm | | Test Operator: | | | |  | | |
| End Time: | | | (DD/MM/YYYY) hh:mm | | Test Duration:52mn | | | |  | | |
| **Purpose:** | | | Check the maximum rate of command packets generated to the LFR is SY\_LFR\_TC\_MAX\_RATE. | | | | | | | | |
| **General remarks about the test** | | | | | | | | | | | |
| Requirement Title | | | | Command packets (DPU SW to RPW Analyzer SW) | | | | | | | |
| dependencies | | | | SSS-IF-DPS-EQ-190 | | | | | | | |
| restrictions | | | | Verification is made only once. It is supposed to be sufficient.  We suppose that there is no interaction between the all equipments on the network. | | | | | | | |
| means | | | |  | | | | | | | |
| input data | | | |  | | | | | | | |
| prerequisite | | | |  | | | | | | | |
| used test programs | | | |  | | | | | | | |
| test script | | | | tc\_maximum\_rate.py | | | | | | | |
| output data | | | |  | | | | | | | |
| notes | | | | AD11: SY\_LFR\_TC\_MAX\_RATE = 5 commands per second. | | | | | | | |
| **Procedure steps** | | | | | | | | | | | |
| **Step** | **Actions** | | | | | **Expected Results** | | | | **Comments** | **Status** |
| 1 | LFR is in SBM1.  Send 1000 TC\_LFR\_LOAD\_COMMON\_PAR at the SY\_LFR\_TC\_MAX\_RATE rate. | | | | | Verify the LFR FSW is healthy. | | | |  |  |
| 2 | LFR is in BURST.  Send 1000 TC\_LFR\_LOAD\_NORMAL\_PAR at the SY\_LFR\_TC\_MAX\_RATE rate. | | | | | Verify the LFR FSW is healthy. | | | |  |  |
| 3 | LFR is in SBM1.  Send 1000 TC\_LFR\_LOAD\_BURST\_PAR at the SY\_LFR\_TC\_MAX\_RATE rate . | | | | | Verify the LFR FSW is healthy. | | | |  |  |
| 4 | LFR is in BURST.  Send 1000 TC\_LFR\_LOAD\_SBM1\_PAR at the SY\_LFR\_TC\_MAX\_RATE rate . | | | | | Verify the LFR FSW is healthy. | | | |  |  |
| 5 | LFR is in SBM1.  Send 1000 TC\_LFR\_LOAD\_SBM2\_PAR at the SY\_LFR\_TC\_MAX\_RATE rate . | | | | | Verify the LFR FSW is healthy. | | | |  |  |
| 6 | LFR is in SBM1.  Send 1000 TC\_LFR\_DUMP\_PAR at the SY\_LFR\_TC\_MAX\_RATE rate. | | | | | Verify the LFR FSW is healthy. | | | |  |  |
| 7 | Send 1000 TC\_LFR\_ENTER\_MODE at the SY\_LFR\_TC\_MAX\_RATE rate.  Change the CP\_LFR\_ENTER\_MODE\_TIME field for to avoid rejected TCs. | | | | | Verify the LFR FSW is healthy. | | | |  |  |
| 8 | LFR is in SBM1.  Send 1000 TC\_LFR\_UPDATE\_INFO at the SY\_LFR\_TC\_MAX\_RATE rate. | | | | | Verify the LFR FSW is healthy. | | | |  |  |
| 9 | LFR is in SBM1.  Send 1000 TC\_LFR\_ENABLE\_CALIBRATION at the SY\_LFR\_TC\_MAX\_RATE rate. | | | | | Verify the LFR FSW is healthy. | | | |  |  |
| 10 | LFR is in SBM1.  Send 1000 TC\_LFR\_DISABLE\_CALIBRATION at the SY\_LFR\_TC\_MAX\_RATE rate. | | | | | Verify the LFR FSW is healthy. | | | |  |  |
| 11 | Send 1000 TC\_LFR\_UPDATE\_TIME at the SY\_LFR\_TC\_MAX\_RATE rate. | | | | | Verify the LFR FSW is healthy. | | | |  |  |
| 12 | Send 1000 TC\_LFR\_LOAD\_KCOEFFICIENTS at the SY\_LFR\_TC\_MAX\_RATE rate. | | | | | Verify the LFR FSW is healthy. | | | |  |  |
| 13 | Send 1000 TC\_LFR\_DUMP\_KCOEFFICIENTS at the SY\_LFR\_TC\_MAX\_RATE rate. | | | | | Verify the LFR FSW is healthy. | | | |  |  |
| 14 | Send 1000 TC\_LFR\_LOAD\_FBINS\_MASK at the SY\_LFR\_TC\_MAX\_RATE rate. | | | | | Verify the LFR FSW is healthy. | | | |  |  |
| 15 | Send 1000 TC\_LFR\_LOAD\_FILTER\_PAR at the SY\_LFR\_TC\_MAX\_RATE rate. | | | | | Verify the LFR FSW is healthy. | | | |  |  |
| 16 | End | | | | |  | | | |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE** | | | | | | | | | | | |
| **Test procedure identifier** | | | | | | **Date** | | | **Assessment** | | |
| SVS-0026\_Ed2 | | | | | | DD/MM/YYYY | | | NT / NA / NOK / POK / OK | | |
| **TS Tested requirements** | | | | | | **RB Tested requirements** | | | | | |
| N/A | | | | | | REQ-LFR-SRS-5241\_Ed1 | | | | | |
| **Component:** | | | |  | | **Version:** | V2 | | | | |
| **Involved subsystems** | | | | | | | | | | | |
| SW: | | 3.2.0.24 | | | | | | | | | |
| HW: | | 1.1.91 StarDundee | | | | | | | | | |
| Start time: | | | (DD/MM/YYYY) hh:mm | | | Test Operator: | |  | | | |
| End Time: | | | (DD/MM/YYYY) hh:mm | | | Test Duration: | | 12h30 | | | |
| **Purpose:** | | | All the counters (error counters, packet counters,etc) managed by LFR FSW shall restart at 0 when they reached their maximum value | | | | | | | | |
| **General remarks about the test** | | | | | | | | | | | |
| Requirement Title | | | | HK counter management | | | | | | | |
| dependencies | | | | SSS-CP-FS-201 | | | | | | | |
| restrictions | | | |  | | | | | | | |
| means | | | |  | | | | | | | |
| input data | | | |  | | | | | | | |
| prerequisite | | | |  | | | | | | | |
| used test programs | | | |  | | | | | | | |
| test script | | | | tm\_sequence\_counter\_loop.py and tm\_sequence\_counter\_xx.py (hk,dump,science) SVS-0019 | | | | | | | |
| output data | | | | update\_time\_and\_info\_cnt\_wrap.py (SVS-0078)  check\_tc\_exe\_and\_reject\_cnt\_wrap.py  update\_timecode\_missing\_cnt\_wrap.py  update\_time\_not\_synchro\_cnt\_wrap.py  update\_timecode\_it\_and\_ctr\_cnt\_wrap.py  update\_timecode\_erroneous\_cnt\_wrap.py | | | | | | | |
| notes | | | |  | | | | | | | |
| **Procedure steps** | | | | | | | | | | | |
| **Step** | **Actions** | | | | **Expected Results** | | | | | **Comments** | **Status** |
| 1 | .Check SEQUENCE\_CNT  for each couple (APID,Destination ID) | | | | Max of SEQUENCE\_CNT is 16383  either 2^14 -1 | | | | | See SVS-0019  (step 9 to 12) |  |
| 2 | Check HK\_LFR\_UPDATE\_INFO\_TC\_CNT | | | | Max of HK\_LFR\_UPDATE\_INFO\_TC\_CNT  is 65535 either 2^16 -1 | | | | | See SVS-0078 (step3) |  |
| 3 | Check HK\_LFR\_UPDATE\_TIME\_TC\_CNT | | | | Max of HK\_LFR\_UPDATE\_TIME\_TC\_CNT  is 65535 either 2^16 -1 | | | | | See SVS-0078 (step3) |  |
| 4 | Check HK\_LFR\_EXE\_TC\_CNT  and  HK\_LFR\_REJ\_TC\_CNT | | | | Max of HK\_LFR\_EXE \_TC\_CNT and  HK\_LFR\_REJ\_TC\_CNT  is 65535 either 2^16 -1 | | | | | check\_tc\_exe\_and\_reject\_cnt\_wrap.py  around 7h30 |  |
| 5 | Check  HK\_LFR\_LE\_CNT | | | | Max of HK\_LFR\_LE\_CNT and  HK\_LFR\_ME\_CNT  is 65535 either 2^16 -1 | | | | | SVS-0078 (step3)  update\_time\_and\_info\_cnt\_wrap.py |  |
| 6 | Check  HK\_LFR\_ME\_CNT | | | | Max of HK\_LFR\_LE\_CNT and  HK\_LFR\_ME\_CNT  is 65535 either 2^16 -1 | | | | | Stress Spacewire errors are too difficult to produce |  |
| 7 | Check  HK\_LFR\_DPU\_SPW\_PKT\_RCV\_CNT  and  HK\_LFR\_DPU\_SPW\_PKT\_SENT\_CNT | | | | Max of HK\_LFR\_DPU\_SPW\_PKT\_RCV\_ CNT  and  HK\_LFR\_DPU\_SPW\_PKT\_SENT\_CNT  is 65535 either 2^16 -1 | | | | | HK\_LFR\_DPU\_SPW\_PKT\_RCV\_CNT  SVS-0078 (step3)  update\_time\_and\_info\_cnt\_wrap.py  HK\_LFR\_DPU\_SPW\_PKT\_SENT\_CNT  test de charge SVS-0900 |  |
| 8 | Check HK\_LFR\_DPU\_TICK\_OUT\_CNT | | | | Max of HK\_LFR\_DPU\_TICK\_OUT\_CNT  is 255 either 2^8 -1 | | | | | update\_timecode\_missing\_cnt\_wrap.py |  |
| 9 | Check 8 spacewire errors  HK\_LFR\_DPU\_SPW\_PARITY  HK\_LFR\_DPU\_SPW\_DISCONNECT  HK\_LFR\_DPU\_SPW\_ESCAPE  HK\_LFR\_DPU\_SPW\_CREDIT  HK\_LFR\_DPU\_SPW\_WRITE\_SYNC  HK\_LFR\_DPU\_SPW\_EARLY\_EOP  HK\_LFR\_DPU\_SPW\_INVALID\_ADDR  HK\_LFR\_DPU\_SPW\_EEP  Check 1 spacewire error  HK\_LFR\_DPU\_SPW\_RX\_TOO\_BIG | | | | Max of each counter is 255 either 2^8 -1 | | | | | spw\_failure.py (SVS-0064)  Too difficult to produce a lot of these  8 spacewire specific errors.  check\_too\_big\_cnt\_wrap.py |  |
| 10 | Check 4 timing errors  HK\_LFR\_TIMECODE\_MISSING  HK\_LFR\_TIMECODE\_IT  HK\_LFR\_TIME\_NOT\_SYNCHRO  HK\_LFR\_TIMECODE\_CTR  HK\_LFR\_TIMECODE\_ERRONEOUS  HK\_LFR\_TIMECODE\_INVALID | | | | Max of each counter is 255 either 2^8 -1 | | | | | update\_timecode\_missing\_cnt\_wrap.py  and  update\_time\_not\_synchro\_cnt\_wrap.py  update\_timecode\_it\_and\_ctr\_cnt\_wrap.py  update\_timecode\_erroneous\_cnt\_wrap.py  Too difficult to produce a lot of.this error  (SVS-0064) someone. |  |
| 11 | End | | | |  | | | | |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE** | | | | | | | | | | | | |
| **Test procedure identifier** | | | | | | | **Date** | | | **Assessment** | | |
| SVS-0027\_Ed2 | | | | | | | DD/MM/YYYY | | | NT / NA / NOK / POK / OK | | |
| **TS Tested requirements** | | | | | | | **RB Tested requirements**  REQ-LFR-SRS-5413\_Ed1 REQ-LFR-SRS-5411\_Ed2  REQ-LFR-SRS-5412\_Ed2 | | | | | |
| N/A | | | | | | |  | | | | | |
| **Component:** | | | | |  | | **Version:** | V1 | | | | |
| **Involved subsystems** | | | | | | | | | | | | |
| SW: | 3.2.0.24 | | | | | | | | | | | |
| HW: | 1.1.91 StarDundee | | | | | | | | | | | |
| Start time: | | | | (DD/MM/YYYY) hh:mm | | | Test Operator: | |  | | | |
| End Time: | | | | (DD/MM/YYYY) hh:mm | | | Test Duration: | | 30mn + 10mn+10mn | | | |
| **Purpose:** | | | Check According to [AD09], the SpW protocol header of the packets exchanged is made up of 4 bytes:  Target Logical Address.  Protocol ID = 0x02 meaning CCSDS.  Reserved byte = 0x00.  User Application byte = 0x00.  Concerning Target Logical Address:  Check all the SpW packets produced toward LFR have the Destination Logical Address field set to SY\_LFR\_DPU\_LA.  Check all the SpW packets produced by LFR have the Destination Logical Address field set to SY\_DPU\_LFR\_LA. | | | | | | | | | |
| **General remarks about the test** | | | | | | | | | | | | |
| Requirement Title | | | | DPU / RPW Analyzer SW communication protocol | | | | | | | | |
| dependencies | | | | SpaceWire – CCSDS packet transfer protocol ECSS-E-ST-50-53C, SSS-IF-DPS-EQ-175 and  SSS-IF-DPS-EQ-071 and SSS-IF-DPS-EQ-071 | | | | | | | | |
| restrictions | | | | Verification is made only once. It is supposed to be sufficient. | | | | | | | | |
| means | | | |  | | | | | | | | |
| input data | | | |  | | | | | | | | |
| prerequisite | | | |  | | | | | | | | |
| used test programs | | | |  | | | | | | | | |
| test script | | | | ccsds\_header\_fields\_asm\_4s.py (step1 to 4, 9 to 11)  ccsds\_header\_fields\_step1.py for SBM1 (5,6)  ccsds\_header\_fields\_step2.py for SBM2 (7,8) | | | | | | | | |
| output data | | | |  | | | | | | | | |
| notes | | | | This rule is checked by the Python treatment, for each TC (resp.TM) received (resp. sent) by the LFR.  ASM=4s and other parameters are default values  All LFR TC are implemented.  We can check the generated file TM.csv with excel.  We can check the generated file TM.csv with excel.  Stress tests use all default parameters (ASM=3600s)  . | | | | | | | | |
| **Procedure steps** | | | | | | | | | | | | |
| **Step** | | **Actions** | | | | **Expected Results** | | | | | **Comments** | **Status** |
|  | | In STANDBY, send each TC. Ensure Destination Logical Address field is set to SY\_LFR\_ DPU LA, Protocol ID = 0x02 meaning CCSDS. Reserved byte = 0x00. User Application byte = 0x00. | | | | For each TM\_LFR\_HK, TM\_LFR\_PARAMETER\_DUMP TM\_LFR\_KCOEFFICIENTS\_DUMP received,  verify:  Destination Logical Address field set to SY\_DPU\_LFR\_LA (0xfe or 0x01)  Protocol ID = 0x02 meaning CCSDS.  Reserved byte = 0x00.  User Application byte = 0x00.  Ensure all the defined TCs are sent toward LFR  with the destination logical address is SY\_LFR\_DPU\_LA (0xfe)  For each TM, verify the destination logical address is SY\_DPU\_LFR\_LA (0x01) | | | | |  |  |
|  | | In NORMAL, send each TC. Ensure Destination Logical Address field is set to SY\_LFR\_ DPU LA, Protocol ID = 0x02 meaning CCSDS. Reserved byte = 0x00. User Application byte = 0x00.  Wait 600s | | | | For each TM\_LFR\_HK, TM\_LFR\_PARAMETER\_DUMP, TM\_LFR\_KCOEFFICIENTS\_DUMP,  TM\_LFR\_SCIENCE\_NORMAL\_ASM\_F0, TM\_LFR\_SCIENCE\_NORMAL\_ASM\_F1, TM\_LFR\_SCIENCE\_NORMAL\_ASM\_F2, TM\_LFR\_SCIENCE\_NORMAL\_BP1\_F0, TM\_LFR\_SCIENCE\_NORMAL\_BP1\_F1, TM\_LFR\_SCIENCE\_NORMAL\_BP1\_F2, TM\_LFR\_SCIENCE\_NORMAL\_BP2\_F0, TM\_LFR\_SCIENCE\_NORMAL\_BP2\_F1, TM\_LFR\_SCIENCE\_NORMAL\_BP2\_F2, TM\_LFR\_SCIENCE\_NORMAL\_CWF\_F3, TM\_LFR\_SCIENCE\_NORMAL\_SWF\_F0, TM\_LFR\_SCIENCE\_NORMAL\_SWF\_F1, TM\_LFR\_SCIENCE\_NORMAL\_SWF\_F2 received,  .  Destination Logical Address field set to SY\_DPU\_LFR\_LA (0xfe or 0x01)  Protocol ID = 0x02 meaning CCSDS.  Reserved byte = 0x00.  User Application byte = 0x00.  Ensure all the defined TCs are sent toward LFR  with the destination logical address is SY\_LFR\_DPU\_LA (0xfe)  For each TM, verify the destination logical address is SY\_DPU\_LFR\_LA (0x01) | | | | |  |  |
|  | | In NORMAL, with SY\_LFR\_N\_CWF\_LONG\_F3 = 1  send each TC. Ensure Destination Logical Address field is set to SY\_LFR\_ DPU LA, Protocol ID = 0x02 meaning CCSDS. Reserved byte = 0x00. User Application byte = 0x00.  Wait 600s | | | | For each TM\_LFR\_HK, TM\_LFR\_PARAMETER\_DUMP,  TM\_LFR\_KCOEFFICIENTS\_DUMP,  TM\_LFR\_SCIENCE\_NORMAL\_ASM\_F0, TM\_LFR\_SCIENCE\_NORMAL\_ASM\_F1, TM\_LFR\_SCIENCE\_NORMAL\_ASM\_F2, TM\_LFR\_SCIENCE\_NORMAL\_BP1\_F0, TM\_LFR\_SCIENCE\_NORMAL\_BP1\_F1, TM\_LFR\_SCIENCE\_NORMAL\_BP1\_F2, TM\_LFR\_SCIENCE\_NORMAL\_BP2\_F0, TM\_LFR\_SCIENCE\_NORMAL\_BP2\_F1, TM\_LFR\_SCIENCE\_NORMAL\_BP2\_F2, TM\_LFR\_SCIENCE\_NORMAL\_CWF\_LONG\_F3, TM\_LFR\_SCIENCE\_NORMAL\_SWF\_F0, TM\_LFR\_SCIENCE\_NORMAL\_SWF\_F1, TM\_LFR\_SCIENCE\_NORMAL\_SWF\_F2 received  Verify:  Destination Logical Address field set to SY\_DPU\_LFR\_LA (0xfe or 0x01)  Protocol ID = 0x02 meaning CCSDS.  Reserved byte = 0x00.  User Application byte = 0x00.  Ensure all the defined TCs are sent toward LFR  with the destination logical address is SY\_LFR\_DPU\_LA (0xfe)  For each TM, verify the destination logical address is SY\_DPU\_LFR\_LA (0x01) | | | | |  |  |
|  | | In BURST, send each TC. Ensure Destination Logical Address field is set to SY\_LFR\_ DPU LA, Protocol ID = 0x02 meaning CCSDS. Reserved byte = 0x00. User Application byte = 0x00.  Wait 50s | | | | For each TM\_LFR\_HK, TM\_LFR\_PARAMETER\_DUMP,  TM\_LFR\_KCOEFFICIENTS\_DUMP,  TM\_LFR\_SCIENCE\_BURST\_BP1\_F0, TM\_LFR\_SCIENCE\_BURST\_BP1\_F1, TM\_LFR\_SCIENCE\_BURST\_BP2\_F0, TM\_LFR\_SCIENCE\_BURST\_BP2\_F1, TM\_LFR\_SCIENCE\_BURST\_CWF\_F2 received,  Verify:  Destination Logical Address field set to SY\_DPU\_LFR\_LA (0xfe or 0x01)  Protocol ID = 0x02 meaning CCSDS.  Reserved byte = 0x00.  User Application byte = 0x00.  Ensure all the defined TCs are sent toward LFR  with the destination logical address is SY\_LFR\_DPU\_LA (0xfe)  For each TM, verify the destination logical address is SY\_DPU\_LFR\_LA (0x01) | | | | |  |  |
|  | | In SBM1, send each TC. Ensure Destination Logical Address field is set to SY\_LFR\_ DPU LA, Protocol ID = 0x02 meaning CCSDS. Reserved byte = 0x00. User Application byte = 0x00.  ASM = 4s  Wait 600s | | | | For each TM\_LFR\_HK, TM\_LFR\_PARAMETER\_DUMP,  TM\_LFR\_KCOEFFICIENTS\_DUMP,  TM\_LFR\_SCIENCE\_NORMAL\_ASM\_F0, TM\_LFR\_SCIENCE\_NORMAL\_ASM\_F1, TM\_LFR\_SCIENCE\_NORMAL\_ASM\_F2,  TM\_LFR\_SCIENCE\_NORMAL\_BP1\_F0, TM\_LFR\_SCIENCE\_NORMAL\_BP1\_F1, TM\_LFR\_SCIENCE\_NORMAL\_BP1\_F2, TM\_LFR\_SCIENCE\_NORMAL\_BP2\_F0, TM\_LFR\_SCIENCE\_NORMAL\_BP2\_F1, TM\_LFR\_SCIENCE\_NORMAL\_BP2\_F2, TM\_LFR\_SCIENCE\_NORMAL\_CWF\_F3, TM\_LFR\_SCIENCE\_NORMAL\_SWF\_F0, TM\_LFR\_SCIENCE\_NORMAL\_SWF\_F1, TM\_LFR\_SCIENCE\_NORMAL\_SWF\_F2, TM\_LFR\_SCIENCE\_SBM1\_BP1\_F0, TM\_LFR\_SCIENCE\_SBM1\_BP2\_F0, TM\_LFR\_SCIENCE\_SBM1\_CWF\_F1 received,  Verify:  Destination Logical Address field set to SY\_DPU\_LFR\_LA (0xfe or 0x01)  Protocol ID = 0x02 meaning CCSDS.  Reserved byte = 0x00.  User Application byte = 0x00.  Ensure all the defined TCs are sent toward LFR  with the destination logical address is SY\_LFR\_DPU\_LA (0xfe)  For each TM, verify the destination logical address is SY\_DPU\_LFR\_LA (0x01) | | | | |  |  |
|  | | In SBM1, send each TC. Ensure Destination Logical Address field is set to SY\_LFR\_ DPU LA, Protocol ID = 0x02 meaning CCSDS. Reserved byte = 0x00. User Application byte = 0x00.  ASM = 4s  SY\_LFR\_N\_CWF\_LONG\_F3=1  Wait 600s | | | | For each TM\_LFR\_HK, TM\_LFR\_PARAMETER\_DUMP,  TM\_LFR\_KCOEFFICIENTS\_DUMP,  TM\_LFR\_SCIENCE\_NORMAL\_ASM\_F0, TM\_LFR\_SCIENCE\_NORMAL\_ASM\_F1, TM\_LFR\_SCIENCE\_NORMAL\_ASM\_F2,  TM\_LFR\_SCIENCE\_NORMAL\_BP1\_F0, TM\_LFR\_SCIENCE\_NORMAL\_BP1\_F1, TM\_LFR\_SCIENCE\_NORMAL\_BP1\_F2, TM\_LFR\_SCIENCE\_NORMAL\_BP2\_F0, TM\_LFR\_SCIENCE\_NORMAL\_BP2\_F1, TM\_LFR\_SCIENCE\_NORMAL\_BP2\_F2, TM\_LFR\_SCIENCE\_NORMAL\_CWF\_LONG\_F3, TM\_LFR\_SCIENCE\_NORMAL\_SWF\_F0, TM\_LFR\_SCIENCE\_NORMAL\_SWF\_F1, TM\_LFR\_SCIENCE\_NORMAL\_SWF\_F2, TM\_LFR\_SCIENCE\_SBM1\_BP1\_F0, TM\_LFR\_SCIENCE\_SBM1\_BP2\_F0, TM\_LFR\_SCIENCE\_SBM1\_CWF\_F1 received,  Verify:  Destination Logical Address field set to SY\_DPU\_LFR\_LA (0xfe or 0x01)  Protocol ID = 0x02 meaning CCSDS.  Reserved byte = 0x00.  User Application byte = 0x00.  Ensure all the defined TCs are sent toward LFR  with the destination logical address is SY\_LFR\_DPU\_LA (0xfe)  For each TM, verify the destination logical address is SY\_DPU\_LFR\_LA (0x01) | | | | |  |  |
|  | | In SBM2, send each TC. Ensure Destination Logical Address field is set to SY\_LFR\_ DPU LA, Protocol ID = 0x02 meaning CCSDS. Reserved byte = 0x00. User Application byte = 0x00.  ASM=4s  Wait 600s | | | | For each TM\_LFR\_HK, TM\_LFR\_PARAMETER\_DUMP,  TM\_LFR\_KCOEFFICIENTS\_DUMP,  TM\_LFR\_SCIENCE\_NORMAL\_ASM\_F0, TM\_LFR\_SCIENCE\_NORMAL\_ASM\_F1, TM\_LFR\_SCIENCE\_NORMAL\_ASM\_F2,  TM\_LFR\_SCIENCE\_NORMAL\_BP1\_F0, TM\_LFR\_SCIENCE\_NORMAL\_BP1\_F1, TM\_LFR\_SCIENCE\_NORMAL\_BP1\_F2, TM\_LFR\_SCIENCE\_NORMAL\_BP2\_F0, TM\_LFR\_SCIENCE\_NORMAL\_BP2\_F1, TM\_LFR\_SCIENCE\_NORMAL\_BP2\_F2, TM\_LFR\_SCIENCE\_NORMAL\_CWF\_F3, TM\_LFR\_SCIENCE\_NORMAL\_SWF\_F0, TM\_LFR\_SCIENCE\_NORMAL\_SWF\_F1, TM\_LFR\_SCIENCE\_NORMAL\_SWF\_F2, TM\_LFR\_SCIENCE\_SBM2\_BP1\_F0, TM\_LFR\_SCIENCE\_SBM2\_BP1\_F1, TM\_LFR\_SCIENCE\_SBM2\_BP2\_F0, TM\_LFR\_SCIENCE\_SBM2\_BP2\_F1, TM\_LFR\_SCIENCE\_SBM2\_CWF\_F2 received,  Verify:  Destination Logical Address field set to SY\_DPU\_LFR\_LA (0xfe or 0x01)  Protocol ID = 0x02 meaning CCSDS.  Reserved byte = 0x00.  User Application byte = 0x00.  Ensure all the defined TCs are sent toward LFR  with the destination logical address is SY\_LFR\_DPU\_LA (0xfe)  For each TM, verify the destination logical address is SY\_DPU\_LFR\_LA (0x01) | | | | |  |  |
|  | | In SBM2, send each TC. Ensure Destination Logical Address field is set to SY\_LFR\_ DPU LA, Protocol ID = 0x02 meaning CCSDS. Reserved byte = 0x00. User Application byte = 0x00.  ASM = 4s  SY\_LFR\_N\_CWF\_LONG\_F3=1  Wait 600s | | | | For each TM\_LFR\_HK, TM\_LFR\_PARAMETER\_DUMP,  TM\_LFR\_KCOEFFICIENTS\_DUMP,  TM\_LFR\_SCIENCE\_NORMAL\_ASM\_F0, TM\_LFR\_SCIENCE\_NORMAL\_ASM\_F1, TM\_LFR\_SCIENCE\_NORMAL\_ASM\_F2,  TM\_LFR\_SCIENCE\_NORMAL\_BP1\_F0, TM\_LFR\_SCIENCE\_NORMAL\_BP1\_F1, TM\_LFR\_SCIENCE\_NORMAL\_BP1\_F2, TM\_LFR\_SCIENCE\_NORMAL\_BP2\_F0, TM\_LFR\_SCIENCE\_NORMAL\_BP2\_F1, TM\_LFR\_SCIENCE\_NORMAL\_BP2\_F2, TM\_LFR\_SCIENCE\_NORMAL\_CWF\_LONG\_F3, TM\_LFR\_SCIENCE\_NORMAL\_SWF\_F0, TM\_LFR\_SCIENCE\_NORMAL\_SWF\_F1, TM\_LFR\_SCIENCE\_NORMAL\_SWF\_F2, TM\_LFR\_SCIENCE\_SBM2\_BP1\_F0, TM\_LFR\_SCIENCE\_SBM2\_BP1\_F1, TM\_LFR\_SCIENCE\_SBM2\_BP2\_F0, TM\_LFR\_SCIENCE\_SBM2\_BP2\_F1, TM\_LFR\_SCIENCE\_SBM2\_CWF\_F2 received,  Verify:  Destination Logical Address field set to SY\_DPU\_LFR\_LA (0xfe or 0x01)  Protocol ID = 0x02 meaning CCSDS.  Reserved byte = 0x00.  User Application byte = 0x00.  Ensure all the defined TCs are sent toward LFR  with the destination logical address is SY\_LFR\_DPU\_LA (0xfe)  For each TM, verify the destination logical address is SY\_DPU\_LFR\_LA (0x01) | | | | |  |  |
|  | | Ensure  TM\_LFR\_TC\_EXE\_NOT\_EXECUTABLE, TM\_LFR\_TC\_EXE\_SUCCESS have been sent in previous steps. | | | | For M\_LFR\_TC\_EXE\_NOT\_EXECUTABLE, TM\_LFR\_TC\_EXE\_NOT\_IMPLEMENTED, TM\_LFR\_TC\_EXE\_SUCCESS received,  Verify:  Destination Logical Address field set to SY\_DPU\_LFR\_LA (0xfe or 0x01)  Protocol ID = 0x02 meaning CCSDS.  Reserved byte = 0x00.  User Application byte = 0x00.  Ensure all the defined TCs are sent toward LFR  with the destination logical address is SY\_LFR\_DPU\_LA (0xfe)  For each TM, verify the destination logical address is SY\_DPU\_LFR\_LA (0x01) | | | | |  |  |
|  | | Provoke a TM\_LFR\_TC\_EXE\_INCONSISTENT: send a TC\_LFR\_ENTER\_MODE with CP\_LFR\_MODE field = 0xF. | | | | For TM\_LFR\_TC\_EXE\_INCONSISTENT received  Verify:  Destination Logical Address field set to SY\_DPU\_LFR\_LA (0xfe or 0x01)  Protocol ID = 0x02 meaning CCSDS.  Reserved byte = 0x00.  User Application byte = 0x00.  Ensure all the defined TCs are sent toward LFR  with the destination logical address is SY\_LFR\_DPU\_LA (0xfe)  For each TM, verify the destination logical address is SY\_DPU\_LFR\_LA (0x01) | | | | |  |  |
|  | | Provoke a TM\_LFR\_TC\_EXE\_CORRUPTED: send a TC\_LFR\_ENTER\_MODE with corrupted CRC. | | | | For TM\_LFR\_TC\_EXE\_CORRUPTED received,  Verify:  Destination Logical Address field set to SY\_DPU\_LFR\_LA (0xfe or 0x01)  Protocol ID = 0x02 meaning CCSDS.  Reserved byte = 0x00.  User Application byte = 0x00.  Ensure all the defined TCs are sent toward LFR  with the destination logical address is SY\_LFR\_DPU\_LA (0xfe)  For each TM, verify the destination logical address is SY\_DPU\_LFR\_LA (0x01) | | | | |  |  |
|  | | End | | | |  | | | | |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE** | | | | | | | | | | | |
| **Test procedure identifier** | | | | | **Date** | | | | **Assessment** | | |
| SVS-0028\_Ed1 | | | | | DD/MM/YYYY | | | | NT / NA / NOK / POK / OK | | |
| **TS Tested requirements** | | | | | **RB Tested requirements** | | | | | | |
| N/A | | | | | REQ-LFR-SRS-5502\_Ed1  REQ-LFR-SRS-5500\_Ed1 | | | | | | |
| **Component:** | | | |  | **Version:** | | V2 | | | | |
| **Involved subsystems** | | | | | | | | | | | |
| SW: | | 3.2.0.24 | | | | | | | | | |
| HW: | | 1.1.91 StarDundee | | | | | | | | | |
| Start time: | | | (DD/MM/YYYY) hh:mm | | Test Operator: | | |  | | | |
| End Time: | | | (DD/MM/YYYY) hh:mm | | Test Duration: | | | 25s | | | |
| **Purpose:** | | | Check, in the STANDBY mode, the LFR FSW accepts commands to configure the HW and the SW.  The LFR shall handle the following modes  **STANDBY** NORMAL BURST SBM1 SBM2 | | | | | | | | |
| **General remarks about the test** | | | | | | | | | | | |
| Requirement Title | | | Equipment mode management | | | | | | | | |
| dependencies | | | SSS-CP-EQS-250 – SSS-CP-EQS-230 | | | | | | | | |
| restrictions | | |  | | | | | | | | |
| means | | |  | | | | | | | | |
| input data | | |  | | | | | | | | |
| prerequisite | | | LFR is in STANDBY mode when RPW is in SERVICE mode; see § “LFR mode management” (§ 5.5.1). | | | | | | | | |
| used test programs | | |  | | | | | | | | |
| test script | | | configure\_standby.py | | | | | | | | |
| output data | | |  | | | | | | | | |
| notes | | |  | | | | | | | | |
| **Procedure steps** | | | | | | | | | | | |
| **Step** | **Actions** | | | | | **Expected Results** | | | | **Comments** | **Status** |
|  | Ensure LFR FSW is in STANDBY mode. Send a TC\_LFR\_LOAD\_COMMON\_PAR command. | | | | | Verify the LFR FSW accepts TC\_LFR\_LOAD\_COMMON\_PAR command.  TM\_LFR\_EXE\_SUCCESS | | | |  |  |
|  | Ensure LFR FSW is in STANDBY mode. Send a TC\_LFR\_LOAD\_NORMAL\_PAR command. | | | | | Verify the LFR FSW accepts TC\_LFR\_LOAD\_NORMAL\_PAR command.  TM\_LFR\_EXE\_SUCCESS | | | |  |  |
|  | Ensure LFR FSW is in STANDBY mode. Send a TC\_LFR\_LOAD\_BURST\_PAR command. | | | | | Verify the LFR FSW accepts TC\_LFR\_LOAD\_BURST\_PAR command.  C | | | |  |  |
|  | Ensure LFR FSW is in STANDBY mode. Send a TC\_LFR\_LOAD\_SBM1\_PAR command. | | | | | Verify the LFR FSW accepts TC\_LFR\_LOAD\_SBM1\_PAR command.  TM\_LFR\_EXE\_SUCCESS | | | |  |  |
|  | Ensure LFR FSW is in STANDBY mode. Send a TC\_LFR\_LOAD\_SBM2\_PAR command. | | | | | Verify the LFR FSW accepts TC\_LFR\_LOAD\_SBM2\_PAR command.  TM\_LFR\_EXE\_SUCCESS | | | |  |  |
|  | Ensure LFR FSW is in STANDBY mode. Send a TC\_LFR\_UPDATE\_TIME command. | | | | | Verify the LFR FSW accepts TC\_LFR\_UPDATE\_TIME command.  No acknowledge | | | |  |  |
|  | Ensure LFR FSW is in STANDBY mode. Send a TC\_LFR\_ENABLE\_CALIBRATION command. | | | | | Verify the LFR FSW accepts the command.  TM\_LFR\_EXE\_SUCCESS | | | |  |  |
|  | Ensure LFR FSW is in STANDBY mode. Send a TC\_LFR\_DISABLE\_CALIBRATION command. | | | | | Verify the LFR FSW accepts the command.  TM\_LFR\_EXE\_SUCCESS | | | |  |  |
|  | Ensure LFR FSW is in STANDBY mode. Send a TC\_LFR\_UPDATE\_INFO command. | | | | | Verify the LFR FSW accepts the command.  No ackknowledge | | | |  |  |
|  | Ensure LFR FSW is in STANDBY mode. Send a TC\_LFR\_LOAD\_KCOEFFICIENTS command. | | | | | Verify the LFR FSW accepts the command.  TM\_LFR\_EXE\_SUCCESS | | | |  |  |
|  | Ensure LFR FSW is in STANDBY mode. Send a TC\_LFR\_DUMP\_KCOEFFICIENTS command. | | | | | Verify the LFR FSW accepts the command.  TM\_LFR\_EXE\_SUCCESS | | | |  |  |
|  | Ensure LFR FSW is in STANDBY mode. Send a TC\_LFR\_LOAD\_FBINS\_MASK command. | | | | | Verify the LFR FSW accepts the command.  TM\_LFR\_EXE\_SUCCESS | | | |  |  |
|  | Ensure LFR FSW is in STANDBY mode. Send a TC\_LFR\_LOAD\_FILTER\_PAR command. | | | | | Verify the LFR FSW accepts the command.  TM\_LFR\_EXE\_SUCCESS | | | |  |  |
|  | End | | | | |  | | | |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE** | | | | | | | | | | | | | |
| **Test procedure identifier** | | | | | | **Date** | | | **Assessment** | | | | |
| SVS-0029\_Ed2 | | | | | | DD/MM/YYYY | | | NT / NA / NOK / POK / OK | | | | |
| **TS Tested requirements** | | | | | | **RB Tested requirements** | | | | | | | |
| N/A | | | | | | REQ-LFR-SRS-5503\_Ed1  REQ-LFR-SRS-5548\_Ed1  REQ-LFR-SRS-5500\_Ed1  REQ-LFR-SRS-5514\_Ed1 | | | | | | | |
| **Component:** | | | |  | | **Version:** | V1 | | | | | | |
| **Involved subsystems** | | | | | | | | | | | | | |
| SW: | | 3.2.0.24 | | | | | | | | | | | |
| HW: | | 1.1.91 Stardundee | | | | | | | | | | | |
| Start time: | | | (DD/MM/YYYY) hh:mm | | | Test Operator: | |  | | | | | |
| End Time: | | | (DD/MM/YYYY) hh:mm | | | Test Duration: | | 1h10 + 15h | | | | | |
| **Purpose:** | | | Check, in the NORMAL mode, the LFR FSW sends one single data stream whose content corresponds to the NORMAL mode parameter set.  The LFR FSW shall handle commands for configuring the different science sub-mode parameter sets.   * NORMAL mode parameters: TC\_LFR\_LOAD\_NORMAL\_PAR   The LFR shall handle the following modes  STANDBY **NORMA**L BURST SBM1 SBM2 | | | | | | | | | | |
| **General remarks about the test** | | | | | | | | | | | | | |
| Requirement Title | | | Equipment mode management and Equipment configuration management | | | | | | | | | | |
| dependencies | | | SSS-CP-EQS-260, REQ-RPW-SYS-0303 SSS-CP-EQS-190 SSS-CP-EQS-230 SSS-CP-EQS-370 | | | | | | | | | | |
| restrictions | | |  | | | | | | | | | | |
| means | | |  | | | | | | | | | | |
| input data | | |  | | | | | | | | | | |
| prerequisite | | | LFR is in NORMAL mode when RPW is in SURVEY\_NORMAL (\_C) mode; see § “LFR mode management” (§ 5.5.1). | | | | | | | | | | |
| used test programs | | |  | | | | | | | | | | |
| test script | | | normal\_mode\_parameter\_set.py (step1 to step9) and  normal\_mode\_parameter\_set\_nominal.py (ASM=3600s) +  two long test with timegen with timegen | | | | | | | | | | |
| output data | | |  | | | | | | | | | | |
| notes | | | This TESTCASE verifies:  -the case of nominal values  -the cases with TC\_LFR\_LOAD\_NORMAL\_PAR sent any mode (except NORMAL).  -Steps must be ensure rule is applied for CWF\_F3 an CWF\_LONG\_F3  SY\_LFR\_N\_SWF\_L = 2048 (not to be changed) | | | | | | | | | | |
| **Procedure steps** | | | | | | | | | | | | | |
| **Step** | **Actions** | | | | | | | | Expected Results | **Comments** | | **Status** |
|  | **Use normal\_mode\_parameter\_set.py**  Launch the FSW LFR.  Only ASM=4s for this test  Send TC\_LFR\_LOAD\_NORMAL\_PAR, in STANDBY mode  with SY\_LFR\_N\_ASM\_P=4 ,  send a TC\_LFR\_DUMP\_PAR to valid the parameter  send TC\_LFR\_ENTER\_MODE with CP\_LFR\_MODE= NORMAL.  With HK, ensure the LFR FSW is in the NORMAL mode.  Wait at least 600s (1.2\*max(SY\_LFR\_N\_SWF\_P, SY\_LFR\_N\_ASM\_P, SY\_LFR\_N\_BP\_P0, SY\_LFR\_N\_BP\_P1). | | | | | | | | Verify the LFR FSW sends one single data stream whose content corresponds to the NORMAL mode parameter set (default values).  TM\_LFR\_PARAMETER\_DUMP shows :  SY\_LFR\_N\_SWF\_L=2048, SY\_LFR\_N\_SWF\_P=300(s), SY\_LFR\_N\_ASM\_P=4(s), SY\_LFR\_N\_BP\_P0=4(s), SY\_LFR\_N\_BP\_P1=20(s), SPARE=0x0, SY\_LFR\_N\_CWF\_LONG\_F3=0 |  | |  |
|  | In STANDBY mode, send TC\_LFR\_LOAD\_NORMAL\_PAR with:  SY\_LFR\_N\_SWF\_L=2048  SY\_LFR\_N\_SWF\_P= (default value) /2  SY\_LFR\_N\_ASM\_P= (default value) /2  SY\_LFR\_N\_BP\_P0= (default value) /2  SY\_LFR\_N\_BP\_P1= (default value) /2  SY\_LFR\_N\_CWF\_LONG\_F3 = default value  send a TC\_LFR\_DUMP\_PAR to valid the parameter  Send TC\_LFR\_ENTER\_MODE with CP\_LFR\_MODE= NORMAL.  With HK, ensure the LFR FSW is in the NORMAL mode. Wait at least 600s 2\*max(SY\_LFR\_N\_SWF\_P, SY\_LFR\_N\_ASM\_P, SY\_LFR\_N\_BP\_P0, SY\_LFR\_N\_BP\_P1). | | | | | | | | Verify the LFR FSW  Rejects this TC  With TM\_LFR\_TC\_EXE\_INCONSISTENT with PA\_RPW\_BYTE=16  = SY\_LFR\_N\_BP\_P0  PA\_RPW\_RCV\_VALUE=2  TM\_LFR\_PARAMETER\_DUMP shows :  SY\_LFR\_N\_SWF\_L=2048, SY\_LFR\_N\_SWF\_P=300(s), SY\_LFR\_N\_ASM\_P=4(s), SY\_LFR\_N\_BP\_P0=4(s), SY\_LFR\_N\_BP\_P1=20(s), SY\_LFR\_N\_CWF\_LONG\_F3=0 |  | |  |
|  | In STANDBY mode, send TC\_LFR\_LOAD\_NORMAL\_PAR  to test maximum rate  with:  SY\_LFR\_N\_SWF\_L=2048  SY\_LFR\_N\_SWF\_P= 22  SY\_LFR\_N\_ASM\_P= 4  SY\_LFR\_N\_BP\_P0= default value  SY\_LFR\_N\_BP\_P1= default value  SY\_LFR\_N\_CWF\_LONG\_F3 = 1  send a TC\_LFR\_DUMP\_PAR to valid the parameter  Send TC\_LFR\_ENTER\_MODE with CP\_LFR\_MODE= NORMAL.  With HK, ensure the LFR FSW is in the NORMAL mode. Wait at least 2\*max(SY\_LFR\_N\_SWF\_P, SY\_LFR\_N\_ASM\_P, SY\_LFR\_N\_BP\_P0, SY\_LFR\_N\_BP\_P1). | | | | | | | | TM\_LFR\_PARAMETER\_DUMP shows :  SY\_LFR\_N\_SWF\_L=2048, SY\_LFR\_N\_SWF\_P=22(s), SY\_LFR\_N\_ASM\_P=4(s), SY\_LFR\_N\_BP\_P0=4(s), SY\_LFR\_N\_BP\_P1=20(s), SY\_LFR\_N\_CWF\_LONG\_F3=1  Verify the LFR FSW sends one single data stream whose content corresponds to the NORMAL mode parameter set. |  | |  |
|  | In NORMAL mode, send TC\_LFR\_LOAD\_NORMAL\_PAR with default values:  Send TC\_LFR\_ENTER\_MODE with CP\_LFR\_MODE= NORMAL.  With HK, ensure the LFR FSW is in the NORMAL mode. Wait at least 2\*max(SY\_LFR\_N\_SWF\_P, SY\_LFR\_N\_ASM\_P, SY\_LFR\_N\_BP\_P0, SY\_LFR\_N\_BP\_P1) of previous values ​​accepted. | | | | | | | | Reject with TM\_LFR\_TC\_EXE\_NOT\_EXECUTABLE.  Verify the LFR FSW sends one single data stream whose content corresponds to the NORMAL mode parameter set following  SY\_LFR\_N\_SWF\_L=2048  SY\_LFR\_N\_SWF\_P= 22  SY\_LFR\_N\_ASM\_P= 300  SY\_LFR\_N\_BP\_P0= default value  SY\_LFR\_N\_BP\_P1= default value  SY\_LFR\_N\_CWF\_LONG\_F3 = 0 |  | |  |
|  | In SBM1 mode, send TC\_LFR\_LOAD\_NORMAL\_PAR with: default value  Send TC\_LFR\_ENTER\_MODE with CP\_LFR\_MODE= NORMAL.  With HK, ensure the LFR FSW is in the NORMAL mode. Wait at least 2\*max(SY\_LFR\_N\_SWF\_P, SY\_LFR\_N\_ASM\_P, SY\_LFR\_N\_BP\_P0, SY\_LFR\_N\_BP\_P1) of previous values ​​accepted. | | | | | | | | Reject with TM\_LFR\_TC\_EXE\_NOT\_EXECUTABLE.  Verify the LFR FSW sends one single data stream whose content corresponds to the NORMAL mode parameter set following  SY\_LFR\_N\_SWF\_L=2048  SY\_LFR\_N\_SWF\_P= 22  SY\_LFR\_N\_ASM\_P= 300  SY\_LFR\_N\_BP\_P0= default value  SY\_LFR\_N\_BP\_P1= default value  SY\_LFR\_N\_CWF\_LONG\_F3 = 1 |  | |  |
|  | In SBM2 mode, send TC\_LFR\_LOAD\_NORMAL\_PAR with default value:  Send TC\_LFR\_ENTER\_MODE with CP\_LFR\_MODE= NORMAL.  With HK, ensure the LFR FSW is in the NORMAL mode. Wait at least 2\*max(SY\_LFR\_N\_SWF\_P, SY\_LFR\_N\_ASM\_P, SY\_LFR\_N\_BP\_P0, SY\_LFR\_N\_BP\_P1) of previous values ​​accepted. | | | | | | | | Reject with TM\_LFR\_TC\_EXE\_NOT\_EXECUTABLE.  Verify the LFR FSW sends one single data stream whose content corresponds to the NORMAL mode parameter set following  SY\_LFR\_N\_SWF\_L=2048  SY\_LFR\_N\_SWF\_P= 22  SY\_LFR\_N\_ASM\_P= 300  SY\_LFR\_N\_BP\_P0= default value  SY\_LFR\_N\_BP\_P1= default value  SY\_LFR\_N\_CWF\_LONG\_F3 = 1 |  | |  |
|  | In STANDBY mode, send TC\_LFR\_LOAD\_NORMAL\_PAR with:  SY\_LFR\_N\_SWF\_L=2048  SY\_LFR\_N\_SWF\_P=0xffff (s)  SY\_LFR\_N\_ASM\_P=0xffff (s)  SY\_LFR\_N\_BP\_P0=0xff (s)  SY\_LFR\_N\_BP\_P1=0xff (s).  SY\_LFR\_N\_CWF\_LONG\_F3 = 1  Then entering in NORMAL mode with maximum values | | | | | | | | Verify the LFR FSW accept the TC\_LFR\_LOAD\_NORMAL\_PAR.  We obtained  TM\_LFR\_SCIENCE\_NORMAL\_CWF\_LONG\_F3,  TM\_LFR\_SCIENCE\_NORMAL\_BP1\_Fi  TM\_LFR\_SCIENCE\_NORMAL\_BP2\_Fi  i=0,1,2  For SWF et ASM products, the time is too long to wait. | Validate the permitted ranges (currently, not the period stream). | |  |
|  | In STANDBY mode, send TC\_LFR\_LOAD\_NORMAL\_PAR with:  SY\_LFR\_N\_SWF\_L=2048  SY\_LFR\_N\_SWF\_P= 22(s)  SY\_LFR\_N\_ASM\_P=4 (s)  SY\_LFR\_N\_BP\_P0=4 (s)  SY\_LFR\_N\_BP\_P1=20 (s).  SY\_LFR\_N\_CWF\_LONG\_F3 = 0  Then entering in NORMAL mode with minimum values | | | | | | | | Verify the LFR FSW accept the TC\_LFR\_LOAD\_NORMAL\_PAR.  Verify the LFR FSW sends one single data stream whose content corresponds to the NORMAL mode parameter set. |  | |  |
|  | Extra case: In STANDBY mode, send TC\_LFR\_LOAD\_NORMAL\_PAR with default value:.  SY\_LFR\_N\_SWF\_L=default value=2048  SY\_LFR\_N\_SWF\_P=default value  SY\_LFR\_N\_ASM\_P=4 (s)  SY\_LFR\_N\_BP\_P0=default value  SY\_LFR\_N\_BP\_P1=default value.  SY\_LFR\_N\_CWF\_LONG\_F3 = default value  Then entering in NORMAL mode. | | | | | | | | Verify the LFR FSW sends one single data stream whose content corresponds to the NORMAL mode parameter set.  For ASM=3600 we don’t wait |  | |  |
|  | A long test is done during 15 hours in NORMAL MODE with defaults values.  SY\_LFR\_N\_SWF\_L=default value  SY\_LFR\_N\_SWF\_P=default value  SY\_LFR\_N\_ASM\_P=defaultvalue  SY\_LFR\_N\_BP\_P0=default value  SY\_LFR\_N\_BP\_P1=default value.  SY\_LFR\_N\_CWF\_LONG\_F3 = default value  We also use the timegen software (send CTR in nominal behavior). | | | | | | | | Verify the LFR FSW sends one single data stream whose content corresponds to the NORMAL mode parameter set. |  | |  |
|  | A long test is done during 14 hours in NORMAL MODE with defaults values.  SY\_LFR\_N\_SWF\_L=default value  SY\_LFR\_N\_SWF\_P=default value  SY\_LFR\_N\_ASM\_P=default value  SY\_LFR\_N\_BP\_P0=default value  SY\_LFR\_N\_BP\_P1=default value.  SY\_LFR\_N\_CWF\_LONG\_F3 = 1  We also use the timegen software (send CTR in nominal behavior). | | | | | | | | Verify the LFR FSW sends one single data stream whose content corresponds to the NORMAL mode parameter set. |  | |  |
|  | **Use normal\_mode\_parameter\_set\_nominal.py**  Launch the FSW LFR.  Without TC\_LFR\_LOAD\_NORMAL\_PAR, in STANDBY mode , send TC\_LFR\_ENTER\_MODE with CP\_LFR\_MODE= NORMAL.  With HK, ensure the LFR FSW is in the NORMAL mode.  Wait at least 3960s (1.2\*max(SY\_LFR\_N\_SWF\_P, SY\_LFR\_N\_ASM\_P, SY\_LFR\_N\_BP\_P0, SY\_LFR\_N\_BP\_P1).  Go to standby mode  Send TC\_LFR\_LOAD\_NORMAL\_PAR with SY\_LFR\_N\_CWF\_LONG\_F3=1  send TC\_LFR\_ENTER\_MODE with CP\_LFR\_MODE= NORMAL.  With HK, ensure the LFR FSW is in the NORMAL mode.  Wait at least 3960s | | | | | | | | Verify the LFR FSW sends one single data stream whose content corresponds to the NORMAL mode parameter set. |  | |  |
|  | End | | | | | | | |  |  | |  |
| **TEST CASE** | | | | | | | | | | | | | | |
| **Test procedure identifier** | | | | | | **Date** | | | **Assessment** | | | | | |
| SVS-0030\_Ed1 | | | | | | DD/MM/YYYY | | | NT / NA / NOK / POK / OK | | | | | |
| **TS Tested requirements** | | | | | | **RB Tested requirements** | | | | | | | | |
| N/A | | | | | | REQ-LFR-SRS-5504\_Ed1 REQ-LFR-SRS-5548\_Ed1  REQ-LFR-SRS-5500\_Ed1 REQ-LFR-SRS-5514\_Ed1 | | | | | | | | |
| **Component:** | | | |  | | **Version:** | V2 V1 | | | | | | | |
| **Involved subsystems** | | | | | | | | | | | | | | |
| SW: | | 3.2.0.24 | | | | | | | | | | | | |
| HW: | | 1.1.91 StarDundee | | | | | | | | | | | | |
| Start time: | | | (DD/MM/YYYY) hh:mm | | | Test Operator: | |  | | | | | | |
| End Time: | | | (DD/MM/YYYY) hh:mm | | | Test Duration: | | 4mn | | | | | | |
| **Purpose:** | | | Check, in the BURST mode, the LFR FSW sends one single data stream whose content corresponds to the BURST mode parameter set.  The LFR FSW shall handle commands for configuring the different science sub-mode parameter sets.   * BURST mode parameters: TC\_LFR\_LOAD\_BURST\_PAR   The LFR shall handle the following modes  STANDBY NORMAL **BURST** SBM1 SBM2 | | | | | | | | | | | |
| **General remarks about the test** | | | | | | | | | | | | | | |
| Requirement Title | | | Equipment mode management and Equipment configuration management | | | | | | | | | | | |
| dependencies | | | SSS-CP-EQS-270 – REQ-RPW-SYS-0305 SSS-CP-EQS-190 – SSS-CP-EQS-230  SSS-CP-EQS-370 | | | | | | | | | | | |
| restrictions | | |  | | | | | | | | | | | |
| means | | |  | | | | | | | | | | | |
| input data | | |  | | | | | | | | | | | |
| prerequisite | | | LFR is in BURST mode when RPW is in SURVEY\_BURST (\_C) mode; see § “LFR mode management” (§ 5.5.1). | | | | | | | | | | | |
| prerequisite | | |  | | | | | | | | | | | |
| used test programs | | |  | | | | | | | | | | | |
| test script | | | burst\_mode\_parameter\_set.py + long test with timegen | | | | | | | | | | | |
| output data | | |  | | | | | | | | | | | |
| notes | | |  | | | | | | | | | | | |
| **Procedure steps** | | | | | | | | | | | | | | |
| **Step** | **Actions** | | | | **Expected Results** | | | | | | **Comments** | **Status** | | |
|  | Launch the FSW LFR  In standby mode no TC LOAD\_BURST\_PAR sent  Enter in Burst mode with  default values  SY\_LFR\_B\_BP\_P0=1s  SY\_LFR\_B\_BP\_P1=5s  Wait 31.5sec | | | | Verify the LFR FSW sends one single data stream whose content corresponds to the BURST mode parameter set.  TM\_LFR\_SCIENCE\_BURST\_BP1\_F0 and F1  TM\_LFR\_SCIENCE\_BURST\_BP2\_F0 and F1  TM\_LFR\_SCIENCE\_BURST\_CWF\_F2  Use TC\_LFR\_DUMP\_PAR to check parameters configuration | | | | | |  |  | | |
| 2 | In standby mode send TC\_LFR\_LOAD\_BURST\_PAR with these values  SY\_LFR\_B\_BP\_P0=4s  SY\_LFR\_B\_BP\_P1=6s  Enter in Burst Mode  Wait 31.5sec | | | | Verify the LFR FSW sends one single data stream whose content corresponds to the BURST mode parameter set.  TM\_LFR\_EXE\_INCONSISTENT on SY\_LFR\_B\_BP\_P0 parameter (PA\_RPW\_BYTE\_POSITION=10 and PA\_RPW\_RCV\_VALUE=4)  Default values are always set.  TM\_LFR\_SCIENCE\_BURST\_BP1\_F0 and F1  TM\_LFR\_SCIENCE\_BURST\_BP2\_F0 and F1  TM\_LFR\_SCIENCE\_BURST\_CWF\_F2 | | | | | |  |  | | |
| 3 | In standby mode send TC\_LFR\_BURST\_PAR with maximum solicitation = default values  SY\_LFR\_B\_BP\_P0=1s  SY\_LFR\_B\_BP\_P1=5s  Enter in Burst Mode  Wait 31.5s | | | | Verify the LFR FSW sends one single data stream whose content corresponds to the BURST mode parameter set.  Default values are set  TM\_LFR\_SCIENCE\_BURST\_BP1\_F0 and F1  TM\_LFR\_SCIENCE\_BURST\_BP2\_F0 and F1  TM\_LFR\_SCIENCE\_BURST\_CWF\_F2 | | | | | |  |  | | |
| 4 | LFR is in the BURST mode.  Send a TC\_LFR\_BURST\_PAR  Wait 31.5s | | | | Verify the LFR FSW sends one single data stream whose content corresponds to the BURST mode parameter set.  TM\_LFR\_EXE\_NOT\_EXECUTABLE  Values are set by step 3 = Default values  TM\_LFR\_SCIENCE\_BURST\_BP1\_F0 and F1  TM\_LFR\_SCIENCE\_BURST\_BP2\_F0 and F1  TM\_LFR\_SCIENCE\_BURST\_CWF\_F2 | | | | | |  |  | | |
| 5 | In standby mode send TC\_LFR\_BURST\_PAR with SY\_LFR\_B\_BP\_P0=0s  SY\_LFR\_B\_BP\_P0=5s  Enter in Burst Mode  Wait 31.5s | | | | Verify the LFR FSW sends one single data stream whose content corresponds to the BURST mode parameter set.  TM\_LFR\_EXE\_INCONSISTENT on SY\_LFR\_B\_BP\_P0 parameter PA\_RPW\_BYTE\_POSITION=10 and PA\_RPW\_RCV\_VALUE=0.  Values are set by step 3 = Default values  TM\_LFR\_SCIENCE\_BURST\_BP1\_F0 and F1  TM\_LFR\_SCIENCE\_BURST\_BP2\_F0 and F1  TM\_LFR\_SCIENCE\_BURST\_CWF\_F2 | | | | | |  |  | | |
| 6 | In standby mode send TC\_LFR\_BURST\_PAR with SY\_LFR\_B\_BP\_P0=1s  SY\_LFR\_B\_BP\_P0=0s  Enter in Burst Mode  Wait 31.5s | | | | Verify the LFR FSW sends one single data stream whose content corresponds to the BURST mode parameter set.  TM\_LFR\_EXE\_INCONSISTENT on SY\_LFR\_B\_BP\_P1 parameter (PA\_RPW\_BYTE\_POSITION=11 and PA\_RPW\_RCV\_VALUE=0 )  Values are set by step 3 = Default values  TM\_LFR\_SCIENCE\_BURST\_BP1\_F0 and F1  TM\_LFR\_SCIENCE\_BURST\_BP2\_F0 and F1  TM\_LFR\_SCIENCE\_BURST\_CWF\_F2 | | | | | |  |  | | |
| 7 | In standby mode send TC\_LFR\_BURST\_PAR with SY\_LFR\_B\_BP\_P0=1s  SY\_LFR\_B\_BP\_P0=15s  Enter in Burst Mode  Wait 31.5s | | | | Verify the LFR FSW sends one single data stream whose content corresponds to the BURST mode parameter set.  Values are set  TM\_LFR\_SCIENCE\_BURST\_BP1\_F0 and F1  TM\_LFR\_SCIENCE\_BURST\_BP2\_F0 and F1  TM\_LFR\_SCIENCE\_BURST\_CWF\_F2 | | | | | |  |  | | |
| 8 | In standby mode send TC\_LFR\_BURST\_PAR with default values  SY\_LFR\_B\_BP\_P0=1s  SY\_LFR\_B\_BP\_P0=5s  Enter in Burst Mode  Wait 31.5s | | | | Verify the LFR FSW sends one single data stream whose content corresponds to the BURST mode parameter set.  Values are set  TM\_LFR\_SCIENCE\_BURST\_BP1\_F0 and F1  TM\_LFR\_SCIENCE\_BURST\_BP2\_F0 and F1  TM\_LFR\_SCIENCE\_BURST\_CWF\_F2 | | | | | |  |  | | |
| 9 | A long test is done during 14 hours in BURST MODE with defaults values.  SY\_LFR\_B\_BP\_P0=1s  SY\_LFR\_B\_BP\_P0=5s  We also use the timegen software (send CTR in nominal behavior). | | | | Verify the LFR FSW sends one single data stream whose content corresponds to the BURST mode parameter set.  Values are set  TM\_LFR\_SCIENCE\_BURST\_BP1\_F0 and F1  TM\_LFR\_SCIENCE\_BURST\_BP2\_F0 and F1  TM\_LFR\_SCIENCE\_BURST\_CWF\_F2 | | | | | |  |  | | |
| 10 | **End** | | | |  | | | | | |  |  | | |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE** | | | | | | | | | | | |
| **Test procedure identifier** | | | | | | **Date** | | | **Assessment** | | |
| SVS-0031\_Ed1 | | | | | | DD/MM/YYYY | | | NT / NA / NOK / POK / OK | | |
| **TS Tested requirements** | | | | | | **RB Tested requirements** | | | | | |
| N/A | | | | | | REQ-LFR-SRS-5505\_Ed1 REQ-LFR-SRS-5548\_Ed1  REQ-LFR-SRS-5500\_Ed1 REQ-LFR-SRS-5514\_Ed1 | | | | | |
| **Component:** | | | |  | | **Version:** | V2 V1 | | | | |
| **Involved subsystems** | | | | | | | | | | | |
| SW: | | 3.2.0.24 | | | | | | | | | |
| HW: | | 1.1.91 StarDundee | | | | | | | | | |
| Start time: | | | (DD/MM/YYYY) hh:mm | | | Test Operator: | |  | | | |
| End Time: | | | (DD/MM/YYYY) hh:mm | | | Test Duration: | | 1h 05 | | | |
| **Purpose:** | | | Check, in the SBM1 mode, the LFR FSW sends properly data streams.  The LFR FSW shall handle commands for configuring the different science sub-mode parameter sets.   * SBM1 mode parameters: TC\_LFR\_LOAD\_SBM1\_PAR   The LFR shall handle the following modes  STANDBY NORMAL BURST **SBM1** SBM2 | | | | | | | | |
| **General remarks about the test** | | | | | | | | | | | |
| Requirement Title | | | | Equipment mode management and Equipment configuration management | | | | | | | |
| dependencies | | | | SSS-CP-EQS-280, REQ-RPW-SYS-0303 REQ-RPW-SYS-0305 SSS-CP-EQS-190 SSS-CP-EQS-230  SSS-CP-EQS-370 | | | | | | | |
| restrictions | | | |  | | | | | | | |
| means | | | |  | | | | | | | |
| input data | | | |  | | | | | | | |
| prerequisite | | | | LFR is in SBM1 mode when RPW is in SBM\_DETECTION (\_C) mode, and when RPW is in SBM1\_DUMP (\_C); see § “LFR mode management” (§ 5.5.1). | | | | | | | |
| prerequisite | | | |  | | | | | | | |
| used test programs | | | |  | | | | | | | |
| test script | | | | sbm1\_mode\_parameter\_set\_nominal.py (ASM=3600s) + sbm1\_mode\_parameter\_set.py +  long test with timegen | | | | | | | |
| output data | | | |  | | | | | | | |
| notes | | | |  | | | | | | | |
| **Procedure steps** | | | | | | | | | | | |
| **Step** | **Actions** | | | | **Expected Results** | | | | | **Comments** | **Status** |
|  | **Use sbm1\_mode\_parameter\_set\_nominal.py**  Launch the FSW LFR  In standby mode  No TC\_LFR\_LOAD\_NORMAL\_PAR  (default values)  noTC\_LFR\_LOAD\_SBM1\_PAR sent  Enter in SBM1 mode with default values  SY\_LFR\_S1\_BP\_P0=0.25s  SY\_LFR\_S1\_BP\_P1=1s  Wait 3960 sec | | | | Verify the LFR FSW sends a high cadence data stream whose content corresponds to the SBM1 mode parameter set.  For normal mode, we use default values. TM\_LFR\_SCIENCE\_NORMAL\_SWF\_Fi  TM\_LFR\_SCIENCE\_NORMAL\_CWF\_F3  TM\_LFR\_SCIENCE\_NORMAL\_BP1\_Fi  TM\_LFR\_SCIENCE\_NORMAL\_BP2\_Fi  TM\_LFR\_SCIENCE\_NORMAL\_ASM\_Fi  TM\_LFR\_SCIENCE\_SBM1\_CWF\_F1  TM\_LFR\_SCIENCE\_SBM1\_BP1\_F0  TM\_LFR\_SCIENCE\_SBM1\_BP2\_F0 | | | | |  |  |
|  | **Use sbm1\_mode\_parameter\_set.py**  Launch the FSW LFR  In standby mode  TC\_LFR\_LOAD\_NORMAL\_PAR with default values except for SY\_LFR\_N\_ASM\_P=4s  aTC\_LFR\_LOAD\_SBM1\_PAR sent  SY\_LFR\_S1\_BP\_P0=1s  SY\_LFR\_S1\_BP\_P1=4s  Enter in SBM1 mode  Wait 420 sec | | | | Verify the LFR FSW sends a high cadence data stream whose content corresponds to the SBM1 mode parameter set.  For normal mode, we use default values. TM\_LFR\_SCIENCE\_NORMAL\_SWF\_Fi  TM\_LFR\_SCIENCE\_NORMAL\_CWF\_F3  TM\_LFR\_SCIENCE\_NORMAL\_BP1\_Fi  TM\_LFR\_SCIENCE\_NORMAL\_BP2\_Fi  TM\_LFR\_SCIENCE\_NORMAL\_ASM\_Fi  TM\_LFR\_SCIENCE\_SBM1\_CWF\_F1  TM\_LFR\_SCIENCE\_SBM1\_BP1\_F0  TM\_LFR\_SCIENCE\_SBM1\_BP2\_F0 | | | | |  |  |
| 2 | In standby mode TC\_LFR\_LOAD\_SBM1\_PAR sends with default values  SY\_LFR\_S1\_BP\_P0=0.25s  SY\_LFR\_S1\_BP\_P1=1s  And LOAD\_NORMAL\_MODE  2048, 22s, 4s,4,20s  CWF\_LONG\_F3 =0  Enter in SBM1 mode  Wait 235 sec | | | | Verify the LFR FSW sends a high cadence data stream whose content corresponds to the SBM1 mode parameter set.  TM\_LFR\_SCIENCE\_NORMAL\_ASM\_Fi  TM\_LFR\_SCIENCE\_NORMAL\_SWF\_Fi  TM\_LFR\_SCIENCE\_NORMAL\_CWF\_F3  TM\_LFR\_SCIENCE\_NORMAL\_BP1\_Fi  TM\_LFR\_SCIENCE\_NORMAL\_BP2\_Fi  TM\_LFR\_SCIENCE\_SBM1\_CWF\_F1  TM\_LFR\_SCIENCE\_SBM1\_BP1\_F0  TM\_LFR\_SCIENCE\_SBM1\_BP2\_F0 | | | | |  |  |
| 3 | In standby mode TC LOAD\_SBM1\_PAR sends with default values  SY\_LFR\_S1\_BP\_P0=0.25s  SY\_LFR\_S1\_BP\_P1=1s  And LOAD\_NORMAL\_MODE  2048, 22s, 4s,4,20s  CWF\_LONG\_F3= 1  Enter in SBM1 mode  Wait 235sec | | | | Verify the LFR FSW sends a high cadence data stream whose content corresponds to the SBM1 mode parameter set.  Maximum solicitation in Normal mode and SBM1  TM\_LFR\_SCIENCE\_NORMAL\_ASM\_Fi  TM\_LFR\_SCIENCE\_NORMAL\_SWF\_Fi  TM\_LFR\_SCIENCE\_NORMAL\_CWF\_LONG\_F3  TM\_LFR\_SCIENCE\_NORMAL\_BP1\_Fi  TM\_LFR\_SCIENCE\_NORMAL\_BP2\_Fi  TM\_LFR\_SCIENCE\_SBM1\_CWF\_F1  TM\_LFR\_SCIENCE\_SBM1\_BP1\_F0  TM\_LFR\_SCIENCE\_SBM1\_BP2\_F0 | | | | |  |  |
| 4 | In SBM1 mode TC LOAD\_SBM1\_PAR sends  SY\_LFR\_S1\_BP\_P0=4s  SY\_LFR\_S1\_BP\_P1=1s  Wait 235 sec | | | | Verify the LFR FSW sends a high cadence data stream whose content corresponds to the SBM1 mode parameter set.  TM\_LFR\_EXE\_NOT\_EXECUTABLE  Keep the values used in step 3  TM\_LFR\_SCIENCE\_NORMAL\_ASM\_Fi  TM\_LFR\_SCIENCE\_NORMAL\_SWF\_Fi  TM\_LFR\_SCIENCE\_NORMAL\_CWF\_LONG\_F3  TM\_LFR\_SCIENCE\_NORMAL\_BP1\_Fi  TM\_LFR\_SCIENCE\_NORMAL\_BP2\_Fi  TM\_LFR\_SCIENCE\_SBM1\_CWF\_F1  TM\_LFR\_SCIENCE\_SBM1\_BP1\_F0  TM\_LFR\_SCIENCE\_SBM1\_BP2\_F0 | | | | |  |  |
| 5 | In standby mode TC LOAD\_SBM1\_PAR sends with one bad value  SY\_LFR\_S1\_BP\_P0=0  SY\_LFR\_S1\_BP\_P1=1s  Enter in SBM1 mode  Wait 235 sec | | | | TM\_LFR\_EXE\_INCONSISTENT on SY\_LFR\_S1\_BP\_P0 parameter (PA\_RPW\_BYTE\_POSITION=10 and PA\_RPW\_RCV\_VALUE=0)  Keep the values used in step 3  TM\_LFR\_SCIENCE\_NORMAL\_ASM\_Fi  TM\_LFR\_SCIENCE\_NORMAL\_SWF\_Fi  TM\_LFR\_SCIENCE\_NORMAL\_CWF\_LONG\_F3  TM\_LFR\_SCIENCE\_NORMAL\_BP1\_Fi  TM\_LFR\_SCIENCE\_NORMAL\_BP2\_Fi  TM\_LFR\_SCIENCE\_SBM1\_CWF\_F1  TM\_LFR\_SCIENCE\_SBM1\_BP1\_F0  TM\_LFR\_SCIENCE\_SBM1\_BP2\_F0 | | | | |  |  |
| 6 | In standby mode TC LOAD\_SBM1\_PAR sends with one bad value  SY\_LFR\_S1\_BP\_P0=0.25  SY\_LFR\_S1\_BP\_P1=0s  Enter in SBM1 mode  Wait 235 sec | | | | Verify the LFR FSW sends a high cadence data stream whose content corresponds to the SBM1 mode parameter set.  TM\_LFR\_EXE\_NOT\_INCONSISTENT on sy\_lfr\_s1\_bp\_p1 parameter (PA\_RPW\_BYTE\_POSITION=11 and PA\_RPW\_RCV\_VALUE=0).  Keep the values used in step 3  TM\_LFR\_SCIENCE\_NORMAL\_ASM\_Fi  TM\_LFR\_SCIENCE\_NORMAL\_SWF\_Fi  TM\_LFR\_SCIENCE\_NORMAL\_CWF\_LONG\_F3  TM\_LFR\_SCIENCE\_NORMAL\_BP1\_Fi  TM\_LFR\_SCIENCE\_NORMAL\_BP2\_Fi  TM\_LFR\_SCIENCE\_SBM1\_CWF\_F1  TM\_LFR\_SCIENCE\_SBM1\_BP1\_F0  TM\_LFR\_SCIENCE\_SBM1\_BP2\_F0 | | | | |  |  |
| 7 | In standby mode TC LOAD\_SBM1\_PAR sends with default values  SY\_LFR\_S1\_BP\_P0=0.75  SY\_LFR\_S1\_BP\_P1=2s  Enter in SBM1 mode  Wait 235 sec | | | | Verify the LFR FSW sends a high cadence data stream whose content corresponds to the SBM1 mode parameter set.  TM\_LFR\_EXE\_NOT\_INCONSISTENT on sy\_lfr\_s1\_bp\_p0 parameter (PA\_RPW\_BYTE\_POSITION=10 and PA\_RPW\_RCV\_VALUE=3).  3=3\*0.25  Keep the values used in step 3  TM\_LFR\_SCIENCE\_NORMAL\_ASM\_Fi  TM\_LFR\_SCIENCE\_NORMAL\_SWF\_Fi  TM\_LFR\_SCIENCE\_NORMAL\_CWF\_LONG\_F3  TM\_LFR\_SCIENCE\_NORMAL\_BP1\_Fi  TM\_LFR\_SCIENCE\_NORMAL\_BP2\_Fi  TM\_LFR\_SCIENCE\_SBM1\_CWF\_F1  TM\_LFR\_SCIENCE\_SBM1\_BP1\_F0  TM\_LFR\_SCIENCE\_SBM1\_BP2\_F0 | | | | |  |  |
| 8 | In standby mode TC LOAD\_SBM1\_PAR sends with max values  SY\_LFR\_S1\_BP\_P0=0xff  SY\_LFR\_S1\_BP\_P1=0xff  Enter in SBM1 mode  Wait 357 sec | | | | Verify the LFR FSW sends a high cadence data stream whose content corresponds to the SBM1 mode parameter set.  SY\_LFR\_S1\_BP\_P0=63.75s  SY\_LFR\_S1\_BP\_P1=255s  TM\_LFR\_SCIENCE\_NORMAL\_ASM\_Fi  TM\_LFR\_SCIENCE\_NORMAL\_SWF\_Fi  TM\_LFR\_SCIENCE\_NORMAL\_CWF\_LONG\_F3  TM\_LFR\_SCIENCE\_NORMAL\_BP1\_Fi  TM\_LFR\_SCIENCE\_NORMAL\_BP2\_Fi  TM\_LFR\_SCIENCE\_SBM1\_CWF\_F1  TM\_LFR\_SCIENCE\_SBM1\_BP1\_F0  TM\_LFR\_SCIENCE\_SBM1\_BP2\_F0 | | | | |  |  |
| 9 | In standby mode TC LOAD\_NORMAL\_PAR sends with default values  TC LOAD\_SBM1\_PAR sends with default values  Enter in SBM1 mode  Wait 420 sec | | | | Verify the LFR FSW sends a high cadence data stream whose content corresponds to the SBM1 mode parameter set.  All default values of parameters are used (SBM1 +NORMAL)  No ASM because default value is 3600s  TM\_LFR\_SCIENCE\_NORMAL\_SWF\_Fi  TM\_LFR\_SCIENCE\_NORMAL\_CWF\_F3  TM\_LFR\_SCIENCE\_NORMAL\_BP1\_Fi  TM\_LFR\_SCIENCE\_NORMAL\_BP2\_Fi  TM\_LFR\_SCIENCE\_SBM1\_CWF\_F1  TM\_LFR\_SCIENCE\_SBM1\_BP1\_F0  TM\_LFR\_SCIENCE\_SBM1\_BP2\_F0 | | | | |  |  |
| 10 | A long test is done during 3 hours in SBM1 MODE with defaults values.  SY\_LFR\_S1\_BP\_P0=0.25s  SY\_LFR\_S1\_BP\_P0=1s  in NORMAL MODE with defaults values.  SY\_LFR\_N\_SWF\_L=default value  SY\_LFR\_N\_SWF\_P=default value  SY\_LFR\_N\_ASM\_P=default value  SY\_LFR\_N\_BP\_P0=default value  SY\_LFR\_N\_BP\_P1=default value.  SY\_LFR\_N\_CWF\_LONG\_F3 = 0  We also use the timegen software (send CTR in nominal behavior). | | | | Verify the LFR FSW sends a high cadence data stream whose content corresponds to the SBM1 mode parameter set.  All default values of parameters are used (SBM1 +NORMAL)  TM\_LFR\_SCIENCE\_NORMAL\_SWF\_Fi  TM\_LFR\_SCIENCE\_NORMAL\_CWF\_F3  TM\_LFR\_SCIENCE\_NORMAL\_BP1\_Fi  TM\_LFR\_SCIENCE\_NORMAL\_BP2\_Fi  TM\_LFR\_SCIENCE\_NORMAL\_ASM\_Fi  TM\_LFR\_SCIENCE\_SBM1\_CWF\_F1  TM\_LFR\_SCIENCE\_SBM1\_BP1\_F0  TM\_LFR\_SCIENCE\_SBM1\_BP2\_F0 | | | | |  |  |
| 11 | A long test is done during 3 hours in SBM1 MODE with defaults values.  SY\_LFR\_S1\_BP\_P0=0.25s  SY\_LFR\_S1\_BP\_P0=1s  in NORMAL MODE with defaults values.  SY\_LFR\_N\_SWF\_L=default value  SY\_LFR\_N\_SWF\_P=default value  SY\_LFR\_N\_ASM\_P=default value  SY\_LFR\_N\_BP\_P0=default value  SY\_LFR\_N\_BP\_P1=default value.  SY\_LFR\_N\_CWF\_LONG\_F3 = 1  We also use the timegen software (send CTR in nominal behavior). | | | | Verify the LFR FSW sends a high cadence data stream whose content corresponds to the SBM1 mode parameter set.  All default values of parameters are used (SBM1 +NORMAL)  TM\_LFR\_SCIENCE\_NORMAL\_SWF\_Fi  TM\_LFR\_SCIENCE\_NORMAL\_CWF\_LONG\_F3  TM\_LFR\_SCIENCE\_NORMAL\_BP1\_Fi  TM\_LFR\_SCIENCE\_NORMAL\_BP2\_Fi  TM\_LFR\_SCIENCE\_NORMAL\_ASM\_Fi  TM\_LFR\_SCIENCE\_SBM1\_CWF\_F1  TM\_LFR\_SCIENCE\_SBM1\_BP1\_F0  TM\_LFR\_SCIENCE\_SBM1\_BP2\_F0 | | | | |  |  |
| 12 | End | | | |  | | | | |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE** | | | | | | | | | | | |
| **Test procedure identifier** | | | | | | **Date** | | | **Assessment** | | |
| SVS0032\_Ed1 | | | | | | DD/MM/YYYY | | | NT / NA / NOK / POK / OK | | |
| **TS Tested requirements** | | | | | | **RB Tested requirements** | | | | | |
| N/A | | | | | | REQ-LFR-SRS-5506\_Ed1 REQ-LFR-SRS-5548\_Ed1  REQ-LFR-SRS-5500\_Ed1 REQ-LFR-SRS-5514\_Ed1 | | | | | |
| **Component:** | | | |  | | **Version:** | V3 V1 | | | | |
| **Involved subsystems** | | | | | | | | | | | |
| SW: | | 3.2.0.24 | | | | | | | | | |
| HW: | | 1.1.91 Stardundee | | | | | | | | | |
| Start time: | | | (DD/MM/YYYY) hh:mm | | | Test Operator: | |  | | | |
| End Time: | | | (DD/MM/YYYY) hh:mm | | | Test Duration: | | 1h05 | | | |
| **Purpose:** | | | Check, in the SBM2 mode, the LFR FSW sends properly data streams.  The LFR FSW shall handle commands for configuring the different science sub-mode parameter sets.   * SBM2 mode parameters: TC\_LFR\_LOAD\_SBM2\_PAR   The LFR shall handle the following modes  STANDBY NORMAL BURST SBM1 **SBM2** | | | | | | | | |
| **General remarks about the test** | | | | | | | | | | | |
| Requirement Title | | | | Equipment mode management and Equipment configuration management | | | | | | | |
| dependencies | | | | SSS-CP-EQS-290, REQ-RPW-SYS-0303 REQ-RPW-SYS-0305 SSS-CP-EQS-190 SSS-CP-EQS-230 SSS-CP-EQS-370 | | | | | | | |
| restrictions | | | |  | | | | | | | |
| means | | | |  | | | | | | | |
| input data | | | |  | | | | | | | |
| prerequisite | | | | LFR is in SBM2 mode when RPW is in SBM2\_ACQUISITION (\_C); see § “LFR mode management” (§ 5.5.1). | | | | | | | |
| prerequisite | | | |  | | | | | | | |
| used test programs | | | |  | | | | | | | |
| test script | | | | Sbm2\_mode\_parameter\_set\_nominal.py (ASM=3600s) + sbm2\_mode\_parameter\_set.py +  long test with timegen | | | | | | | |
| output data | | | |  | | | | | | | |
| notes | | | | V3 | | | | | | | |
| **Procedure steps** | | | | | | | | | | | |
| **Step** | **Actions** | | | | **Expected Results** | | | | | **Comments** | **Status** |
|  | **Use sbm2\_mode\_parameter\_set\_nominal.py**  Launch the FSW LFR  In standby mode  No TC\_LFR\_LOAD\_NORMAL\_PAR  (default values)  noTC\_LFR\_LOAD\_SBM2\_PAR sent  Enter in SBM1 mode with default values  SY\_LFR\_S2\_BP\_P0=1s  SY\_LFR\_S2\_BP\_P1=5s  Wait 3960 sec. | | | | Verify the LFR FSW sends high cadence data stream whose content corresponds to the SBM2 mode parameter set.  For normal mode, we use default values. TM\_LFR\_SCIENCE\_NORMAL\_SWF\_Fi  TM\_LFR\_SCIENCE\_NORMAL\_CWF\_F3  TM\_LFR\_SCIENCE\_NORMAL\_BP1\_Fi  TM\_LFR\_SCIENCE\_NORMAL\_BP2\_Fi  TM\_LFR\_SCIENCE\_NORMAL\_ASM\_Fi  TM\_LFR\_SCIENCE\_SBM2\_CWF\_F2  TM\_LFR\_SCIENCE\_SBM2\_BP1\_F0  TM\_LFR\_SCIENCE\_SBM2\_BP1\_F1  TM\_LFR\_SCIENCE\_SBM2\_BP2\_F0  TM\_LFR\_SCIENCE\_SBM2\_BP2\_F1 | | | | |  |  |
|  | **Use sbm2\_mode\_parameter\_set.py**  Launch the FSW LFR  In standby mode  TC\_LFR\_LOAD\_NORMAL\_PAR with default values except for SY\_LFR\_N\_ASM\_P=4s  no TC\_LFR\_LOAD\_SBM2\_PAR sent  Enter in SBM2 mode with default values  SY\_LFR\_S2\_BP\_P0=4s  SY\_LFR\_S2\_BP\_P1=20s  Wait 420 sec | | | | Verify the LFR FSW sends high cadence data stream whose content corresponds to the SBM2 mode parameter set.  For normal mode, we use default values. TM\_LFR\_SCIENCE\_NORMAL\_SWF\_Fi  TM\_LFR\_SCIENCE\_NORMAL\_CWF\_F3  TM\_LFR\_SCIENCE\_NORMAL\_BP1\_Fi  TM\_LFR\_SCIENCE\_NORMAL\_BP2\_Fi  TM\_LFR\_SCIENCE\_NORMAL\_ASM\_Fi  TM\_LFR\_SCIENCE\_SBM2\_CWF\_F2  TM\_LFR\_SCIENCE\_SBM2\_BP1\_F0  TM\_LFR\_SCIENCE\_SBM2\_BP1\_F1  TM\_LFR\_SCIENCE\_SBM2\_BP2\_F0  TM\_LFR\_SCIENCE\_SBM2\_BP2\_F1 | | | | |  |  |
|  | In standby mode TC\_LFR\_LOAD\_SBM2\_PAR sends with default values  SY\_LFR\_S2\_BP\_P0=1  SY\_LFR\_S2\_BP\_P1=5s  And LOAD\_NORMAL\_MODE  2048, 22s, 4s,4,20s  CWF\_LONG\_F3 =1  Enter in SBM1 mode  Wait 235 sec | | | | Verify the LFR FSW sends high cadence data stream whose content corresponds to the SBM2 mode parameter set.  For normal mode, we use default values. TM\_LFR\_SCIENCE\_NORMAL\_SWF\_Fi  TM\_LFR\_SCIENCE\_NORMAL\_CWF\_LONG\_F3  TM\_LFR\_SCIENCE\_NORMAL\_BP1\_Fi  TM\_LFR\_SCIENCE\_NORMAL\_BP2\_Fi  TM\_LFR\_SCIENCE\_NORMAL\_ASM\_Fi  TM\_LFR\_SCIENCE\_SBM2\_CWF\_F2  TM\_LFR\_SCIENCE\_SBM2\_BP1\_F0  TM\_LFR\_SCIENCE\_SBM2\_BP1\_F1  TM\_LFR\_SCIENCE\_SBM2\_BP2\_F0  TM\_LFR\_SCIENCE\_SBM2\_BP2\_F1 | | | | |  |  |
|  | In SBM2 mode TC LOAD\_SBM2\_PAR sends  SY\_LFR\_S2\_BP\_P0=4s  SY\_LFR\_S2\_BP\_P1=5s  Wait 235 sec | | | | Verify the LFR FSW sends a high cadence data stream whose content corresponds to the SBM2 mode parameter set.  TM\_LFR\_EXE\_NOT\_EXECUTABLE  Keep the values used in step 3  TM\_LFR\_SCIENCE\_NORMAL\_SWF\_Fi  TM\_LFR\_SCIENCE\_NORMAL\_CWF\_LONG\_F3  TM\_LFR\_SCIENCE\_NORMAL\_BP1\_Fi  TM\_LFR\_SCIENCE\_NORMAL\_BP2\_Fi  TM\_LFR\_SCIENCE\_NORMAL\_ASM\_Fi  TM\_LFR\_SCIENCE\_SBM2\_CWF\_F2  TM\_LFR\_SCIENCE\_SBM2\_BP1\_F0  TM\_LFR\_SCIENCE\_SBM2\_BP1\_F1  TM\_LFR\_SCIENCE\_SBM2\_BP2\_F0  TM\_LFR\_SCIENCE\_SBM2\_BP2\_F1 | | | | |  |  |
|  | In standby mode TC LOAD\_SBM2\_PAR sends with one bad value  SY\_LFR\_S2\_BP\_P0=0  SY\_LFR\_S2\_BP\_P1=5s  Enter in SBM2 mode  Wait 235 sec | | | | Verify the LFR FSW sends a high cadence data stream whose content corresponds to the SBM2 mode parameter set.  TM\_LFR\_EXE\_NOT\_INCONSISTENT on sy\_lfr\_s2\_bp\_p0 parameter (PA\_RPW\_BYTE\_POSITION=10 and PA\_RPW\_RCV\_VALUE=0)  Keep the values used in step 3  TM\_LFR\_SCIENCE\_NORMAL\_SWF\_Fi  TM\_LFR\_SCIENCE\_NORMAL\_CWF\_LONG\_F3  TM\_LFR\_SCIENCE\_NORMAL\_BP1\_Fi  TM\_LFR\_SCIENCE\_NORMAL\_BP2\_Fi  TM\_LFR\_SCIENCE\_NORMAL\_ASM\_Fi  TM\_LFR\_SCIENCE\_SBM2\_CWF\_F2  TM\_LFR\_SCIENCE\_SBM2\_BP1\_F0  TM\_LFR\_SCIENCE\_SBM2\_BP1\_F1  TM\_LFR\_SCIENCE\_SBM2\_BP2\_F0  TM\_LFR\_SCIENCE\_SBM2\_BP2\_F1 | | | | |  |  |
|  | In standby mode TC LOAD\_SBM2\_PAR sends with one bad value  SY\_LFR\_S2\_BP\_P0=1s  SY\_LFR\_S2\_BP\_P1=0s  Enter in SBM2 mode  Wait 235 sec | | | | Verify the LFR FSW sends a high cadence data stream whose content corresponds to the SBM2 mode parameter set.  TM\_LFR\_EXE\_NOT\_INCONSISTENT on sy\_lfr\_s2\_bp\_p1 parameter (PA\_RPW\_BYTE\_POSITION=11 and PA\_RPW\_RCV\_VALUE=0)  Keep the values used in step 3  TM\_LFR\_SCIENCE\_NORMAL\_SWF\_Fi  TM\_LFR\_SCIENCE\_NORMAL\_CWF\_LONG\_F3  TM\_LFR\_SCIENCE\_NORMAL\_BP1\_Fi  TM\_LFR\_SCIENCE\_NORMAL\_BP2\_Fi  TM\_LFR\_SCIENCE\_NORMAL\_ASM\_Fi  TM\_LFR\_SCIENCE\_SBM2\_CWF\_F2  TM\_LFR\_SCIENCE\_SBM2\_BP1\_F0  TM\_LFR\_SCIENCE\_SBM2\_BP1\_F1  TM\_LFR\_SCIENCE\_SBM2\_BP2\_F0  TM\_LFR\_SCIENCE\_SBM2\_BP2\_F1 | | | | |  |  |
|  | In standby mode TC LOAD\_SBM2\_PAR sends with one bad value  SY\_LFR\_S2\_BP\_P0=3s  SY\_LFR\_S2\_BP\_P1=20s  Enter in SBM2 mode  Wait 235 sec | | | | Verify the LFR FSW sends a high cadence data stream whose content corresponds to the SBM2 mode parameter set.  TM\_LFR\_EXE\_NOT\_INCONSISTENT on sy\_lfr\_s2\_bp\_p0 parameter (PA\_RPW\_BYTE\_POSITION=10 and PA\_RPW\_RCV\_VALUE=3)  Keep the values used in step 3  TM\_LFR\_SCIENCE\_NORMAL\_SWF\_Fi  TM\_LFR\_SCIENCE\_NORMAL\_CWF\_LONG\_F3  TM\_LFR\_SCIENCE\_NORMAL\_BP1\_Fi  TM\_LFR\_SCIENCE\_NORMAL\_BP2\_Fi  TM\_LFR\_SCIENCE\_NORMAL\_ASM\_Fi  TM\_LFR\_SCIENCE\_SBM2\_CWF\_F2  TM\_LFR\_SCIENCE\_SBM2\_BP1\_F0  TM\_LFR\_SCIENCE\_SBM2\_BP1\_F1  TM\_LFR\_SCIENCE\_SBM2\_BP2\_F0  TM\_LFR\_SCIENCE\_SBM2\_BP2\_F1 | | | | |  |  |
|  | In standby mode TC LOAD\_SBM2\_PAR sends with max values.  SY\_LFR\_S2\_BP\_P0=0xff  SY\_LFR\_S2\_BP\_P1=0xff  Enter in SBM2 mode  Wait 357 sec | | | | Verify the LFR FSW sends a high cadence data stream whose content corresponds to the SBM2 mode parameter set.  TM\_LFR\_SCIENCE\_NORMAL\_SWF\_Fi  TM\_LFR\_SCIENCE\_NORMAL\_CWF\_LONG\_F3  TM\_LFR\_SCIENCE\_NORMAL\_BP1\_Fi  TM\_LFR\_SCIENCE\_NORMAL\_BP2\_Fi  TM\_LFR\_SCIENCE\_NORMAL\_ASM\_Fi  TM\_LFR\_SCIENCE\_SBM2\_CWF\_F2  TM\_LFR\_SCIENCE\_SBM2\_BP1\_F0  TM\_LFR\_SCIENCE\_SBM2\_BP1\_F1  TM\_LFR\_SCIENCE\_SBM2\_BP2\_F0  TM\_LFR\_SCIENCE\_SBM2\_BP2\_F1 | | | | |  |  |
|  | In standby mode TC LOAD\_NORMAL\_PAR sends with default values  TC LOAD\_SBM2\_PAR sends with default values  Enter in SBM2 mode  Wait 420 sec | | | | Verify the LFR FSW sends a high cadence data stream whose content corresponds to the SBM2 mode parameter set.  No ASM because default value is 3600s  TM\_LFR\_SCIENCE\_NORMAL\_SWF\_Fi  TM\_LFR\_SCIENCE\_NORMAL\_CWF\_F3  TM\_LFR\_SCIENCE\_NORMAL\_BP1\_Fi  TM\_LFR\_SCIENCE\_NORMAL\_BP2\_Fi  TM\_LFR\_SCIENCE\_SBM2\_CWF\_F2  TM\_LFR\_SCIENCE\_SBM2\_BP1\_F0  TM\_LFR\_SCIENCE\_SBM2\_BP1\_F1  TM\_LFR\_SCIENCE\_SBM2\_BP2\_F0  TM\_LFR\_SCIENCE\_SBM2\_BP2\_F1 | | | | |  |  |
|  | A long test is done during 3 hours in SBM2 MODE with defaults values.  SY\_LFR\_S2\_BP\_P0=1s  SY\_LFR\_S2\_BP\_P0=5s  in NORMAL MODE with defaults values.  SY\_LFR\_N\_SWF\_L=default value  SY\_LFR\_N\_SWF\_P=default value  SY\_LFR\_N\_ASM\_P=default value  SY\_LFR\_N\_BP\_P0=default value  SY\_LFR\_N\_BP\_P1=default value.  SY\_LFR\_N\_CWF\_LONG\_F3 = 0  We also use the timegen software (send CTR in nominal behavior). | | | | Verify the LFR FSW sends a high cadence data stream whose content corresponds to the SBM2 mode parameter set.  TM\_LFR\_SCIENCE\_NORMAL\_SWF\_Fi  TM\_LFR\_SCIENCE\_NORMAL\_CWF\_F3  TM\_LFR\_SCIENCE\_NORMAL\_BP1\_Fi  TM\_LFR\_SCIENCE\_NORMAL\_BP2\_Fi  TM\_LFR\_SCIENCE\_NORMAL\_ASM\_Fi  TM\_LFR\_SCIENCE\_SBM2\_CWF\_F2  TM\_LFR\_SCIENCE\_SBM2\_BP1\_F0  TM\_LFR\_SCIENCE\_SBM2\_BP1\_F1  TM\_LFR\_SCIENCE\_SBM2\_BP2\_F0  TM\_LFR\_SCIENCE\_SBM2\_BP2\_F1 | | | | |  |  |
|  | A long test is done during 3 hours in SBM2 MODE with defaults values.  SY\_LFR\_S2\_BP\_P0=1s  SY\_LFR\_S2\_BP\_P0=5s  in NORMAL MODE with defaults values.  SY\_LFR\_N\_SWF\_L=default value  SY\_LFR\_N\_SWF\_P=default value  SY\_LFR\_N\_ASM\_P=default value  SY\_LFR\_N\_BP\_P0=default value  SY\_LFR\_N\_BP\_P1=default value.  SY\_LFR\_N\_CWF\_LONG\_F3 = 1  We also use the timegen software (send CTR in nominal behavior). | | | | Verify the LFR FSW sends a high cadence data stream whose content corresponds to the SBM2 mode parameter set.  TM\_LFR\_SCIENCE\_NORMAL\_SWF\_Fi  TM\_LFR\_SCIENCE\_NORMAL\_CWF\_F3  TM\_LFR\_SCIENCE\_NORMAL\_BP1\_Fi  TM\_LFR\_SCIENCE\_NORMAL\_BP2\_Fi  TM\_LFR\_SCIENCE\_NORMAL\_ASM\_Fi  TM\_LFR\_SCIENCE\_SBM2\_CWF\_F2  TM\_LFR\_SCIENCE\_SBM2\_BP1\_F0  TM\_LFR\_SCIENCE\_SBM2\_BP1\_F1  TM\_LFR\_SCIENCE\_SBM2\_BP2\_F0  TM\_LFR\_SCIENCE\_SBM2\_BP2\_F1 | | | | |  |  |
|  | End | | | |  | | | | |  |  |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE** | | | | | | | | | | |
| **Test procedure identifier** | | | | | **Date** | | | **Assessment** | | |
| SVS-0033\_Ed1 | | | | | DD/MM/YYYY | | | NT / NA / NOK / POK / OK | | |
| **TS Tested requirements** | | | | | **RB Tested requirements** | | | | | |
| N/A | | | | | REQ-LFR-SRS-5507\_Ed1 REQ-LFR-SRS-5508\_Ed1 | | | | | |
| **Component:** | | |  | | **Version:** | V1 | | | | |
| **Involved subsystems** | | | | | | | | | | |
| SW: | 3.2.0.24 | | | | | | | | | |
| HW: | 1.1.91 Stardundee | | | | | | | | | |
| Start time: | | (DD/MM/YYYY) hh:mm | | | Test Operator: | |  | | | |
| End Time: | | (DD/MM/YYYY) hh:mm | | | Test Duration: | | 2h30 | | | |
| **Purpose:** | | Check mode transitions correspond to the activation by the LFR FSW of the set of configuration parameters corresponding to the final mode. | | | | | | | | |
| **General remarks about the test** | | | | | | | | | | |
| Requirement Title | | Equipment mode management | | | | | | | | |
| dependencies | | SSS-CP-EQS-300 SSS-CP-EQS-310 | | | | | | | | |
| restrictions | |  | | | | | | | | |
| means | |  | | | | | | | | |
| input data | |  | | | | | | | | |
| prerequisite | |  | | | | | | | | |
| used test programs | |  | | | | | | | | |
| test script | | transition\_activation\_4s.py then analyzeCuc() to check | | | | | | | | |
| output data | |  | | | | | | | | |
| notes | | Note TM\_LFR\_PARAMETER\_DUMP (after TC\_LFR\_DUMP\_PAR) provides the parameters values.  For each step, wait a convincing delay. For each mode, theoretical delay is T=max(Time between two snapshots, time between two ASM, time between two products, ...); concrete delay should be >2\*T.  TC\_LFR\_ENTER\_MODE with CP\_LFR\_MODE=current mode are tested here. | | | | | | | | |
| **Procedure steps** | | | | | | | | | | |
| **Step** | **Actions** | | | **Expected Results** | | | | | **Comments** | **Status** |
|  | Launch the LFR FSW. Ensure the mode is STANDBY.  TC\_LFR\_LOAD\_NORMAL\_PAR is sent with SY\_LFR\_N\_ASM\_P= 4s  Realize the STANDBY->NORMAL modes transition.  Ensure the current mode is NORMAL  Send TC\_LFR\_DUMP\_PAR.  Wait sufficient delay (see note above). | | | Verify the data stream is compliant with default values of NORMAL mode.  In TM\_LFR\_PARAMETER\_DUMP, verify parameters are consistent with current mode. | | | | |  |  |
|  | Realize the STANDBY->BURST modes transition.  Ensure the current mode is BURST.  Send TC\_LFR\_DUMP\_PAR.  Wait sufficient delay (see note above). | | | Verify the data stream is compliant with default values of BURST mode.  In TM\_LFR\_PARAMETER\_DUMP, verify parameters are consistent with current mode. | | | | |  |  |
|  | Realize the STANDBY->SBM1 modes transition.  Ensure the current mode is SBM1.  Send TC\_LFR\_DUMP\_PAR.  Wait sufficient delay (see note above). | | | Verify the data stream is compliant with default values of SBM1 mode.  In TM\_LFR\_PARAMETER\_DUMP, verify parameters are consistent with current mode. | | | | |  |  |
|  | Realize the STANDBY->SBM2 modes transition.  Ensure the current mode is SBM2.  Send TC\_LFR\_DUMP\_PAR.  Wait sufficient delay (see note above). | | | Verify the data stream is compliant with default values of SBM2 mode.  In TM\_LFR\_PARAMETER\_DUMP, verify parameters are consistent with current mode. | | | | |  |  |
|  | Realize the SBM2->NORMAL modes transition.  Ensure the current mode is NORMAL.  Send TC\_LFR\_DUMP\_PAR.  Wait sufficient delay (see note above). | | | Verify the data stream is compliant with default values of NORMAL mode.  In TM\_LFR\_PARAMETER\_DUMP, verify parameters are consistent with current mode. | | | | |  |  |
|  | Realize the NORMAL ->NORMAL modes transition.  Ensure the current mode is NORMAL.  Send TC\_LFR\_DUMP\_PAR.  Wait sufficient delay (see note above). | | | Verify the data stream is compliant with default values of NORMAL mode.  In TM\_LFR\_PARAMETER\_DUMP, verify parameters are consistent with current mode. | | | | |  |  |
|  | Realize the NORMAL ->BURST modes transition.  Ensure the current mode is BURST.  Send TC\_LFR\_DUMP\_PAR.  Wait sufficient delay (see note above). | | | Verify the data stream is compliant with default values of BURST mode.  In TM\_LFR\_PARAMETER\_DUMP, verify parameters are consistent with current mode. | | | | |  |  |
|  | Realize the BURST -> NORMAL modes transition.  Ensure the current mode is NORMAL.  Send TC\_LFR\_DUMP\_PAR.  Wait sufficient delay (see note above). | | | Verify the data stream is compliant with default values of NORMAL mode.  In TM\_LFR\_PARAMETER\_DUMP, verify parameters are consistent with current mode. | | | | |  |  |
|  | Realize the NORMAL ->SBM1 modes transition.  Ensure the current mode is SBM1.  Send TC\_LFR\_DUMP\_PAR.  Wait sufficient delay (see note above). | | | Verify the data stream is compliant with default values of SBM1 mode.  In TM\_LFR\_PARAMETER\_DUMP, verify parameters are consistent with current mode. | | | | |  |  |
|  | Realize the SBM1-> NORMAL modes transition.  Ensure the current mode is NORMAL.  Send TC\_LFR\_DUMP\_PAR.  Wait sufficient delay (see note above). | | | Verify the data stream is compliant with default values of NORMAL mode.  In TM\_LFR\_PARAMETER\_DUMP, verify parameters are consistent with current mode. | | | | |  |  |
|  | Realize the NORMAL ->SBM2 modes transition.  Ensure the current mode is SBM2.  Send TC\_LFR\_DUMP\_PAR.  Wait sufficient delay (see note above). | | | Verify the data stream is compliant with default values of SBM2 mode.  In TM\_LFR\_PARAMETER\_DUMP, verify parameters are consistent with current mode. | | | | |  |  |
|  | Realize the SBM2->BURST modes transition.  Ensure the current mode is BURST.  Send TC\_LFR\_DUMP\_PAR.  Wait sufficient delay (see note above). | | | Verify the data stream is compliant with default values of BURST mode.  In TM\_LFR\_PARAMETER\_DUMP, verify parameters are consistent with current mode. | | | | |  |  |
|  | Realize the BURST-> BURST modes transition.  Ensure the current mode is BURST.  Send TC\_LFR\_DUMP\_PAR.  Wait sufficient delay (see note above). | | | Verify the data stream is compliant with default values of BURST mode.  In TM\_LFR\_PARAMETER\_DUMP, verify parameters are consistent with current mode. | | | | |  |  |
|  | Realize the BURST-> SBM1 modes transition.  Ensure the current mode is SBM1.  Send TC\_LFR\_DUMP\_PAR.  Wait sufficient delay (see note above). | | | Verify the data stream is compliant with default values of SBM1mode.  In TM\_LFR\_PARAMETER\_DUMP, verify parameters are consistent with current mode. | | | | |  |  |
|  | Realize the SBM1-> BURST modes transition.  Ensure the current mode is BURST.  Send TC\_LFR\_DUMP\_PAR.  Wait sufficient delay (see note above). | | | Verify the data stream is compliant with default values of BURST mode.  In TM\_LFR\_PARAMETER\_DUMP, verify parameters are consistent with current mode. | | | | |  |  |
|  | Realize the BURST->SBM2 modes transition.  Ensure the current mode is SBM2.  Send TC\_LFR\_DUMP\_PAR.  Wait sufficient delay (see note above). | | | Verify the data stream is compliant with default values of SBM2mode.  In TM\_LFR\_PARAMETER\_DUMP, verify parameters are consistent with current mode. | | | | |  |  |
|  | Realize the SBM2->SBM1 modes transition.  Ensure the current mode is SBM1.  Send TC\_LFR\_DUMP\_PAR.  Wait sufficient delay (see note above). | | | Verify the data stream is compliant with default values of SBM1mode.  In TM\_LFR\_PARAMETER\_DUMP, verify parameters are consistent with current mode. | | | | |  |  |
|  | Realize the SBM1-> SBM1 modes transition.  Ensure the current mode is SBM1.  Send TC\_LFR\_DUMP\_PAR.  Wait sufficient delay (see note above). | | | Verify the data stream is compliant with default values of SBM1 mode.  In TM\_LFR\_PARAMETER\_DUMP, verify parameters are consistent with current mode. | | | | |  |  |
|  | Realize the SBM1->SBM2 modes transition.  Ensure the current mode is SBM2.  Send TC\_LFR\_DUMP\_PAR.  Wait sufficient delay (see note above). | | | Verify the data stream is compliant with default values of SBM2mode.  In TM\_LFR\_PARAMETER\_DUMP, verify parameters are consistent with current mode. | | | | |  |  |
|  | Realize the SBM2->SBM2 modes transition.  Ensure the current mode is SBM2.  Send TC\_LFR\_DUMP\_PAR.  Wait sufficient delay (see note above). | | | Verify the data stream is compliant with default values of SBM2 mode.  In TM\_LFR\_PARAMETER\_DUMP, verify parameters are consistent with current mode. | | | | |  |  |
|  | Realize the SBM2->STANDBY modes transition.  Ensure the current mode is SBM2.  Wait sufficient delay (see note above). | | | Verify there’s no science data stream. | | | | |  |  |
|  | End | | |  | | | | |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE** | | | | | | | | | | | | |
| **Test procedure identifier** | | | | | | | **Date** | | | **Assessment** | | |
| SVS-0034\_Ed2 | | | | | | | DD/MM/YYYY | | | NT / NA / NOK / POK / OK | | |
| **TS Tested requirements** | | | | | | | **RB Tested requirements** | | | | | |
| N/A | | | | | | | REQ-LFR-SRS-5509\_Ed1 REQ-LFR-SRS-5508\_Ed1  REQ-LFR-SRS-5592\_Ed1 | | | | | |
| **Component:** | | | | |  | | **Version:** | V2 | | | | |
| **Involved subsystems** | | | | | | | | | | | | |
| SW: | | | | 3.2.0.24 | | | | | | | | |
| HW: | | | | 1.1.91 Stardundee | | | | | | | | |
| Start time: | | (DD/MM/YYYY) hh:mm | | | | | Test Operator: | |  | | | |
| End Time: | | (DD/MM/YYYY) hh:mm | | | | | Test Duration: | | 2h02 + 10mn | | | |
| **Purpose:** | | Check the synchronization of the LFR mode transition on the due date, using its internal time.  The equipment flight software shall ignore the synchronization bit of the CP\_LFR\_ENTER\_MODE\_TIME parameter (most significant bit) when they handle a TC\_LFR\_ENTER\_MODE command. | | | | | | | | | | |
| **General remarks about the test** | | | | | | | | | | | | |
| Requirement Title | | | Equipment mode management | | | | | | | | | |
| dependencies | | | Data time-stamping mechanism, SSS-CP-EQS-320 SSS-CP-EQS-310 SSS-CP-EQS-328 | | | | | | | | | |
| restrictions | | |  | | | | | | | | | |
| means | | |  | | | | | | | | | |
| input data | | |  | | | | | | | | | |
| prerequisite | | |  | | | | | | | | | |
| used test programs | | |  | | | | | | | | | |
| test script | | | activateLfrModeTiming.py (step 1&2)  activateLfrModeNominal Cuc (step 3&4). | | | | | | | | | |
| output data | | |  | | | | | | | | | |
| notes | | | For the internal time, the synchronized/de-synchronized cases are treated consecutively. | | | | | | | | | |
| **Procedure steps** | | | | | | | | | | | | |
| **Step** | **Actions** | | | | | **Expected Results** | | | | | **Comments** | **Status** |
|  | Ensure the LFR FSW is synchronized (MSB of the TIME fields is 0).  For that, we send one TC\_LFR\_UPDATE\_TIME + a valid time code .  Go to SBM2 mode and  Realize all the allowed transitions. | | | | | For each transition:  Verify the TM\_LFR\_SCIENCE\_\* are compliant with LFR mode.  With PA\_LFR\_ACQUISITION\_TIME field of CWF products, verify transition occurs on the due date. | | | | |  |  |
|  | Ensure the FSW LFR is not synchronized (MSB of the TIME fields is 1).  For that, we wait during SY\_RPW\_DELAY\_WITHOUT\_CTR\*1,1s  Go to SBM2 made and  Realize all the allowed transitions. | | | | | For each transition:  Verify the TM\_LFR\_SCIENCE\_\* are compliant with LFR mode.  With PA\_LFR\_ACQUISITION\_TIME field of CWF products, verify transition occurs on the due date. | | | | |  |  |
|  | MSB of CP\_LFR\_ENTER\_MODE\_TIME = 0  The CP\_LFR\_ENTER\_MODE\_TIME is the interne time + delay; realize all the allowed transitions with msb=0 for delay=1, 2, 3, 4s  the fine time of the CP\_LFR\_ENTER\_MODE\_TIME is 0. | | | | | Check the synchronization of the LFR mode transition on the due date.  For delay 1,2,3, all mode transitions are accepted.  For delay > 3s , the transition is refused and a TM\_LFR\_TC\_EXE\_NOT\_EXECUTABLE is generated.  HK\_LFR\_MODE of TM\_LFR\_TC\_EXE\_NOT\_EXECUTABLE showns the current working mode and  not the asked mode.  The MSB bit of the CP\_LFR\_ENTER\_MODE\_TIME has no-effect level.  Transition at due date is realized.  LFR FSW executes the mode transition immediately.  HK\_LFR\_MODE of TM\_LFR\_TC\_EXE\_SUCCESS is equal to the asked mode. | | | | | REQ-LFR-SRS-5592\_Ed1 |  |
|  | MSB of CP\_LFR\_ENTER\_MODE\_TIME = 1  The CP\_LFR\_ENTER\_MODE\_TIME is the interne time + delay; realize all the allowed transitions with msb=1 for delay=1, 2, 3, 4s  .  the fine time of the CP\_LFR\_ENTER\_MODE\_TIME is 0. | | | | | Check the synchronization of the LFR mode transition on the due date..  For delay > 3s , the transition is refused and a  TM\_LFR\_TC\_EXE\_NOT\_EXECUTABLE is generated.  HK\_LFR\_MODE of TM\_LFR\_TC\_EXE\_NOT\_EXECUTABLE showns the current working mode and  not the asked mode.  The MSB bit of the CP\_LFR\_ENTER\_MODE\_TIME has no-effect level.  Transition at due date is realized.  LFR FSW executes the mode transition immediately.  HK\_LFR\_MODE of TM\_LFR\_TC\_EXE\_SUCCESS is equal to the asked mode. | | | | | REQ-LFR-SRS-5592\_Ed1 |  |
|  | End | | | | |  | | | | |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE** | | | | | | | | | | | | | |
| **Test procedure identifier** | | | | | | | | **Date** | | | **Assessment** | | |
| SVS-0035\_Ed1 | | | | | | | | DD/MM/YYYY | | | NT / NA / NOK / POK / OK | | |
| **TS Tested requirements** | | | | | | | | **RB Tested requirements** | | | | | |
| N/A | | | | | | | | REQ-LFR-SRS-5511\_Ed1 REQ-LFR-SRS-5512\_Ed1 | | | | | |
| **Component:** | | | | | |  | | **Version:** |  | | | | |
| **Involved subsystems** | | | | | | | | | | | | | |
| SW: | | | | 3.2.0.24 | | | | | | | | | |
| HW: | | | | 1.1.91 StarDundee | | | | | | | | | |
| Start time: | | | | | (DD/MM/YYYY) hh:mm | | | Test Operator: | |  | | | |
| End Time: | | | | | (DD/MM/YYYY) hh:mm | | | Test Duration: | |  | | | |
| **Purpose:** | | Check LFR FSW synchronizes the continuous waveform acquisitions on 1 s, with a precision higher than 500 μs, by using the SpW time code.  For snapshot waveforms : check that LFR FSW centers, with a precision higher than 1 ms, this time  window on the reference time that is transmitted as a parameter of the mode activating command. | | | | | | | | | | | |
| **General remarks about the test** | | | | | | | | | | | | | |
| Requirement Title | | | Acquisition synchronization | | | | | | | | | | |
| dependencies | | | SSS-CP-EQS-340 and SSS-CP-EQS-350 REQ-RPW-SYS-0350, REQ-RPW-SYS-0351 | | | | | | | | | | |
| restrictions | | |  | | | | | | | | | | |
| means | | |  | | | | | | | | | | |
| input data | | |  | | | | | | | | | | |
| prerequisite | | |  | | | | | | | | | | |
| used test programs | | | time\_tests\_cwf.pro + time\_tests\_swf.pro (T.Chust IDL programs) | | | | | | | | | | |
| test script | | | CTC-512 (SSS-CP-EQS-340) and CTC-200 (SSS-CP-EQS-350) | | | | | | | | | | |
| output data | | |  | | | | | | | | | | |
| notes | | |  | | | | | | | | | | |
| **Procedure steps** | | | | | | | | | | | | | |
| **Step** | **Actions** | | | | | | **Expected Results** | | | | | **Comments** | **Status** |
|  | Realize a mode transition terminating in **NORMAL** mode: realize an acquisition**.** | | | | | | Ensure that the frame rate is consistent with the Normal mode parameters :   * Verify the continuous waveform acquisitions are synchronized on the second, with a precision higher than 500 μs (SSS-CP-EQS-340). * Verify that the snapshot waveforms time windows (F0, F1 and F2) are centered with a precision higher than 1ms and correctly positioned compared to the reference time that is transmitted as a parameter of mode activating command (SSS-CP-EQS-350). | | | | |  |  |
|  | Realize a mode transition terminating in **BURST** mode: realize an acquisition**.** | | | | | | Ensure that the frame rate is consistent with the Burst mode parameters.  Ensure that the frame rate is consistent with the Normal mode parameters :   * Verify the continuous waveform acquisitions are synchronized on the second, with a precision higher than 500 μs (SSS-CP-EQS-340). | | | | |  |  |
|  | Realize a mode transition terminating in **SBM1** mode; realize an acquisition**.** | | | | | | Ensure that the frame rate is consistent with the SBM1 mode parameters.   * Verify the continuous waveform acquisitions are synchronized on the second, with a precision higher than 500 μs (SSS-CP-EQS-340). * Verify that the snapshot waveforms time windows (F0, F1 and F2) are centered with a precision higher than 1ms and correctly positioned compared to the reference time that is transmitted as a parameter of mode activating command (SSS-CP-EQS-350). | | | | |  |  |
|  | Realize a mode transition terminating in **SBM2** mode; realize an acquisition**.** | | | | | | Ensure that the frame rate is consistent with the SBM2 mode parameters.   * Verify the continuous waveform acquisitions are synchronized on the second, with a precision higher than 500 μs (SSS-CP-EQS-340). * Verify that the snapshot waveforms time windows (F0, F1 and F2) are centered with a precision higher than 1ms and correctly positioned compared to the reference time that is transmitted as a parameter of mode activating command (SSS-CP-EQS-350). | | | | |  |  |
|  | End | | | | | |  | | | | |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE** | | | | | | | | | | | |
| **Test procedure identifier** | | | | | **Date** | | | | **Assessment** | | |
| SVS-0036\_Ed1 | | | | | DD/MM/YYYY | | | | NT / NA / NOK / POK / OK | | |
| **TS Tested requirements** | | | | | **RB Tested requirements** | | | | | | |
| N/A | | | | | REQ-LFR-SRS-5553\_Ed1 | | | | | | |
| **Component:** | | | |  | **Version:** | | V2 | | | | |
| **Involved subsystems** | | | | | | | | | | | |
| SW: | | 3.2.0.24 | | | | | | | | | |
| HW: | | 1.1.91 Stardundee | | | | | | | | | |
| Start time: | | | (DD/MM/YYYY) hh:mm | | Test Operator: | | |  | | | |
| End Time: | | | (DD/MM/YYYY) hh:mm | | Test Duration: | | | 4mn 30 | | | |
| **Purpose:** | | | Check the LFR FSW is able to receive and process the shared status/HK at regular time intervals (≤ 2000 ms) as TC\_LFR\_UPDATE\_INFO packets. | | | | | | | | |
| **General remarks about the test** | | | | | | | | | | | |
| Requirement Title | | | Inter-equipment shared data  TC\_xxx\_UPDATE\_INFO packet. SSS-CP-EQS-351 | | | | | | | | |
| dependencies | | | SSS-CP-DAS-544: When its transmission is enabled, in all the modes except STANDBY, the DAS shall generate every SY\_DPU\_INFO\_PKT\_GEN\_PER milliseconds (≤ 2000), toward the analyzers (LFR, THR, TDS), a TC\_xxx\_UPDATE\_INFO packet. | | | | | | | | |
| restrictions | | |  | | | | | | | | |
| means | | |  | | | | | | | | |
| input data | | |  | | | | | | | | |
| prerequisite | | | REQ-LFR-SRS-5564\_Ed1 | | | | | | | | |
| used test programs | | |  | | | | | | | | |
| test script | | | update\_info.py | | | | | | | | |
| output data | | |  | | | | | | | | |
| notes | | | No mode constraint and no delay constraint. | | | | | | | | |
| **Procedure steps** | | | | | | | | | | | |
| **Step** | **Actions** | | | | | **Expected Results** | | | | **Comments** | **Status** |
|  | For each mode  For delay 0.5,1,1.5,2,2.5  Generate a TC\_LFR\_UPDATE\_INFO shared status/HK packet. | | | | | Verify the LFR FSW receives and processes the TC\_LFR\_UPDATE\_INFO packet.  In standby mode, LFR accepts TC\_LFR\_UPDATE\_INFO. | | | |  |  |
|  | End | | | | |  | | | |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE** | | | | | | | | | | | |
| **Test procedure identifier** | | | | | | **Date** | | | **Assessment** | | |
| SVS-0037\_Ed1 | | | | | | DD/MM/YYYY | | | NT / NA / NOK / POK / OK | | |
| **TS Tested requirements** | | | | | | **RB Tested requirements** | | | | | |
| N/A | | | | | | REQ-LFR-SRS-5541\_Ed1 and REQ-LFR-5542\_Ed1 | | | | | |
| **Component:** | | | |  | | **Version:** | V2 | | | | |
| **Involved subsystems** | | | | | | | | | | | |
| SW: | | 3.2.0.24 | | | | | | | | | |
| HW: | | 1.1.91 Stardundee | | | | | | | | | |
| Start time: | | | (DD/MM/YYYY) hh:mm | | | Test Operator: | |  | | | |
| End Time: | | | (DD/MM/YYYY) hh:mm | | | Test Duration: | |  | | | |
| **Purpose:** | | | The LFR FSW shall also report anomalies and errors (including SpaceWire errors)  by using its periodic HK report and by managing the following parameters:  - Low level error counter: incremented each time a new low level error / anomaly is detected     by the equipment flight software.  - Medium level error counter: incremented each time a new medium level error / anomaly is     detected by the equipment flight software.  - Last error report id: the error report id corresponds to the category of the error (AHB,     SpaceWire, Buffer management, etc.); the last error report id parameter contains the id of    the last error that has occurred.  - Last error code: in a given error category, each error or anomaly that can occur is identified     by an error code; the last error code parameter contains the code of the last error that has    occurred.  - Time of the last error.   - Individual error counters: each individual error identified by its category and its code is     associated to a specific error counter.  The LFR FSW shall report the 2 levels of severity: low and medium . | | | | | | | | |
| **General remarks about the test** | | | | | | | | | | | |
| Requirement Title | | | Inter-equipment shared data | | | | | | | | |
| dependencies | | | SSS-CP-EQS-120 and SSS-CP-EQS-130 | | | | | | | | |
| restrictions | | |  | | | | | | | | |
| means | | |  | | | | | | | | |
| input data | | |  | | | | | | | | |
| prerequisite | | |  | | | | | | | | |
| used test programs | | | spw\_failure.py (SVS-0064) starting\_time.py (SVS-0011)  starting\_time\_set\_time.py starting\_time\_start.py starting\_time\_after.py (SVS-0057)  update\_timecode\_missing\_cnt\_wrap.py update\_time\_and\_info\_not\_synchro\_cnt\_wrap.py  update\_timecode\_it\_and\_ctr\_cnt\_wrap  check\_rx\_too\_big\_cnt\_wrap (SVS-0026) | | | | | | | | |
| test script | | |  | | | | | | | | |
| output data | | |  | | | | | | | | |
| notes | | |  | | | | | | | | |
| **Procedure steps** | | | | | | | | | | | |
| **Step** | **Actions** | | | | **Expected Results** | | | | | **Comments** | **Status** |
| 1 | Low error management | | | | HK\_LFR\_LE\_CNT is the sum of these errors counters:  spacewire errors  HK\_LFR\_DPU\_SPW\_PARITY, HK\_LFR\_DPU\_SPW\_DISCONNECT, HK\_LFR\_DPU\_SPW\_ESCAPE  HK\_LFR\_DPU\_SPW\_CREDIT  HK\_LFR\_DPU\_SPW\_WRITE\_SYNC,  specific counters  HK\_LFR\_AHB\_CORRECTABLE  timing  HK\_LFR\_TIMECODE\_ERRONEOUS HK\_LFR\_TIMECODE\_MISSING HK\_LFR\_TIMECODE\_INVALID HK\_LFR\_TIME\_TIMECODE\_IT  HK\_LFR\_TIME\_NOT\_SYNCHRO  HK\_LFR\_TIME\_TIMECODE\_CTR  **The last error detected updates these fields**  HK\_LFR\_LAST\_ER\_RID  HK\_LFR\_ER\_CODE  HK\_LFR\_LAST\_ER\_TIME | | | | | See  SVS-0064  spw\_failure.py  No test  See SVS-0011  and starting\_time\_\*.py (SVS-0057)  and  update\_timecode\_\*  (SVS-0026) |  |
| 2 | medium error management | | | | HK\_LFR\_ME\_CNT is the sum of these errors counters:  spacewire errors  HK\_LFR\_DPU\_SPW\_EARLY\_EOP, HK\_LFR\_DPU\_SPW\_INVALID\_ADDR, HK\_LFR\_DPU\_SPW\_EEP, HK\_LFR\_DPU\_SPW\_RX\_TOO\_BIG  The last error detected updates these fields  HK\_LFR\_LAST\_ER\_RID  HK\_LFR\_ER\_CODE  HK\_LFR\_LAST\_ER\_TIME | | | | | See  SVS-0064  spw\_failure.py  and  check\_rx\_too\_big\_cnt\_wrap (SVS-0026) |  |
|  | End | | | |  | | | | |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE** | | | | | | | | | | | |
| **Test procedure identifier** | | | | | | **Date** | | | **Assessment** | | |
| SVS-0038\_Ed1 | | | | | | DD/MM/YYYY | | | NT / NA / NOK / POK / OK | | |
| **TS Tested requirements** | | | | | | **RB Tested requirements** | | | | | |
| N/A | | | | | | REQ-LFR-SRS-5555\_Ed1 | | | | | |
| **Component:** | | | |  | | **Version:** | V2 | | | | |
| **Involved subsystems** | | | | | | | | | | | |
| SW: | | 3.2.0.24 | | | | | | | | | |
| HW: | | 1.1.91 Stardundee | | | | | | | | | |
| Start time: | | | (DD/MM/YYYY) hh:mm | | | Test Operator: | |  | | | |
| End Time: | | | (DD/MM/YYYY) hh:mm | | | Test Duration: | | 45mn | | | |
| **Purpose:** | | | Check the LFR FSW extracts the relevant parameters from the shared status/HK packets for inserting them into their scientific TM packets. | | | | | | | | |
| **General remarks about the test** | | | | | | | | | | | |
| Requirement Title | | | | Inter-equipment shared data | | | | | | | |
| dependencies | | | | SSS-CP-EQS-353, REQ-RPW-SYS-160 | | | | | | | |
| restrictions | | | |  | | | | | | | |
| means | | | |  | | | | | | | |
| input data | | | |  | | | | | | | |
| prerequisite | | | |  | | | | | | | |
| used test programs | | | |  | | | | | | | |
| test script | | | | 5 scripts, one by mode update\_info\_xxx.py (normal\_long, long,sbm1,sbm2) | | | | | | | |
| output data | | | |  | | | | | | | |
| notes | | | | The procedure step described actions for each mode | | | | | | | |
| **Procedure steps** | | | | | | | | | | | |
| **Step** | **Actions** | | | | **Expected Results** | | | | | **Comments** | **Status** |
|  | For NORMAL mode, SBM1 mode and SBM2 mode  send a TC\_LFR\_LOAD\_NORMAL\_PAR  to set SY\_LFR\_N\_ASM\_P=4s  Enter into the mode  Send a TC\_LFR\_UPDATE\_INFO  With CP\_DPU\_BIAS\_ON\_OFF = 1  Wait the delay to see all TM\_LFR\_SCIENCE | | | | Verify the LFR FSW extracts the relevant parameters from the shared status/HK packets for inserting them into their scientific TM packets.  List of the fields in TM\_LFR\_SCIENCE\*  PA\_BIA\_MODE\_MUX\_SET  PA\_BIA\_MODE\_HV\_ENABLED  PA\_\_BIA\_MODE\_BIAS1\_ENABLED  PA\_\_BIA\_MODE\_BIAS2\_ENABLED  PA\_\_BIA\_MODE\_BIAS3\_ENABLED  PA\_BIA\_ON\_OFF  In this step, only PA\_BIA\_ON\_OFF is updated = 1 | | | | |  |  |
|  | Send a TC\_LFR\_UPDATE\_INFO  With CP\_BIA\_MODE\_MUX\_SET = 0,1,2,3,4,5,5,7  Wait the delay to see all TM\_LFR\_SCIENCE | | | | Verify the LFR FSW extracts the relevant parameters from the shared status/HK packets for inserting them into their scientific TM packets.  List of the fields in TM\_LFR\_SCIENCE\*  PA\_BIA\_ON\_OFF = 1  And PA\_BIA\_MODE\_MUX\_SET = [0,7] | | | | |  |  |
|  | Send a TC\_LFR\_UPDATE\_INFO  With CP\_BIA\_MODE\_HV\_ENABLED = 1  Wait the delay to see all TM\_LFR\_SCIENCE | | | | Verify the LFR FSW extracts the relevant parameters from the shared status/HK packets for inserting them into their scientific TM packets.  List of the fields in TM\_LFR\_SCIENCE\*  PA\_BIA\_ON\_OFF = 1  PA\_BIA\_MODE\_MUX\_SET = 7  PA\_BIA\_MODE\_HV\_ENABLED = 1 | | | | |  |  |
|  | Send a TC\_LFR\_UPDATE\_INFO  With CP\_BIA\_MODE\_BIAS1\_ENABLED = 1  Wait the delay to see all TM\_LFR\_SCIENCE | | | | Verify the LFR FSW extracts the relevant parameters from the shared status/HK packets for inserting them into their scientific TM packets.  List of the fields in TM\_LFR\_SCIENCE\*  PA\_BIA\_ON\_OFF = 1  PA\_BIA\_MODE\_MUX\_SET = 7  PA\_BIA\_MODE\_HV\_ENABLED = 1  PA\_ BIA\_MODE\_BIAS1\_ENABLED =1 | | | | |  |  |
|  | Send a TC\_LFR\_UPDATE\_INFO  With CP\_BIA\_MODE\_BIAS2\_ENABLED = 1  Wait the delay to see all TM\_LFR\_SCIENCE | | | | Verify the LFR FSW extracts the relevant parameters from the shared status/HK packets for inserting them into their scientific TM packets.  List of the fields in TM\_LFR\_SCIENCE\*  PA\_BIA\_ON\_OFF = 1  PA\_BIA\_MODE\_MUX\_SET = 7  PA\_BIA\_MODE\_HV\_ENABLED = 1  PA\_ BIA\_MODE\_BIAS1\_ENABLED =1  PA\_ BIA\_MODE\_BIAS2\_ENABLED =1 | | | | |  |  |
|  | Send a TC\_LFR\_UPDATE\_INFO  With CP\_BIA\_MODE\_BIAS3\_ENABLED = 1  Wait the delay to see all TM\_LFR\_SCIENCE | | | | Verify the LFR FSW extracts the relevant parameters from the shared status/HK packets for inserting them into their scientific TM packets.  List of the fields in TM\_LFR\_SCIENCE\*  PA\_BIA\_ON\_OFF = 1  PA\_BIA\_MODE\_MUX\_SET = 7  PA\_BIA\_MODE\_HV\_ENABLED = 1  PA\_ BIA\_MODE\_BIAS1\_ENABLED =1  PA\_ BIA\_MODE\_BIAS2\_ENABLED =1  PA\_ BIA\_MODE\_BIAS3\_ENABLED =1 | | | | |  |  |
|  | End | | | |  | | | | |  |  |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE** | | | | | | | | | | |
| **Test procedure identifier** | | | | | **Date** | | | **Assessment** | | |
| SVS-0040\_Ed2 | | | | | DD/MM/YYYY | | | NT / NA / NOK / POK / OK | | |
| **TS Tested requirements** | | | | | **RB Tested requirements** | | | | | |
| N/A | | | | | REQ-LFR-SRS-5515\_Ed1 REQ-LFR-SRS-5573\_Ed1  REQ-LFR-SRS-5516\_Ed1 | | | | | |
| **Component:** | | | |  | **Version:** | V1 V2 | | | | |
| **Involved subsystems** | | | | | | | | | | |
| SW: | | 3.2.0.24 | | | | | | | | |
| HW: | | 1.1.91 StarDundee | | | | | | | | |
| Start time: | | | (DD/MM/YYYY) hh:mm | | Test Operator: | |  | | | |
| End Time: | | | (DD/MM/YYYY) hh:mm | | Test Duration: | | 35mn | | | |
| **Purpose:** | | | Check the LFR FSW transmits the science data as packets compliant to the PUS service n°21.  Check the LFR FSW always sets the segmentation grouping flag of the scientific TM packets to the value “Stand-alone packet”.(Value = 3) | | | | | | | |
| **General remarks about the test** | | | | | | | | | | |
| Requirement Title | | | | Science data acquisition and processing | | | | | | |
| dependencies | | | | SSS-GEN-FS-010, SSS-CP-EQS-380 SSS-CP-EQS-415 SSS-CP-EQS-390 | | | | | | |
| restrictions | | | |  | | | | | | |
| means | | | |  | | | | | | |
| input data | | | |  | | | | | | |
| prerequisite | | | |  | | | | | | |
| used test programs | | | |  | | | | | | |
| test script | | | | and service21\_consistency\_asm\_4s.py | | | | | | |
| output data | | | |  | | | | | | |
| notes | | | | Used verif\_fields.py for compliance.  We only played service21\_consistency\_asm\_4s.py | | | | | | |
| **Procedure steps** | | | | | | | | | | |
| **Step** | **Actions** | | | | **Expected Results** | | | | **Comments** | **Status** |
|  | In standby mode,  Send a TC\_LFR\_LOAD\_NORMAL\_PAR to set SY\_LFR\_ASM\_P=4s  Reach the NORMAL mode.  LFR FSW transmits each science data: TM\_LFR\_SCIENCE\_NORMAL\_SWF\_F0, TM\_LFR\_SCIENCE\_NORMAL\_SWF\_F1, TM\_LFR\_SCIENCE\_NORMAL\_SWF\_F2, TM\_LFR\_SCIENCE\_NORMAL\_CWF\_F3, TM\_LFR\_SCIENCE\_NORMAL\_ASM\_F0, TM\_LFR\_SCIENCE\_NORMAL\_ASM\_F1, TM\_LFR\_SCIENCE\_NORMAL\_ASM\_F2, TM\_LFR\_SCIENCE\_NORMAL\_BP1\_F0, TM\_LFR\_SCIENCE\_NORMAL\_BP1\_F1, TM\_LFR\_SCIENCE\_NORMAL\_BP1\_F2, TM\_LFR\_SCIENCE\_NORMAL\_BP2\_F0, TM\_LFR\_SCIENCE\_NORMAL\_BP2\_F1, TM\_LFR\_SCIENCE\_NORMAL\_BP2\_F2 | | | | Verify packets are compliant to the PUS service n°21 (PUS =”PUS\_VERSION = 1”,  SERVICE\_TYPE = “SCIENCE\_DATA\_TRANSFER = 21”, SERVICE\_SUBTYPE = “SCIENCE\_REPORT = 3”.  Used default values of parameters except ASM=4s | | | |  |  |
|  | Reach the BURST mode.  LFR FSW transmits each science data: TM\_LFR\_SCIENCE\_BURST\_CWF\_F2, TM\_LFR\_SCIENCE\_BURST\_BP1\_F0, TM\_LFR\_SCIENCE\_BURST\_BP2\_F0, TM\_LFR\_SCIENCE\_BURST\_BP1\_F1, TM\_LFR\_SCIENCE\_BURST\_BP2\_F1. | | | | Verify packets are compliant to the PUS service n°21 (PUS\_=”PUS\_VERSION = 1”,  SERVICE\_TYPE = “SCIENCE\_DATA\_TRANSFER = 21”, SERVICE\_SUBTYPE = “SCIENCE\_REPORT = 3”. | | | |  |  |
|  | Reach the SBM1 mode.  LFR FSW transmits each science data: TM\_LFR\_SCIENCE\_SBM1\_CWF\_F1, TM\_LFR\_SCIENCE\_SBM1\_BP1\_F0, TM\_LFR\_SCIENCE\_SBM1\_BP2\_F0, TM\_LFR\_SCIENCE\_NORMAL\_SWF\_F0, TM\_LFR\_SCIENCE\_NORMAL\_SWF\_F1, TM\_LFR\_SCIENCE\_NORMAL\_SWF\_F2, TM\_LFR\_SCIENCE\_NORMAL\_CWF\_F3, TM\_LFR\_SCIENCE\_NORMAL\_ASM\_F0, TM\_LFR\_SCIENCE\_NORMAL\_ASM\_F1, TM\_LFR\_SCIENCE\_NORMAL\_ASM\_F2, TM\_LFR\_SCIENCE\_NORMAL\_BP1\_F0, TM\_LFR\_SCIENCE\_NORMAL\_BP1\_F1, TM\_LFR\_SCIENCE\_NORMAL\_BP1\_F2, TM\_LFR\_SCIENCE\_NORMAL\_BP2\_F0, TM\_LFR\_SCIENCE\_NORMAL\_BP2\_F1, TM\_LFR\_SCIENCE\_NORMAL\_BP2\_F2. | | | | Verify packets are compliant to the PUS service n°21 (PUS\_=”PUS\_VERSION = 1”,  SERVICE\_TYPE = “SCIENCE\_DATA\_TRANSFER = 21”, SERVICE\_SUBTYPE = “SCIENCE\_REPORT = 3”. | | | |  |  |
|  | Reach the SBM2 mode.  LFR FSW transmits each science data: TM\_LFR\_SCIENCE\_SBM2\_CWF\_F2, TM\_LFR\_SCIENCE\_SBM2\_BP1\_F0, TM\_LFR\_SCIENCE\_SBM2\_BP2\_F0, TM\_LFR\_SCIENCE\_SBM2\_BP1\_F1, TM\_LFR\_SCIENCE\_SBM2\_BP2\_F1, TM\_LFR\_SCIENCE\_NORMAL\_SWF\_F0, TM\_LFR\_SCIENCE\_NORMAL\_SWF\_F1, TM\_LFR\_SCIENCE\_NORMAL\_SWF\_F2, TM\_LFR\_SCIENCE\_NORMAL\_CWF\_F3, TM\_LFR\_SCIENCE\_NORMAL\_ASM\_F0, TM\_LFR\_SCIENCE\_NORMAL\_ASM\_F1, TM\_LFR\_SCIENCE\_NORMAL\_ASM\_F2, TM\_LFR\_SCIENCE\_NORMAL\_BP1\_F0, TM\_LFR\_SCIENCE\_NORMAL\_BP1\_F1, TM\_LFR\_SCIENCE\_NORMAL\_BP1\_F2, TM\_LFR\_SCIENCE\_NORMAL\_BP2\_F0, TM\_LFR\_SCIENCE\_NORMAL\_BP2\_F1, TM\_LFR\_SCIENCE\_NORMAL\_BP2\_F2. | | | | Verify packets are compliant to the PUS service n°21 (PUS\_=”PUS\_VERSION = 1”,  SERVICE\_TYPE = “SCIENCE\_DATA\_TRANSFER = 21”, SERVICE\_SUBTYPE = “SCIENCE\_REPORT = 3”. | | | |  |  |
|  | Configure the TM\_LFR\_SCIENCE\_NORMAL\_CWF\_LONG\_F3 emission.  Reach the NORMAL mode.  LFR FSW transmits each science data: TM\_LFR\_SCIENCE\_NORMAL\_SWF\_F0, TM\_LFR\_SCIENCE\_NORMAL\_SWF\_F1, TM\_LFR\_SCIENCE\_NORMAL\_SWF\_F2, TM\_LFR\_SCIENCE\_NORMAL\_CWF\_LONG\_F3, TM\_LFR\_SCIENCE\_NORMAL\_ASM\_F0, TM\_LFR\_SCIENCE\_NORMAL\_ASM\_F1, TM\_LFR\_SCIENCE\_NORMAL\_ASM\_F2, TM\_LFR\_SCIENCE\_NORMAL\_BP1\_F0, TM\_LFR\_SCIENCE\_NORMAL\_BP1\_F1, TM\_LFR\_SCIENCE\_NORMAL\_BP1\_F2, TM\_LFR\_SCIENCE\_NORMAL\_BP2\_F0, TM\_LFR\_SCIENCE\_NORMAL\_BP2\_F1, TM\_LFR\_SCIENCE\_NORMAL\_BP2\_F2 | | | | Verify packets are compliant to the PUS service n°21 (PUS =”PUS\_VERSION = 1”,  SERVICE\_TYPE = “SCIENCE\_DATA\_TRANSFER = 21”, SERVICE\_SUBTYPE = “SCIENCE\_REPORT = 3”. | | | |  |  |
|  | Reach the SBM1 mode.  LFR FSW transmits each science data: TM\_LFR\_SCIENCE\_SBM1\_CWF\_F1, TM\_LFR\_SCIENCE\_SBM1\_BP1\_F0, TM\_LFR\_SCIENCE\_SBM1\_BP2\_F0, TM\_LFR\_SCIENCE\_NORMAL\_SWF\_F0, TM\_LFR\_SCIENCE\_NORMAL\_SWF\_F1, TM\_LFR\_SCIENCE\_NORMAL\_SWF\_F2, TM\_LFR\_SCIENCE\_NORMAL\_CWF\_LONG\_F3, TM\_LFR\_SCIENCE\_NORMAL\_ASM\_F0, TM\_LFR\_SCIENCE\_NORMAL\_ASM\_F1, TM\_LFR\_SCIENCE\_NORMAL\_ASM\_F2, TM\_LFR\_SCIENCE\_NORMAL\_BP1\_F0, TM\_LFR\_SCIENCE\_NORMAL\_BP1\_F1, TM\_LFR\_SCIENCE\_NORMAL\_BP1\_F2, TM\_LFR\_SCIENCE\_NORMAL\_BP2\_F0, TM\_LFR\_SCIENCE\_NORMAL\_BP2\_F1, TM\_LFR\_SCIENCE\_NORMAL\_BP2\_F2. | | | | Verify packets are compliant to the PUS service n°21 (PUS\_=”PUS\_VERSION = 1”,  SERVICE\_TYPE = “SCIENCE\_DATA\_TRANSFER = 21”, SERVICE\_SUBTYPE = “SCIENCE\_REPORT = 3”. | | | |  |  |
|  | Reach the SBM2 mode.  LFR FSW transmits each science data: TM\_LFR\_SCIENCE\_SBM2\_CWF\_F2, TM\_LFR\_SCIENCE\_SBM2\_BP1\_F0, TM\_LFR\_SCIENCE\_SBM2\_BP2\_F0, TM\_LFR\_SCIENCE\_SBM2\_BP1\_F1, TM\_LFR\_SCIENCE\_SBM2\_BP2\_F1, TM\_LFR\_SCIENCE\_NORMAL\_SWF\_F0, TM\_LFR\_SCIENCE\_NORMAL\_SWF\_F1, TM\_LFR\_SCIENCE\_NORMAL\_SWF\_F2, TM\_LFR\_SCIENCE\_NORMAL\_CWF\_LONG\_F3, TM\_LFR\_SCIENCE\_NORMAL\_ASM\_F0, TM\_LFR\_SCIENCE\_NORMAL\_ASM\_F1, TM\_LFR\_SCIENCE\_NORMAL\_ASM\_F2, TM\_LFR\_SCIENCE\_NORMAL\_BP1\_F0, TM\_LFR\_SCIENCE\_NORMAL\_BP1\_F1, TM\_LFR\_SCIENCE\_NORMAL\_BP1\_F2, TM\_LFR\_SCIENCE\_NORMAL\_BP2\_F0, TM\_LFR\_SCIENCE\_NORMAL\_BP2\_F1, TM\_LFR\_SCIENCE\_NORMAL\_BP2\_F2. | | | | Verify packets are compliant to the PUS service n°21 (PUS\_=”PUS\_VERSION = 1”,  SERVICE\_TYPE = “SCIENCE\_DATA\_TRANSFER = 21”, SERVICE\_SUBTYPE = “SCIENCE\_REPORT = 3”. | | | |  |  |
|  | End | | | |  | | | |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE** | | | | | | | | | | | |
| **Test procedure identifier** | | | | | | **Date** | | | **Assessment** | | |
| SVS-0041\_Ed2 | | | | | | DD/MM/YYYY | | | NT / NA / NOK / POK / OK | | |
| **TS Tested requirements** | | | | | | **RB Tested requirements** | | | | | |
| N/A | | | | | | REQ-LFR-SRS-5519\_Ed1 REQ-LFR-SRS-5524\_Ed2  REQ-LFR-SRS-5513\_Ed1 REQ-LFR-SRS-5569\_Ed1 | | | | | |
| **Component:** | | | |  | | **Version:** | V1 | | | | |
| **Involved subsystems** | | | | | | | | | | | |
| SW: | | 3.2.0.24 | | | | | | | | | |
| HW: | | 1.1.91 Stardundee | | | | | | | | | |
| Start time: | | | (DD/MM/YYYY) hh:mm | | | Test Operator: | |  | | | |
| End Time: | | | (DD/MM/YYYY) hh:mm | | | Test Duration: | | 20mn + 23h | | | |
| **Purpose:** | | | Check the LFR FSW is able to generate continuous waveform in each science mode.  Check the LFR FSW transmits to the DPU the continuous waveforms in data packet. | | | | | | | | |
| **General remarks about the test** | | | | | | | | | | | |
| Requirement Title | | | LFR data products - continuous waveforms and  LFR data packets - continuous waveforms | | | | | | | | |
| dependencies | | | SSS-CP-EQS-420 – SSS-CP-EQS-470 SSS-CP-EQS-360 | | | | | | | | |
| restrictions | | | Case STANDBY is not tested here. | | | | | | | | |
| means | | |  | | | | | | | | |
| input data | | |  | | | | | | | | |
| prerequisite | | |  | | | | | | | | |
| used test programs | | |  | | | | | | | | |
| test script | | | cwf\_generation.py + CTC-800  long tests + periodicity.py | | | | | | | | |
| output data | | |  | | | | | | | | |
| notes | | |  | | | | | | | | |
| **Procedure steps** | | | | | | | | | | | |
| **Step** | **Actions** | | | | **Expected Results** | | | | | **Comments** | **Status** |
|  | Launch LFR FSW  LFR is in **STANDBY** mode . | | | | Verify LFR doesn’t generate continuous waveforms. | | | | |  |  |
|  | LFR is in **NORMAL** mode without TC\_LFR\_LOAD\_NORMAL\_PAR Packet.; it generates continuous waveforms. | | | | In TM\_LFR\_SCIENCE\_NORMAL\_CWF\_F3 packet, verify continuous waveforms contains the following components:  -Electric field components sampled at f3: V\_f3, E1\_f3, E2\_f3.  After 168s (at least) in the current mode  We receive a train of 4 CWF\_F3 packets with a delta time of 42s between each one. | | | | |  |  |
|  | LFR is in **BURST** mode; it generates continuous waveforms. | | | | In TM\_LFR\_SCIENCE\_BURST\_CWF\_F2 packet, verify continuous waveforms contains the following components:  -Electric field components sampled at f2: V\_f2, E1\_f2, E2\_f2.  -Magnetic field components sampled at f2: B1\_f2, B2\_f2, B3\_f2.  After 11s in the current mode  We receive a train of 8 CWF\_F2 packets with a delta time of 1.315s between each one. | | | | |  |  |
|  | LFR is in **SBM1** mode; it generates continuous waveforms. | | | | In TM\_LFR\_SCIENCE\_SBM1\_CWF\_F1 packet, verify continuous waveforms contains the following components:  -Electric field components sampled at f1: V\_f1, E1\_f1, E2\_f1.  -Magnetic field components sampled at f1: B1\_f1, B2\_f1, B3\_f1.  After 1.2s in the current mode  We receive a train of 8 CWF\_F1 packets with a delta time of 0.0820 between each one.  In TM\_LFR\_SCIENCE\_NORMAL\_CWF\_F3 packet, verify continuous waveforms contains the following components:  -Electric field components sampled at f3: V\_f3, E1\_f3, E2\_f3.  After 168s (at least) in the current mode  We receive a train of 4 CWF\_F3 packets with a delta time of 42s between each one. | | | | |  |  |
|  | LFR is in **SBM2** mode; it generates continuous waveforms. | | | | In TM\_LFR\_SCIENCE\_SBM2\_CWF\_F2 Packet, verify continuous waveforms contains the following components:  -Electric field components sampled at f2: V\_f2, E1\_f2, E2\_f2.  -Magnetic field components sampled at f2: B1\_f2, B2\_f2, B3\_f2.  After 11s in the current mode  We receive a train of 8 CWF\_F2 packets with a delta time of 1.315s between each one.  In TM\_LFR\_SCIENCE\_NORMAL\_CWF\_F3 packet, verify continuous waveforms contains the following components:  -Electric field components sampled at f3: V\_f3, E1\_f3, E2\_f3.  We receive a train of 4 CWF\_F3 packets with a delta time of 42s between each one.  The CWF\_F3 dataflow is not be reset.  We observe a delta time of 42s between the last CWF\_F3 received in SBM1 and the first one of the train received in SBM2 mode. | | | | |  |  |
|  | LFR is in **NORMAL** mode after TC\_LFR\_LOAD\_NORMAL\_PAR Packet.;  **SY\_LFR\_N\_CWF\_LONG\_F3 = 1**  it generates continuous waveforms. | | | | In TM\_LFR\_SCIENCE\_NORMAL\_CWF\_LONG\_F3 packet, verify continuous waveforms contains the following components:  -Electric field components sampled at f3: V\_f3, E1\_f3, E2\_f3.  -Magnetic field components sampled at f3: B1\_f3, B2\_f3, B3\_f3.  After 168s (at least) in the current mode.  We receive a train of 8 CWF\_LONG\_F3 packets with a delta time of 21s between each one. | | | | |  |  |
|  | LFR is in **SBM1** mode; it generates continuous waveforms.  **SY\_LFR\_N\_CWF\_LONG\_F3 = 1** | | | | In TM\_LFR\_SCIENCE\_SBM1\_CWF\_F1 packet, verify continuous waveforms contains the following components:  -Electric field components sampled at f1: V\_f1, E1\_f1, E2\_f1.  -Magnetic field components sampled at f1: B1\_f1, B2\_f1, B3\_f1.  After 1.2s in the current mode  We receive a train of 8 CWF\_F1 packets with a delta time of 0.0820 between each one.  In TM\_LFR\_SCIENCE\_NORMAL\_CWF\_LONG\_F3 packet, verify continuous waveforms contains the following components:  -Electric field components sampled at f3: V\_f3, E1\_f3, E2\_f3.  -Magnetic field components sampled at f3: B1\_f3, B2\_f3, B3\_f3.  The CWF\_F3 dataflow is not be reset.  We observe a delta time of 21s between the last CWF\_LONG\_F3 received in NORMAL mode and the first one of the train received in SBM1 mode. | | | | |  |  |
|  | LFR is in **SBM2** mode; it generates continuous waveforms.  **SY\_LFR\_N\_CWF\_LONG\_F3 = 1** | | | | In TM\_LFR\_SCIENCE\_SBM2\_CWF\_F2 Packet, verify continuous waveforms contains the following components:  -Electric field components sampled at f2: V\_f2, E1\_f2, E2\_f2.  -Magnetic field components sampled at f2: B1\_f2, B2\_f2, B3\_f2.  After 11s in the current mode  We receive a train of 8 CWF\_F2 packets with a delta time of 1.315s between each one.  In TM\_LFR\_SCIENCE\_NORMAL\_CWFLONG\_\_F3 packet, verify continuous waveforms contains the following components:  -Electric field components sampled at f3: V\_f3, E1\_f3, E2\_f3.  -Magnetic field components sampled at f3: B1\_f3, B2\_f3, B3\_f3.  The CWF\_F3 dataflow is not be reset.  We observe a delta time of 21s between the last CWF\_LONG\_F3 received in SBM1 mode and the first one of the train received in SBM2 mode. | | | | |  |  |
|  | A long test is done during 4h30 in NORMAL MODE with defaults values.  We also use the timegen software (send CTR in nominal behavior). | | | | In TM\_LFR\_SCIENCE\_NORMAL\_CWF\_F3 packet, verify continuous waveforms contains the following components:  -Electric field components sampled at f3: V\_f3, E1\_f3, E2\_f3. | | | | |  |  |
|  | A long test is done during 15 hours in NORMAL MODE with  SY\_LFR\_N\_CWF\_LONG\_F3 = 0  SY\_LFR\_N\_SWF\_P = 22s  SY\_LFR\_N\_ASM\_P = 4s  We also use the timegen software (send CTR in nominal behavior). | | | | In TM\_LFR\_SCIENCE\_NORMAL\_CWF\_LONG\_F3 packet, verify continuous waveforms contains the following components:  -Electric field components sampled at f3: V\_f3, E1\_f3, E2\_f3.  -Magnetic field components sampled at f3: B1\_f3, B2\_f3, B3\_f3. | | | | |  |  |
|  | A long test is done during 14 hours in BURST MODE with defaults values.  We also use the timegen software (send CTR in nominal behavior). | | | | In TM\_LFR\_SCIENCE\_BURST\_CWF\_F2 packet, verify continuous waveforms contains the following components:  -Electric field components sampled at f2: V\_f2, E1\_f2, E2\_f2.  -Magnetic field components sampled at f2: B1\_f2, B2\_f2, B3\_f2. | | | | |  |  |
|  | A long test is done during 2h30 in SBM1 MODE with defaults values.  in NORMAL MODE with defaults values  We also use the timegen software (send CTR in nominal behavior). | | | | In TM\_LFR\_SCIENCE\_SBM1\_CWF\_F1 packet, verify continuous waveforms contains the following components:  -Electric field components sampled at f1: V\_f1, E1\_f1, E2\_f1.  -Magnetic field components sampled at f1: B1\_f1, B2\_f1, B3\_f1.  In TM\_LFR\_SCIENCE\_NORMAL\_CWF\_F3 packet, verify continuous waveforms contains the following components:  -Electric field components sampled at f3: V\_f3, E1\_f3, E2\_f3. | | | | |  |  |
|  | A long test is done during 3 hours in SBM1 MODE with defaults values.  in NORMAL MODE with defaults values except to SY\_LFR\_N\_CWF\_LONG\_F3 = 1  We also use the timegen software (send CTR in nominal behavior). | | | | In TM\_LFR\_SCIENCE\_SBM1\_CWF\_F1 packet, verify continuous waveforms contains the following components:  -Electric field components sampled at f1: V\_f1, E1\_f1, E2\_f1.  -Magnetic field components sampled at f1: B1\_f1, B2\_f1, B3\_f1.  In TM\_LFR\_SCIENCE\_NORMAL\_CWF\_LONG\_F3 packet, verify continuous waveforms contains the following components:  -Electric field components sampled at f3: V\_f3, E1\_f3, E2\_f3.  -Magnetic field components sampled at f3: B1\_f3, B2\_f3, B3\_f3. | | | | |  |  |
|  | A long test is done during 1 hour in SBM2 MODE with defaults values.  in NORMAL MODE with defaults values  We also use the timegen software (send CTR in nominal behavior). | | | | In TM\_LFR\_SCIENCE\_SBM2\_CWF\_F2 Packet, verify continuous waveforms contains the following components:  -Electric field components sampled at f2: V\_f2, E1\_f2, E2\_f2.  -Magnetic field components sampled at f2: B1\_f2, B2\_f2, B3\_f2.  In TM\_LFR\_SCIENCE\_NORMAL\_CWF\_LONG\_F3 packet, verify continuous waveforms contains the following components:  -Electric field components sampled at f3: V\_f3, E1\_f3, E2\_f3.  -Magnetic field components sampled at f3: B1\_f3, B2\_f3, B3\_f3. | | | | |  |  |
|  | A long test is done during 3 hours in SBM2 MODE with defaults values.  in NORMAL MODE with defaults values except to SY\_LFR\_N\_CWF\_LONG\_F3 = 1  We also use the timegen software (send CTR in nominal behavior). | | | | In TM\_LFR\_SCIENCE\_SBM2\_CWF\_F2 Packet, verify continuous waveforms contains the following components:  -Electric field components sampled at f2: V\_f2, E1\_f2, E2\_f2.  -Magnetic field components sampled at f2: B1\_f2, B2\_f2, B3\_f2.  In TM\_LFR\_SCIENCE\_NORMAL\_CWF\_F3 packet, verify continuous waveforms contains the following components:  -Electric field components sampled at f3: V\_f3, E1\_f3, E2\_f3. | | | | |  |  |
|  | End | | | |  | | | | |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE** | | | | | | | | | | | |
| **Test procedure identifier** | | | | | | **Date** | | | **Assessment** | | |
| SVS-0042\_Ed2 | | | | | | DD/MM/YYYY | | | NT / NA / NOK / POK / OK | | |
| **TS Tested requirements** | | | | | | **RB Tested requirements** | | | | | |
| N/A | | | | | | REQ-LFR-SRS-5520\_Ed1 REQ-LFR-SRS-5525\_Ed1  REQ-LFR-SRS-5513\_Ed1 | | | | | |
| **Component:** | | | |  | | **Version:** | V1 | | | | |
| **Involved subsystems** | | | | | | | | | | | |
| SW: | | 3.2.0.24 | | | | | | | | | |
| HW: | | 1.1.91 Stardundee | | | | | | | | | |
| Start time: | | | (DD/MM/YYYY) hh:mm | | | Test Operator: | |  | | | |
| End Time: | | | (DD/MM/YYYY) hh:mm | | | Test Duration: | | 42mn + 23h | | | |
| **Purpose:** | | | Check the LFR FSW is able to generate waveform snapshots.  Check the LFR FSW transmits to the DPU the waveform snapshot in data packets. | | | | | | | | |
| **General remarks about the test** | | | | | | | | | | | |
| Requirement Title | | | | LFR data products - waveform snapshots and  LFR data packets - waveform snapshots | | | | | | | |
| dependencies | | | | SSS-CP-EQS-430 SSS-CP-EQS-480 SSS-CP-EQS-360 | | | | | | | |
| restrictions | | | |  | | | | | | | |
| means | | | |  | | | | | | | |
| input data | | | |  | | | | | | | |
| prerequisite | | | |  | | | | | | | |
| used test programs | | | |  | | | | | | | |
| test script | | | | swf\_generation.py + verifs\_fields.py  long tests + periodicity.py | | | | | | | |
| output data | | | |  | | | | | | | |
| notes | | | |  | | | | | | | |
| **Procedure steps** | | | | | | | | | | | |
| **Step** | **Actions** | | | | **Expected Results** | | | | | **Comments** | **Status** |
|  | LFR is in **STANDBY** mode without TC\_LFR\_LOAD\_NORMAL\_PAR Packet.  Wait ~330s. | | | | The LFR FSW doesn’t generate waveform snapshots. | | | | |  |  |
|  | LFR realizes **NORMAL** mode transition (without TC\_LFR\_LOAD\_NORMAL\_PAR Packet).  The LFR FSW generates waveform snapshots. | | | | With TM\_LFR\_SCIENCE\_NORMAL\_SWF\_F0, verify the waveform snapshot generated contains the following components:  -2048 samples every 300 seconds.  -Electric field components sampled at f0: V\_f0, E1\_f0, E2\_f0.  -Magnetic field components sampled at f0: B1\_f0, B2\_f0, B3\_f0.  With TM\_LFR\_SCIENCE\_NORMAL\_SWF\_F1, verify the waveform snapshot generated contains the following components:  -2048 samples every 300 seconds.  -Electric field components sampled at f1: V\_f1, E1\_f1, E2\_f1.  -Magnetic field components sampled at f1: B1\_f1, B2\_f1, B3\_f1.  With TM\_LFR\_SCIENCE\_NORMAL\_SWF\_F2, verify the waveform snapshot generated contains the following components:  -2048 samples every 300 seconds.  -Electric field components sampled at f2: V\_f2, E1\_f2, E2\_f2.  -Magnetic field components sampled at f2: B1\_f2, B2\_f2, B3\_f2.  After 300s (at least ) in the current mode  We receive a train of 7 NORMAL\_SWF\_Fi packets i=[0,2]  F0: delta time between is 810/811 fine time = 0.03149s  F1: delta time between is 1300 fine time = 0.0742s  F2: delta time between is 3000 fine time = 0.1875s | | | | |  |  |
|  | LFR realizes **BURST** mode transition without TC\_LFR\_LOAD\_NORMAL\_PAR Packet.  Wait ~330s. | | | | The LFR FSW doesn’t generate waveform snapshots. | | | | |  |  |
|  | LFR realizes **SBM1** mode transition (without TC\_LFR\_LOAD\_NORMAL\_PAR Packet).  The LFR FSW generates waveform snapshots. | | | | With TM\_LFR\_SCIENCE\_NORMAL\_SWF\_F0, verify the waveform snapshot generated contains the following components:  -2048 samples every 300 seconds.  -Electric field components sampled at f0: V\_f0, E1\_f0, E2\_f0.  -Magnetic field components sampled at f0: B1\_f0, B2\_f0, B3\_f0.  With TM\_LFR\_SCIENCE\_NORMAL\_SWF\_F1, verify the waveform snapshot generated contains the following components:  -2048 samples every 296 seconds.  -Electric field components sampled at f1: V\_f1, E1\_f1, E2\_f1.  -Magnetic field components sampled at f1: B1\_f1, B2\_f1, B3\_f1.  With TM\_LFR\_SCIENCE\_NORMAL\_SWF\_F2, verify the waveform snapshot generated contains the following components:  -2048 samples every 296 seconds.  -Electric field components sampled at f2: V\_f2, E1\_f2, E2\_f2.  -Magnetic field components sampled at f2: B1\_f2, B2\_f2, B3\_f2.  The SWF\_Fi dataflow is not be reset. i=[0,2]  We observe a delta time of right time between the last SWF\_Fi received in SBM1 mode and the first one of the train received in NORMAL mode.  F0: delta time between is 810/811 fine time = 0.03149s  F1: delta time between is 1300 fine time = 0.0742s  F2: delta time between is 3000 fine time = 0.1875s | | | | |  |  |
|  | LFR realizes **SBM2** mode transition (without TC\_LFR\_LOAD\_NORMAL\_PAR Packet).  The LFR FSW generates waveform snapshots. | | | | With TM\_LFR\_SCIENCE\_NORMAL\_SWF\_F0, verify the waveform snapshot generated contains the following components:  -2048 samples every 300 seconds.  -Electric field components sampled at f0: V\_f0, E1\_f0, E2\_f0.  -Magnetic field components sampled at f0: B1\_f0, B2\_f0, B3\_f0.  With TM\_LFR\_SCIENCE\_NORMAL\_SWF\_F1, verify the waveform snapshot generated contains the following components:  -2048 samples every 296 seconds.  -Electric field components sampled at f1: V\_f1, E1\_f1, E2\_f1.  -Magnetic field components sampled at f1: B1\_f1, B2\_f1, B3\_f1.  With TM\_LFR\_SCIENCE\_NORMAL\_SWF\_F2, verify the waveform snapshot generated contains the following components:  -2048 samples every 296 seconds.  -Electric field components sampled at f2: V\_f2, E1\_f2, E2\_f2.  -Magnetic field components sampled at f2: B1\_f2, B2\_f2, B3\_f2.  The SWF\_Fi dataflow is not be reset. I=[0,2]  We observe a delta time of right time between the last SWF\_Fi received in SBM2 mode and the first one of the train received in SBM1 mode.  F0: delta time between is 810/811 fine time = 0.03149s  F1: delta time between is 1300 fine time = 0.0742s  F2: delta time between is 3000 fine time = 0.1875s | | | | |  |  |
|  | A long test is done during 4h30 in NORMAL MODE with defaults values.  We also use the timegen software (send CTR in nominal behavior). | | | | With TM\_LFR\_SCIENCE\_NORMAL\_SWF\_F0, verify the waveform snapshot generated contains the following components:  -2048 samples every 300 seconds.  -Electric field components sampled at f0: V\_f0, E1\_f0, E2\_f0.  -Magnetic field components sampled at f0: B1\_f0, B2\_f0, B3\_f0.  With TM\_LFR\_SCIENCE\_NORMAL\_SWF\_F1, verify the waveform snapshot generated contains the following components:  -2048 samples every 300 seconds.  -Electric field components sampled at f1: V\_f1, E1\_f1, E2\_f1.  -Magnetic field components sampled at f1: B1\_f1, B2\_f1, B3\_f1.  With TM\_LFR\_SCIENCE\_NORMAL\_SWF\_F2, verify the waveform snapshot generated contains the following components:  -2048 samples every 300 seconds.  -Electric field components sampled at f2: V\_f2, E1\_f2, E2\_f2.  -Magnetic field components sampled at f2: B1\_f2, B2\_f2, B3\_f2.  At 300s (at least) in the current mode  We receive a train of 7 NORMAL\_SWF\_Fi packets i=[0,2]  F0: delta time between is 810/811 fine time = 0.03149s  F1: delta time between is 1300 fine time = 0.0742s  F2: delta time between is 3000 fine time = 0.1875s | | | | |  |  |
|  | A long test is done during 15 hours in NORMAL MODE with  SY\_LFR\_N\_CWF\_LONG\_F3 = 0  SY\_LFR\_N\_SWF\_P = 22s  SY\_LFR\_N\_ASM\_P = 4s  We also use the timegen software (send CTR in nominal behavior). | | | | With TM\_LFR\_SCIENCE\_NORMAL\_SWF\_F0, verify the waveform snapshot generated contains the following components:  -2048 samples every 22 seconds.  -Electric field components sampled at f0: V\_f0, E1\_f0, E2\_f0.  -Magnetic field components sampled at f0: B1\_f0, B2\_f0, B3\_f0.  With TM\_LFR\_SCIENCE\_NORMAL\_SWF\_F1, verify the waveform snapshot generated contains the following components:  -2048 samples every 22 seconds.  -Electric field components sampled at f1: V\_f1, E1\_f1, E2\_f1.  -Magnetic field components sampled at f1: B1\_f1, B2\_f1, B3\_f1.  With TM\_LFR\_SCIENCE\_NORMAL\_SWF\_F2, verify the waveform snapshot generated contains the following components:  -2048 samples every 22 seconds.  -Electric field components sampled at f2: V\_f2, E1\_f2, E2\_f2.  -Magnetic field components sampled at f2: B1\_f2, B2\_f2, B3\_f2.  At 38s (at least) in the current mode  We receive a train of 7 NORMAL\_SWF\_Fi packets i=[0,2]  F0: delta time between is 810/811 fine time = 0.03149s  F1: delta time between is 1300 fine time = 0.0742s  F2: delta time between is 3000 fine time = 0.1875s  After the delta time between each first packet of a train is 22s. | | | | |  |  |
|  | A long test is done during 14 hours in BURST MODE with defaults values.  We also use the timegen software (send CTR in nominal behavior). | | | | The LFR FSW doesn’t generate waveform snapshots. | | | | |  |  |
|  | A long test is done during 2h30 in SBM1 MODE with defaults values.  in NORMAL MODE with defaults values  We also use the timegen software (send CTR in nominal behavior). | | | | With TM\_LFR\_SCIENCE\_NORMAL\_SWF\_F0, verify the waveform snapshot generated contains the following components:  -2048 samples every 300 seconds.  -Electric field components sampled at f0: V\_f0, E1\_f0, E2\_f0.  -Magnetic field components sampled at f0: B1\_f0, B2\_f0, B3\_f0.  With TM\_LFR\_SCIENCE\_NORMAL\_SWF\_F1, verify the waveform snapshot generated contains the following components:  -2048 samples every 300 seconds.  -Electric field components sampled at f1: V\_f1, E1\_f1, E2\_f1.  -Magnetic field components sampled at f1: B1\_f1, B2\_f1, B3\_f1.  With TM\_LFR\_SCIENCE\_NORMAL\_SWF\_F2, verify the waveform snapshot generated contains the following components:  -2048 samples every 300 seconds.  -Electric field components sampled at f2: V\_f2, E1\_f2, E2\_f2.  -Magnetic field components sampled at f2: B1\_f2, B2\_f2, B3\_f2.  At 300s (at least) in the current mode  We receive a train of 7 NORMAL\_SWF\_Fi packets i=[0,2]  F0: delta time between is 810/811 fine time = 0.03149s  F1: delta time between is 1300 fine time = 0.0742s  F2: delta time between is 3000 fine time = 0.1875s | | | | |  |  |
|  | A long test is done during 1 hour in SBM2 MODE with defaults values.  in NORMAL MODE with defaults values  We also use the timegen software (send CTR in nominal behavior). | | | | With TM\_LFR\_SCIENCE\_NORMAL\_SWF\_F0, verify the waveform snapshot generated contains the following components:  -2048 samples every 300 seconds.  -Electric field components sampled at f0: V\_f0, E1\_f0, E2\_f0.  -Magnetic field components sampled at f0: B1\_f0, B2\_f0, B3\_f0.  With TM\_LFR\_SCIENCE\_NORMAL\_SWF\_F1, verify the waveform snapshot generated contains the following components:  -2048 samples every 300 seconds.  -Electric field components sampled at f1: V\_f1, E1\_f1, E2\_f1.  -Magnetic field components sampled at f1: B1\_f1, B2\_f1, B3\_f1.  With TM\_LFR\_SCIENCE\_NORMAL\_SWF\_F2, verify the waveform snapshot generated contains the following components:  -2048 samples every 300 seconds.  -Electric field components sampled at f2: V\_f2, E1\_f2, E2\_f2.  -Magnetic field components sampled at f2: B1\_f2, B2\_f2, B3\_f2.  At 300s (at least) in the current mode  We receive a train of 7 NORMAL\_SWF\_Fi packets i=[0,2]  F0: delta time between is 810/811 fine time = 0.03149s  F1: delta time between is 1300 fine time = 0.0742s  F2: delta time between is 3000 fine time = 0.1875s | | | | |  |  |
|  | End | | | |  | | | | |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE** | | | | | | | | | | | |
| **Test procedure identifier** | | | | | | **Date** | | | **Assessment** | | |
| SVS-0043\_Ed1 | | | | | | DD/MM/YYYY | | | NT / NA / NOK / POK / OK | | |
| **TS Tested requirements** | | | | | | **RB Tested requirements** | | | | | |
| N/A | | | | | | REQ-LFR-SRS-5521\_Ed1 REQ-LFR-SRS-5526\_Ed1  REQ-LFR-SRS-5513\_Ed1 | | | | | |
| **Component:** | | | |  | | **Version:** | V2 | | | | |
| **Involved subsystems** | | | | | | | | | | | |
| SW: | | 3.2.0.24 | | | | | | | | | |
| HW: | | 1.1.91 StarDundee | | | | | | | | | |
| Start time: | | | (DD/MM/YYYY) hh:mm | | | Test Operator: | |  | | | |
| End Time: | | | (DD/MM/YYYY) hh:mm | | | Test Duration: | | 9mn + 23h | | | |
| **Purpose:** | | | Check the LFR FSW is able to generate averaged spectral matrixes.  Check the LFR FSW transmits to the DPU the averaged spectral matrixes in data packets. | | | | | | | | |
| **General remarks about the test** | | | | | | | | | | | |
| Requirement Title | | | | LFR data products - averaged spectral matrixes and  LFR data packets - averaged spectral matrixes | | | | | | | |
| dependencies | | | | SSS-CP-EQS-440 SSS-CP-EQS-490 SSS-CP-EQS-360 | | | | | | | |
| restrictions | | | |  | | | | | | | |
| means | | | |  | | | | | | | |
| input data | | | |  | | | | | | | |
| prerequisite | | | |  | | | | | | | |
| used test programs | | | |  | | | | | | | |
| test script | | | | Asm\_generation\_4s.py  + verif\_fields.py with tm\_lfr\_period rule  Asm\_generation.py + asm\_generation\_long\_period\_Normal/SBM1/SBM2.py (See long tests)  + periodicity.py | | | | | | | |
| output data | | | |  | | | | | | | |
| notes | | | |  | | | | | | | |
| **Procedure steps** | | | | | | | | | | | |
| **Step** | **Actions** | | | | **Expected Results** | | | | | **Comments** | **Status** |
|  | **Use asm\_generation\_asm\_4s.py**  Send a TC\_LFR\_LOAD\_NORMAL\_PAR with SY\_LFR\_ASM\_P=4s  LFR is in **NORMAL** mode.  The LFR FSW generates averaged spectral matrixes.  Wait 100s | | | | Verify averaged spectral matrixes generated:  - every 4s seconds:  -Averaged spectral matrix from the EM data stream at f0: ASM\_f0  -Averaged spectral matrix from the EM data stream at f1: ASM\_f1  -Averaged spectral matrix from the EM data stream at f2: ASM\_f2. | | | | |  |  |
|  | LFR is in **SBM1** mode.  The LFR FSW generates averaged spectral matrixes. | | | | Verify averaged spectral matrixes generated:  - every 4s seconds:  -Averaged spectral matrix from the EM data stream at f0: ASM\_f0  -Averaged spectral matrix from the EM data stream at f1: ASM\_f1  -Averaged spectral matrix from the EM data stream at f2: ASM\_f2.  the NORMAL\_ASM\_Fi dataflow is reset.i=[0,2] | | | | |  |  |
|  | LFR is in **SBM2** mode.  The LFR FSW generates averaged spectral matrixes. | | | | Verify averaged spectral matrixes generated:  - every 4s seconds:  -Averaged spectral matrix from the EM data stream at f0: ASM\_f0  -Averaged spectral matrix from the EM data stream at f1: ASM\_f1  -Averaged spectral matrix from the EM data stream at f2: ASM\_f2.  the NORMAL\_ASM\_Fi dataflow is reset.i=[0,2] | | | | |  |  |
|  | A long test is done during 4h30 in NORMAL MODE with defaults values.  We also use the timegen software (send CTR in nominal behavior). | | | | Verify averaged spectral matrixes generated:  - every 3600s seconds:  -Averaged spectral matrix from the EM data stream at f0: ASM\_f0  -Averaged spectral matrix from the EM data stream at f1: ASM\_f1  -Averaged spectral matrix from the EM data stream at f2: ASM\_f2. | | | | |  |  |
|  | A long test is done during 15 hours in NORMAL MODE SY\_LFR\_N\_CWF\_LONG\_F3 =0  SY\_LFR\_N\_SWF\_P = 22s  SY\_LFR\_N\_ASM\_P = 4s  We also use the timegen software (send CTR in nominal behavior). | | | | Verify averaged spectral matrixes generated:  - every 3600s seconds:  -Averaged spectral matrix from the EM data stream at f0: ASM\_f0  -Averaged spectral matrix from the EM data stream at f1: ASM\_f1  -Averaged spectral matrix from the EM data stream at f2: ASM\_f2. | | | | |  |  |
|  | A long test is done during 2h30 in SBM1 MODE with defaults values.  in NORMAL MODE with defaults values  We also use the timegen software (send CTR in nominal behavior). | | | | Verify averaged spectral matrixes generated:  - every 3600s seconds:  -Averaged spectral matrix from the EM data stream at f0: ASM\_f0  -Averaged spectral matrix from the EM data stream at f1: ASM\_f1  -Averaged spectral matrix from the EM data stream at f2: ASM\_f2. | | | | |  |  |
|  | A long test is done during 3 hours in SBM1 MODE with defaults values.  in NORMAL MODE with defaults values except to  SY\_LFR\_N\_CWF\_LONG\_F3 = 1  We also use the timegen software (send CTR in nominal behavior). | | | | Verify averaged spectral matrixes generated:  - every 3600s seconds:  -Averaged spectral matrix from the EM data stream at f0: ASM\_f0  -Averaged spectral matrix from the EM data stream at f1: ASM\_f1  -Averaged spectral matrix from the EM data stream at f2: ASM\_f2. | | | | |  |  |
|  | A long test is done during 1 hour in SBM2 MODE with defaults values.  in NORMAL MODE with defaults values.  We also use the timegen software (send CTR in nominal behavior). | | | | Verify averaged spectral matrixes generated:  - every 3600s seconds:  -Averaged spectral matrix from the EM data stream at f0: ASM\_f0  -Averaged spectral matrix from the EM data stream at f1: ASM\_f1  -Averaged spectral matrix from the EM data stream at f2: ASM\_f2. | | | | |  |  |
|  | A long test is done during 3 hours in SBM2 MODE with defaults values.  in NORMAL MODE with defaults values, except to  SY\_LFR\_N\_CWF\_LONG\_F3 = 1  We also use the timegen software (send CTR in nominal behavior). | | | | Verify averaged spectral matrixes generated:  - every 3600s seconds:  -Averaged spectral matrix from the EM data stream at f0: ASM\_f0  -Averaged spectral matrix from the EM data stream at f1: ASM\_f1  -Averaged spectral matrix from the EM data stream at f2: ASM\_f2. | | | | |  |  |
|  | End | | | |  | | | | |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE** | | | | | | | | | | | |
| **Test procedure identifier** | | | | | | **Date** | | | **Assessment** | | |
| SVS-0044\_Ed1 | | | | | | DD/MM/YYYY | | | NT / NA / NOK / POK / OK | | |
| **TS Tested requirements** | | | | | | **RB Tested requirements** | | | | | |
| N/A | | | | | | REQ-LFR-SRS-5522\_Ed2  REQ-LFR-SRS-5527\_Ed1 | | | | | |
| **Component:** | | | |  | | **Version:** | V2 | | | | |
| **Involved subsystems** | | | | | | | | | | | |
| SW: | | 3.2.0.24 | | | | | | | | | |
| HW: | | 1.1.91 StarDundee | | | | | | | | | |
| Start time: | | | (DD/MM/YYYY) hh:mm | | | Test Operator: | |  | | | |
| End Time: | | | (DD/MM/YYYY) hh:mm | | | Test Duration: | | 50mn + 23h | | | |
| **Purpose:** | | | Check the LFR FSW is able to generate the set of basic parameters “set 1”  Check the LFR FSW transmits to the DPU the set of basic parameters 1 in data packets. | | | | | | | | |
| **General remarks about the test** | | | | | | | | | | | |
| Requirement Title | | | LFR data products - basic parameters set 1 and  LFR data packets - basic parameters set 1 | | | | | | | | |
| dependencies | | | SSS-CP-EQS-450 SSS-CP-EQS-500 | | | | | | | | |
| restrictions | | |  | | | | | | | | |
| means | | |  | | | | | | | | |
| input data | | |  | | | | | | | | |
| prerequisite | | |  | | | | | | | | |
| used test programs | | | LFR\_packet\_decom-linux-x86\_64 (C++ LPP decom program) | | | | | | | | |
| test script | | | bp\_generation\_all\_modes.py + verif\_fields.py with tm\_lfr\_period rule  + long tests + periodicity | | | | | | | | |
| output data | | |  | | | | | | | | |
| notes | | | PE\_fn is the spectral power of E field from the electric data stream (2E) at fn; 12 or 24 bins  PB\_fn is the spectral power of B field from the magnetic data stream (3B) at fn; 12 or 24 bins  nvec\_fn is the wave normal vector from the magnetic data stream (3B) at fn; 12 or 24 bins  ellip\_fn is the wave ellipticity from the magnetic data stream (3B) at fn; 12 or 24 bins  dop\_fn is the degree of polarization from the magnetic data stream (3B) at fn; 12 or 24 bins  Sz\_fn is the normalized z-Poynting flux from the EM data stream (2E +3B) at fn; 12 or 24 bins  Vphi\_fn is the phase speed from the EM data stream (2E +3B) at fn; 12 or 24 bins  Wait 650s in each mode | | | | | | | | |
| **Procedure steps** | | | | | | | | | | | |
| **Step** | **Actions** | | | | **Expected Results** | | | | | **Comments** | **Status** |
|  | LFR is in **NORMAL** mode.  The LFR FSW generates set of basic parameters “set 1”. | | | | Verify the set of basic parameters generated:  **Every 4 seconds**:  -Basic parameter 1 at f0: PE\_f0, PB\_f0, nvec\_f0, ellip\_f0, dop\_f0, Sz\_f0, Vphi\_f0  -Basic parameter 1 at f1: PE\_f1, PB\_f1, nvec\_f1, ellip\_f1, dop\_f1, Sz\_f1, Vphi\_f1  -Basic parameter 1 at f2: PE\_f2, PB\_f2, nvec\_f2, ellip\_f2, dop\_f2, Sz\_f2, Vphi\_f2 | | | | |  |  |
|  | LFR is in **BURST** mode.  The LFR FSW generates set of basic parameters “set 1”. | | | | Verify the set of basic parameters generated:  **Every 1 second**:  -Basic parameter 1 at f0: PE\_f0, PB\_f0, nvec\_f0, ellip\_f0, dop\_f0, Sz\_f0, Vphi\_f0  -Basic parameter 1 at f1: PE\_f1, PB\_f1, nvec\_f1, ellip\_f1, dop\_f1, Sz\_f1, Vphi\_f1 | | | | |  |  |
|  | LFR is in **SBM1** mode.  The LFR FSW generates set of basic parameters “set 1”. | | | | Verify the set of basic parameters generated:  **Every 4 seconds**:  -Basic parameter 1 at f0: PE\_f0, PB\_f0, nvec\_f0, ellip\_f0, dop\_f0, Sz\_f0, Vphi\_f0  -Basic parameter 1 at f1: PE\_f1, PB\_f1, nvec\_f1, ellip\_f1, dop\_f1, Sz\_f1, Vphi\_f1  -Basic parameter 1 at f2: PE\_f2, PB\_f2, nvec\_f2, ellip\_f2, dop\_f2, Sz\_f2, Vphi\_f2  **Every 0.25 seconds**:  -Basic parameter 1 at f0: PE\_f0, PB\_f0, nvec\_f0, ellip\_f0, dop\_f0, Sz\_f0, Vphi\_f0. | | | | |  |  |
|  | LFR is in **SBM2** mode.  The LFR FSW generates set of basic parameters “set 1”. | | | | Verify the set of basic parameters generated:  **Every 4 seconds**:  -Basic parameter 1 at f0: PE\_f0, PB\_f0, nvec\_f0, ellip\_f0, dop\_f0, Sz\_f0, Vphi\_f0.  -Basic parameter 1 at f1: PE\_f1, PB\_f1, nvec\_f1, ellip\_f1, dop\_f1, Sz\_f1, Vphi\_f1.  -Basic parameter 1 at f2: PE\_f2, PB\_f2, nvec\_f2, ellip\_f2, dop\_f2, Sz\_f2, Vphi\_f2.  **Every** **1 second**:  -Basic parameter 1 at f0: PE\_f0, PB\_f0, nvec\_f0, ellip\_f0, dop\_f0, Sz\_f0, Vphi\_f0.  -Basic parameter 1 at f1: PE\_f1, PB\_f1, nvec\_f1, ellip\_f1, dop\_f1, Sz\_f1, Vphi\_f1. | | | | |  |  |
|  | A long test is done during 4h30 in NORMAL MODE with defaults values.  We also use the timegen software (send CTR in nominal behavior). | | | | Verify the set of basic parameters generated:  **Every 4 seconds**:  -Basic parameter 1 at f0: PE\_f0, PB\_f0, nvec\_f0, ellip\_f0, dop\_f0, Sz\_f0, Vphi\_f0.  -Basic parameter 1 at f1: PE\_f1, PB\_f1, nvec\_f1, ellip\_f1, dop\_f1, Sz\_f1, Vphi\_f1.  -Basic parameter 1 at f2: PE\_f2, PB\_f2, nvec\_f2, ellip\_f2, dop\_f2, Sz\_f2, Vphi\_f2. | | | | |  |  |
|  | A long test is done during 15 hours  in NORMAL MODE with  SY\_LFR\_N\_CWF\_LONG\_F3 = 0  SY\_LFR\_N\_SWF\_P = 22s  SY\_LFR\_N\_ASM\_P = 4s  We also use the timegen software (send CTR in nominal behavior). | | | | Verify the set of basic parameters generated:  **Every 4 seconds**:  -Basic parameter 1 at f0: PE\_f0, PB\_f0, nvec\_f0, ellip\_f0, dop\_f0, Sz\_f0, Vphi\_f0.  -Basic parameter 1 at f1: PE\_f1, PB\_f1, nvec\_f1, ellip\_f1, dop\_f1, Sz\_f1, Vphi\_f1.  -Basic parameter 1 at f2: PE\_f2, PB\_f2, nvec\_f2, ellip\_f2, dop\_f2, Sz\_f2, Vphi\_f2. | | | | |  |  |
|  | A long test is done during 14 hours in BURST MODE with defaults values.  We also use the timegen software (send CTR in nominal behavior). | | | | Verify the set of basic parameters generated:  **Every 1 second**:  -Basic parameter 1 at f0: PE\_f0, PB\_f0, nvec\_f0, ellip\_f0, dop\_f0, Sz\_f0, Vphi\_f0  -Basic parameter 1 at f1: PE\_f1, PB\_f1, nvec\_f1, ellip\_f1, dop\_f1, Sz\_f1, Vphi\_f1 | | | | |  |  |
|  | A long test is done during 2h30 in SBM1 MODE with defaults values.  in NORMAL MODE with defaults values.  We also use the timegen software (send CTR in nominal behavior). | | | | Verify the set of basic parameters generated:  **Every 4 seconds**:  -Basic parameter 1 at f0: PE\_f0, PB\_f0, nvec\_f0, ellip\_f0, dop\_f0, Sz\_f0, Vphi\_f0  -Basic parameter 1 at f1: PE\_f1, PB\_f1, nvec\_f1, ellip\_f1, dop\_f1, Sz\_f1, Vphi\_f1  -Basic parameter 1 at f2: PE\_f2, PB\_f2, nvec\_f2, ellip\_f2, dop\_f2, Sz\_f2, Vphi\_f2  **Every 0.25 seconds**:  -Basic parameter 1 at f0: PE\_f0, PB\_f0, nvec\_f0, ellip\_f0, dop\_f0, Sz\_f0, Vphi\_f0 | | | | |  |  |
|  | A long test is done during 3 hours in SBM1 MODE with defaults values.  in NORMAL MODE with defaults values except to  SY\_LFR\_N\_CWF\_LONG\_F3 = 1  We also use the timegen software (send CTR in nominal behavior). | | | | Verify the set of basic parameters generated:  **Every 4 seconds**:  -Basic parameter 1 at f0: PE\_f0, PB\_f0, nvec\_f0, ellip\_f0, dop\_f0, Sz\_f0, Vphi\_f0  -Basic parameter 1 at f1: PE\_f1, PB\_f1, nvec\_f1, ellip\_f1, dop\_f1, Sz\_f1, Vphi\_f1  -Basic parameter 1 at f2: PE\_f2, PB\_f2, nvec\_f2, ellip\_f2, dop\_f2, Sz\_f2, Vphi\_f2  **Every 0.25 seconds**:  -Basic parameter 1 at f0: PE\_f0, PB\_f0, nvec\_f0, ellip\_f0, dop\_f0, Sz\_f0, Vphi\_f0 | | | | |  |  |
|  | A long test is done during 1 hour in SBM2 MODE with defaults values.  in NORMAL MODE with defaults values.  We also use the timegen software (send CTR in nominal behavior). | | | | Verify the set of basic parameters generated:  **Every 4 seconds**:  -Basic parameter 1 at f0: PE\_f0, PB\_f0, nvec\_f0, ellip\_f0, dop\_f0, Sz\_f0, Vphi\_f0.  -Basic parameter 1 at f1: PE\_f1, PB\_f1, nvec\_f1, ellip\_f1, dop\_f1, Sz\_f1, Vphi\_f1.  -Basic parameter 1 at f2: PE\_f2, PB\_f2, nvec\_f2, ellip\_f2, dop\_f2, Sz\_f2, Vphi\_f2.  **Every** **1 second**:  -Basic parameter 1 at f0: PE\_f0, PB\_f0, nvec\_f0, ellip\_f0, dop\_f0, Sz\_f0, Vphi\_f0.  -Basic parameter 1 at f1: PE\_f1, PB\_f1, nvec\_f1, ellip\_f1, dop\_f1, Sz\_f1, Vphi\_f1. | | | | |  |  |
|  | A long test is done during 3 hours in SBM2 MODE with defaults values.  in NORMAL MODE with defaults values except to  SY\_LFR\_N\_CWF\_LONG\_F3 = 1  We also use the timegen software (send CTR in nominal behavior). | | | | Verify the set of basic parameters generated:  **Every 4 seconds**:  -Basic parameter 1 at f0: PE\_f0, PB\_f0, nvec\_f0, ellip\_f0, dop\_f0, Sz\_f0, Vphi\_f0.  -Basic parameter 1 at f1: PE\_f1, PB\_f1, nvec\_f1, ellip\_f1, dop\_f1, Sz\_f1, Vphi\_f1.  -Basic parameter 1 at f2: PE\_f2, PB\_f2, nvec\_f2, ellip\_f2, dop\_f2, Sz\_f2, Vphi\_f2.  **Every** **1 second**:  -Basic parameter 1 at f0: PE\_f0, PB\_f0, nvec\_f0, ellip\_f0, dop\_f0, Sz\_f0, Vphi\_f0.  -Basic parameter 1 at f1: PE\_f1, PB\_f1, nvec\_f1, ellip\_f1, dop\_f1, Sz\_f1, Vphi\_f1. | | | | |  |  |
|  | End | | | |  | | | | |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE** | | | | | | | | | | | |
| **Test procedure identifier** | | | | | | **Date** | | | **Assessment** | | |
| SVS-0045\_Ed1 | | | | | | DD/MM/YYYY | | | NT / NA / NOK / POK / OK | | |
| **TS Tested requirements** | | | | | | **RB Tested requirements** | | | | | |
| N/A | | | | | | REQ-LFR-SRS-5523\_Ed2  REQ-LFR-SRS-5528\_Ed1 | | | | | |
| **Component:** | | | |  | | **Version:** | V2 | | | | |
| **Involved subsystems** | | | | | | | | | | | |
| SW: | | 3.2.0.24 | | | | | | | | | |
| HW: | | 1.1.91 StarDundee | | | | | | | | | |
| Start time: | | | (DD/MM/YYYY) hh:mm | | | Test Operator: | |  | | | |
| End Time: | | | (DD/MM/YYYY) hh:mm | | | Test Duration: | | 50mn + 23h | | | |
| **Purpose:** | | | Check the LFR FSW is able to generate the set of basic parameters “set 2”  Check LFR FSW transmits to the DPU the set of basic parameters 2 in data packets | | | | | | | | |
| **General remarks about the test** | | | | | | | | | | | |
| Requirement Title | | | LFR data products - basic parameters set 2 and  LFR data packets - basic parameters set 2 | | | | | | | | |
| dependencies | | | SSS-CP-EQS-510 | | | | | | | | |
| restrictions | | |  | | | | | | | | |
| means | | |  | | | | | | | | |
| input data | | |  | | | | | | | | |
| prerequisite | | |  | | | | | | | | |
| used test programs | | | LFR\_packet\_decom-linux-x86\_64 (C++ LPP decom program) | | | | | | | | |
| test script | | | bp\_generation\_all\_modes. + verif\_fields.py with tm\_lfr\_period rule  long tests + periodicity.py | | | | | | | | |
| output data | | |  | | | | | | | | |
| notes | | | auto\_fn corresponds to 5 auto variances from the EM data stream (2E +3B) at fn ; 12 or 24 bins  cross\_fn corresponds to 10 complex cross correlations from the EM data stream (2E +3B) at fn ; 12 or 24 bins | | | | | | | | |
| **Procedure steps** | | | | | | | | | | | |
| **Step** | **Actions** | | | | **Expected Results** | | | | | **Comments** | **Status** |
|  | LFR is in **NORMAL** mode.  The LFR FSW generates set of basic parameters “set 2”. | | | | Verify the set of basic parameters generated:  **Every 20 seconds**:  -Basic parameter 2 at f0: auto\_f0, cross\_f0  -Basic parameter 2 at f1: auto\_f1, cross\_f1  -Basic parameter 2 at f2: auto\_f2, cross\_f2 | | | | |  |  |
|  | LFR is in **BURST** mode.  The LFR FSW generates set of basic parameters “set 2”. | | | | Verify the set of basic parameters generated:  **Every 5 seconds**:  -Basic parameter 2 at f0: auto\_f0, cross\_f0  -Basic parameter 2 at f1: auto\_f1, cross\_f1 | | | | |  |  |
|  | LFR is in **SBM1** mode.  The LFR FSW generates set of basic parameters “set 2”. | | | | Verify the set of basic parameters generated:  **Every 20 seconds**:  -Basic parameter 2 at f0: auto\_f0, cross\_f0  -Basic parameter 2 at f1: auto\_f1, cross\_f1  -Basic parameter 2 at f2: auto\_f2, cross\_f2  **Every 1 seconds**:  -Basic parameter 2 at f0: auto\_f0, cross\_f0 | | | | |  |  |
|  | LFR is in **SBM2** mode.  The LFR FSW generates set of basic parameters “set 2”. | | | | Verify the set of basic parameters generated:  **Every 20 seconds**:  -Basic parameter 2 at f0: auto\_f0, cross\_f0  -Basic parameter 2 at f1: auto\_f1, cross\_f1  -Basic parameter 2 at f2: auto\_f2, cross\_f2  **Every 5 seconds**:  -Basic parameter 2 at f0: auto\_f0, cross\_f0  -Basic parameter 2 at f1: auto\_f1, cross\_f1 | | | | |  |  |
|  | A long test is done during 4h30 in NORMAL MODE with defaults values.  We also use the timegen software (send CTR in nominal behavior). | | | | Verify the set of basic parameters generated:  **Every 20 seconds**:  -Basic parameter 2 at f0: auto\_f0, cross\_f0  -Basic parameter 2 at f1: auto\_f1, cross\_f1  -Basic parameter 2 at f2: auto\_f2, cross\_f2 | | | | |  |  |
|  | A long test is done during 15 hours in NORMAL MODE with  SY\_LFR\_N\_CWF\_LONG\_F3 = 0  SY\_LFR\_N\_SWF\_P = 22s  SY\_LFR\_N\_ASM\_P = 4s  We also use the timegen software (send CTR in nominal behavior). | | | | Verify the set of basic parameters generated:  **Every 20 seconds**:  -Basic parameter 2 at f0: auto\_f0, cross\_f0  -Basic parameter 2 at f1: auto\_f1, cross\_f1  -Basic parameter 2 at f2: auto\_f2, cross\_f2 | | | | |  |  |
|  | A long test is done during 14 hours in BURST MODE with defaults values.  We also use the timegen software (send CTR in nominal behavior). | | | | Verify the set of basic parameters generated:  **Every 5 seconds**:  -Basic parameter 2 at f0: auto\_f0, cross\_f0  -Basic parameter 2 at f1: auto\_f1, cross\_f1 | | | | |  |  |
|  | A long test is done during 2h30 in SBM1 MODE with defaults values.  in NORMAL MODE with defaults values.  We also use the timegen software (send CTR in nominal behavior). | | | | Verify the set of basic parameters generated:  **Every 20 seconds**:  -Basic parameter 2 at f0: auto\_f0, cross\_f0  -Basic parameter 2 at f1: auto\_f1, cross\_f1  -Basic parameter 2 at f2: auto\_f2, cross\_f2  **Every 1 seconds**:  -Basic parameter 2 at f0: auto\_f0, cross\_f0 | | | | |  |  |
|  | A long test is done during 3 hours in SBM1 MODE with defaults values.  in NORMAL MODE with defaults values except to  SY\_LFR\_N\_CWF\_LONG\_F3 = 1  We also use the timegen software (send CTR in nominal behavior). | | | | Verify the set of basic parameters generated:  **Every 20 seconds**:  -Basic parameter 2 at f0: auto\_f0, cross\_f0  -Basic parameter 2 at f1: auto\_f1, cross\_f1  -Basic parameter 2 at f2: auto\_f2, cross\_f2  **Every 1 seconds**:  -Basic parameter 2 at f0: auto\_f0, cross\_f0 | | | | |  |  |
|  | A long test is done during 1 hours in SBM2 MODE with defaults values.  in NORMAL MODE with defaults values.  We also use the timegen software (send CTR in nominal behavior). | | | | Verify the set of basic parameters generated:  **Every 20 seconds**:  -Basic parameter 2 at f0: auto\_f0, cross\_f0  -Basic parameter 2 at f1: auto\_f1, cross\_f1  -Basic parameter 2 at f2: auto\_f2, cross\_f2  **Every 5 seconds**:  -Basic parameter 2 at f0: auto\_f0, cross\_f0  -Basic parameter 2 at f1: auto\_f1, cross\_f1 | | | | |  |  |
|  | A long test is done during 3 hours in SBM2 MODE with defaults values.  in NORMAL MODE with defaults values except to  SY\_LFR\_N\_CWF\_LONG\_F3 = 1  We also use the timegen software (send CTR in nominal behavior). | | | | Verify the set of basic parameters generated:  **Every 20 seconds**:  -Basic parameter 2 at f0: auto\_f0, cross\_f0  -Basic parameter 2 at f1: auto\_f1, cross\_f1  -Basic parameter 2 at f2: auto\_f2, cross\_f2  **Every 5 seconds**:  -Basic parameter 2 at f0: auto\_f0, cross\_f0  -Basic parameter 2 at f1: auto\_f1, cross\_f1 | | | | |  |  |
|  | End | | | |  | | | | |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE** | | | | | | | | | | | |
| **Test procedure identifier** | | | | | | **Date** | | | **Assessment** | | |
| SVS-0053\_Ed2 | | | | | | DD/MM/YYYY | | | NT / NA / NOK / POK / OK | | |
| **TS Tested requirements** | | | | | | **RB Tested requirements** | | | | | |
| N/A | | | | | | REQ-LFR-SRS-5558\_Ed1  REQ-LFR-SRS-5556\_Ed1  REQ-LFR-SRS-5557\_Ed1 | | | | | |
| **Component:** | | | |  | | **Version:** | V2 | | | | |
| **Involved subsystems** | | | | | | | | | | | |
| SW: | | 3.2.0.24 | | | | | | | | | |
| HW: | | 1.1.91 Stardundee | | | | | | | | | |
| Start time: | | | (DD/MM/YYYY) hh:mm | | | Test Operator: | |  | | | |
| End Time: | | | (DD/MM/YYYY) hh:mm | | | Test Duration: | | 23mn | | | |
| **Purpose:** | | | Check the LFR FSW reports in its periodic HK packet (TM\_LFR\_HK) the enable/disable status of the calibration function.  Check upon reception of a TC\_LFR\_ENABLE\_CALIBRATION, the LFR FSW enables the LFR calibration function.  Check upon reception of a TC\_LFR\_DISABLE\_CALIBRATION, the LFR FSW disables the LFR calibration function. | | | | | | | | |
| **General remarks about the test** | | | | | | | | | | | |
| Requirement Title | | | LFR calibration function | | | | | | | | |
| dependencies | | | SSS-CP-EQS-524 - SSS-CP-EQS-522 SSS-CP-EQS-523 | | | | | | | | |
| restrictions | | |  | | | | | | | | |
| means | | |  | | | | | | | | |
| input data | | |  | | | | | | | | |
| prerequisite | | |  | | | | | | | | |
| used test programs | | |  | | | | | | | | |
| test script | | | hkCalEnabled\_signal.py + Acceptance tests | | | | | | | | |
| output data | | | See plots into Calibration report. | | | | | | | | |
| notes | | |  | | | | | | | | |
| **Procedure steps** | | | | | | | | | | | |
| **Step** | **Actions** | | | | **Expected Results** | | | | | **Comments** | **Status** |
|  | Run the LFR FSW  wait 5s | | | | After the boot sequence the  Field HK\_LFR\_CALIB\_ENABLED = 0 (DISABLED) | | | | |  |  |
|  | Send a  TC\_LFR\_LOAD\_NORMAL\_PAR with SWF\_P = 22s  For each mode:  -LFR receipts a TC\_LFR\_ENABLE\_CALIBRATION.  .  .  and stays 250 sec  Wait 10s  LFR receipts a  TC\_LFR\_DISABLE\_CALIBRATION | | | | TC is acknowledged by a TM\_LFR\_TC\_EXE\_SUCCESS.  Verify the LFR FSW enables the LFR calibration function (generation of the calibration signal for the SCM).  Start the calibration signal  Verify the LFR FSW reports in its periodic HK packet (TM\_LFR\_HK) the enable status of the calibration function.  Field HK\_LFR\_CALIB\_ENABLED = 1 when TC\_LFR\_ENABLE\_CALIBRATION  TC is acknowledged by a TM\_LFR\_TC\_EXE\_SUCCESS.  Verify the LFR FSW disabled the LFR calibration function (generation of the calibration signal for the SCM).  Stop the calibration signal  Field HK\_LFR\_CALIB\_ENABLED = 0 when TC\_LFR\_DISABLE\_CALIBRATION | | | | | No mode constraint  Only Normal, sbm1,sbm2  generate SWF products |  |
|  | For each mode  Send TC\_LFR\_ENABLE\_CALIBRATION  Wait 3s  Send TC\_LFR\_ENABLE\_CALIBRATION  Wait 3s  Send TC\_LFR\_DISABLE\_CALIBRATION  Wait 3s  Send TC\_LFR\_DISABLE\_CALIBRATION | | | | Each TC is acknowledged by a TM\_LFR\_TC\_EXE\_SUCCESS.  In TM\_LFR\_HK  Field HK\_LFR\_CALIB\_ENABLED = 1 when TC\_LFR\_ENABLE\_CALIBRATION  Field HK\_LFR\_CALIB\_ENABLED = 0 when TC\_LFR\_DISABLE\_CALIBRATION | | | | |  |  |
|  | End | | | |  | | | | |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE** | | | | | | | | | | | | | |
| **Test procedure identifier** | | | | | | | **Date** | | | | **Assessment** | | |
| SVS-0054\_Ed2 | | | | | | | DD/MM/YYYY | | | | NT / NA / NOK / POK / OK | | |
| **TS Tested requirements** | | | | | | | **RB Tested requirements** | | | | | | |
| N/A | | | | | | | REQ-LFR-SRS-5530\_Ed1 | | | | | | |
| **Component:** | | | | |  | **Version:** | | V1 | | | | | |
| **Involved subsystems** | | | | | | | | | | | | | |
| SW: | | 3.2.0.24 | | | | | | | | | | | |
| HW: | | 1.1.91 StarDundee | | | | | | | | | | | |
| Start time: | | | (DD/MM/YYYY) hh:mm | | | Test Operator: | | | |  | | | |
| End Time: | | | (DD/MM/YYYY) hh:mm | | | Test Duration: | | | 4s | | | | |
| **Purpose:** | | | Check the LFR FSW is ready to accept commands within SY\_LFR\_DELAY\_ACC\_TC milliseconds after the boot process has been completed. (500ms) | | | | | | | | | | |
| **General remarks about the test** | | | | | | | | | | | | | |
| Requirement Title | | | | Startup phase | | | | | | | | | |
| dependencies | | | | SSS-CP-EQS-010 | | | | | | | | | |
| restrictions | | | |  | | | | | | | | | |
| means | | | |  | | | | | | | | | |
| input data | | | | AD11 | | | | | | | | | |
| prerequisite | | | |  | | | | | | | | | |
| used test programs | | | |  | | | | | | | | | |
| test script | | | | delay\_after\_boot\_process.py | | | | | | | | | |
| output data | | | |  | | | | | | | | | |
| notes | | | | Tc\_lfr\_reset performs exit(0) | | | | | | | | | |
| **Procedure steps** | | | | | | | | | | | | | |
| **Step** | **Actions** | | | | | | **Expected Results** | | | | | **Comments** | **Status** |
|  | Reset the LFR board  By the Python script :  -FSW is loaded, and not started.  Realize as fast as possible the actions below:  -Run the FSW,  -Launch a TC\_LFR\_ENTER\_MODE | | | | | | Taking into account the technical uncertainty verify:  -LFR FSW stays healthy during the boot process and startup phase.  -when the boot process has just been completed, the LFR FSW accepts commands after SY\_LFR\_DELAY\_ACC\_TC.  TC accepts with a TM\_LFR\_TC\_EXE\_SUCCESS  Use in TM\_LFR\_HK  HK\_LFR\_LAST\_EXE\_TC\_TYPE=181  HK\_LFR\_LAST\_EXE\_SUBTYPE=41  HK\_LFR\_LAST\_EXE\_TC\_TIME. | | | | |  |  |
|  | End | | | | | |  | | | | |  |  |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE** | | | | | | | | | | |
| **Test procedure identifier** | | | | **Date** | | | | **Assessment** | | |
| SVS-0055\_Ed2 | | | | DD/MM/YYYY | | | | NT / NA / NOK / POK / OK | | |
| **TS Tested requirements** | | | | **RB Tested requirements** | | | | | | |
| N/A | | | | REQ-LFR-SRS-5531\_Ed1 REQ-LFR-SRS-5535\_Ed1 | | | | | | |
| **Component:** | | |  | **Version:** | V2 | | | | | |
| **Involved subsystems** | | | | | | | | | | |
| SW: | 3.2.0.24 | | | | | | | | | |
| HW: | 1.1.91 StarDundee | | | | | | | | | |
| Start time: | | (DD/MM/YYYY) hh:mm | | Test Operator: | | |  | | | |
| End Time: | | (DD/MM/YYYY) hh:mm | | Test Duration: | | 5s | | | | |
| **Purpose:** | | Check after successful time synchronization just after Startup phase, the LFR FSW enables the generation of its periodic housekeeping reports. | | | | | | | | |
| **General remarks about the test** | | | | | | | | | | |
| Requirement Title | | | Startup phase | | | | | | | |
| dependencies | | | SSS-CP-EQS-020 SSS-CP-EQS-060 | | | | | | | |
| restrictions | | |  | | | | | | | |
| means | | |  | | | | | | | |
| input data | | |  | | | | | | | |
| prerequisite | | |  | | | | | | | |
| used test programs | | |  | | | | | | | |
| test script | | | synchro\_startup\_phase.py | | | | | | | |
| output data | | |  | | | | | | | |
| notes | | | The test case does not check the TC\_LFR\_RESET case.  Use excel one file.csv and check that HK is compliant to PUS service 3 with verif\_fields.py | | | | | | | |
| **Procedure steps** | | | | | | | | | | |
| **Step** | **Actions** | | | **Expected Results** | | | | | **Comments** | **Status** |
|  | -Launch the FSW  Send a TC\_LFR\_UPDATE\_TIME + time-code  - Wait the generation of its periodic housekeeping reports.. | | | Verify the LFR FSW enables the generation of its periodic housekeeping reports.  First TM\_LFR\_HK has its time=0x8xxxxxxxx  After synchronization , the TM\_LFR\_HK has its time=0x00000002xxxx  time is updated by CP-RPW\_TIME of TC\_LFR\_UPDATE\_TIME  and MSB = 1 for synchronization  HK\_LFR\_DPU\_SPW\_TICK\_OUT\_CNT=1, HK\_LFR\_DPU\_SPW\_LAST\_TIMC=1  HK\_LFR\_UPDATE\_TIME\_TC\_CNT=1 | | | | |  |  |
|  | See previous action. | | | Verify the LFR FSW sends its HK report as a packet compliant to the PUS service n°3 (PACKET\_CATEGORY=”HK\_ROUTINE = 4”, PUS\_VERSION=”PUS\_VERSION = 1”, SERVICE\_TYPE=”HOUSEKEEPING\_AND\_DIAGNOSTIC\_DATA\_REPORTING = 3)”. | | | | | REQ-LFR-SRS-5535 |  |
|  | End | | |  | | | | |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE** | | | | | | | | | | | | |
| **Test procedure identifier** | | | | | | **Date** | | | | **Assessment** | | |
| SVS-0056\_Ed2 | | | | | | DD/MM/YYYY | | | | NT / NA / NOK / POK / OK | | |
| **TS Tested requirements** | | | | | | **RB Tested requirements** | | | | | | |
| N/A | | | | | | REQ-LFR-SRS-5532\_Ed1 REQ-LFR-SRS-5535\_Ed1 | | | | | | |
| **Component:** | | | |  | | **Version:** | V2 | | | | | |
| **Involved subsystems** | | | | | | | | | | | | |
| SW: | 3.2.0.24 | | | | | | | | | | | |
| HW: | 1.1.91 StarDundee | | | | | | | | | | | |
| Start time: | | (DD/MM/YYYY) hh:mm | | | | Test Operator: | | |  | | | |
| End Time: | | (DD/MM/YYYY) hh:mm | | | | Test Duration: | | 37s | | | | |
| **Purpose:** | | After Startup phase, check if no time synchronization has occurred within SY\_LFR\_TIME\_SYN\_TIMEOUT, the LFR FSW initiates the generation of its housekeeping reports using a non-synchronized time value. | | | | | | | | | | |
| **General remarks about the test** | | | | | | | | | | | | |
| Requirement Title | | | Startup phase | | | | | | | | | |
| dependencies | | | SSS-CP-EQS-030 SSS-CP-EQS-060 | | | | | | | | | |
| restrictions | | |  | | | | | | | | | |
| means | | |  | | | | | | | | | |
| input data | | |  | | | | | | | | | |
| prerequisite | | |  | | | | | | | | | |
| used test programs | | |  | | | | | | | | | |
| test script | | | timeout\_synchro\_after\_boot.step01.py and timeout\_synchro\_after\_boot.step02.py and  and timeout\_synchro\_after\_boot.step03.py | | | | | | | | | |
| output data | | |  | | | | | | | | | |
| notes | | | SY\_LFR\_TIME\_SYN\_TIMEOUT (Analyzer time synchronization timeout) is in AD11, 2000ms  The TEST CASE does not treat the startup case after a TC\_LFR\_RESET. | | | | | | | | | |
| **Procedure steps** | | | | | | | | | | | | |
| **Step** | **Actions** | | | | **Expected Results** | | | | | | **Comments** | **Status** |
|  | FSW is loaded, and not started.  Launch the Python script.  Start the FSW (run):  neither TC\_LFR\_UPDATE\_TIMEs, neither time code. | | | | SY\_LFR\_TIME\_SYN\_TIMEOUT after startup, verify the LFR FSW initiates the generation of its housekeeping reports using a non-synchronized time value: in  TM\_LFR\_HK packets: MSB of TIME field is set to 1 .b1 (bit).  in TM\_LFR\_HK  HK\_LFR\_TIMECODE\_MISSING=1  because automatic timecode is not launched.  HK\_LFR\_LAST\_ER\_RID = LE\_LFR\_TIMEC (42129)  HK\_LFR\_LAST\_ER\_CODE = MISSING (21)  No timecodes:  HK\_LFR\_DPU\_SPW\_TICK\_OUT\_CNT=0, HK\_LFR\_DPU\_SPW\_LAST\_TIMC=0  HK\_LFR\_TIME\_NOT\_SYNCHRO= 0  this field is not updated because LFR must be synchronize first.    HK\_LFR\_LE\_CNT = 1  Verify the LFR FSW sends its HK report as a packet compliant to the PUS service n°3 (PACKET\_CATEGORY=”HK\_ROUTINE = 4”, PUS\_VERSION=”PUS\_VERSION = 1”, SERVICE\_TYPE=”HOUSEKEEPING\_AND\_DIAGNOSTIC\_DATA\_REPORTING = 3)”. | | | | | | Time of the first TM\_LFR\_HK=  0x80000002xxxx  REQ-LFR-SRS-5535\_Ed1 |  |
|  | FSW is loaded, and not started.  Launch the Python script.  Start the FSW (run): TC\_LFR\_UPDATE\_TIMEs without time code. | | | | SY\_LFR\_TIME\_SYN\_TIMEOUT after startup, verify in TM\_LFR\_HK packets: MSB of TIME field is set to 1 b1 (bit).  in TM\_LFR\_HK  HK\_LFR\_TIMECODE\_MISSING=1  HK\_LFR\_TIME\_TIMECODE\_IT = 20  HK\_LFR\_LAST\_ER\_RID = LE\_LFR\_TIMEC (42129)  HK\_LFR\_LAST\_ER\_CODE = MISSING (21)  HK\_LFR\_TIME\_NOT\_SYNCHRO=0 this field is not updated because LFR must be synchronize first.  No timecodes:  HK\_LFR\_DPU\_SPW\_TICK\_OUT\_CNT=0, HK\_LFR\_DPU\_SPW\_LAST\_TIMC=0  HK\_LFR\_UPDATE\_TIME\_CNT = 1  HK\_LFR\_LE\_CNT = 1  Verify the LFR FSW sends its HK report as a packet compliant to the PUS service n°3 (PACKET\_CATEGORY=”HK\_ROUTINE = 4”, PUS\_VERSION=”PUS\_VERSION = 1”, SERVICE\_TYPE=”HOUSEKEEPING\_AND\_DIAGNOSTIC\_DATA\_REPORTING = 3)”. | | | | | | Time of the first TM\_LFR\_HK=  0x80000002xxxx  REQ-LFR-SRS-5535\_Ed1 |  |
|  | Exit of socexplorer.  FSW is loaded, and not started.  Launch the Python script.  Start the FSW (run):20 time codes without TC\_LFR\_UPDATE\_TIME.  After 20 timecodes we stop it. | | | | SY\_LFR\_TIME\_SYN\_TIMEOUT after startup, verify in TM\_LFR\_HK packets: MSB of TIME field is set to 1 b1 (bit).  send 20 timecodes:  HK\_LFR\_DPU\_SPW\_TICK\_OUT\_CNT=20, HK\_LFR\_DPU\_SPW\_LAST\_TIMC=20  HK\_LFR\_TIME\_TIMECODE\_IT = 20  HK\_LFR\_LAST\_ER\_RID = LE\_LFR\_TIME (42119)  HK\_LFR\_LAST\_ER\_CODE = TIMECODE\_IT (24)  HK\_LFR\_UPDATE\_TIME\_CNT is not incremented = 0  HK\_LFR\_LE\_CNT =20  After automatic timecode stop  HK\_LFR\_TIME\_MISSING = 1  HK\_LFR\_LAST\_ER\_RID = LE\_LFR\_TIMEC (42129)  HK\_LFR\_LAST\_ER\_CODE = MISSING (21)  HK\_LFR\_LE\_CNT =21  Verify the LFR FSW sends its HK report as a packet compliant to the PUS service n°3 (PACKET\_CATEGORY=”HK\_ROUTINE = 4”, PUS\_VERSION=”PUS\_VERSION = 1”, SERVICE\_TYPE=”HOUSEKEEPING\_AND\_DIAGNOSTIC\_DATA\_REPORTING = 3)”. | | | | | | Time of the first TM\_LFR\_HK=  0x80000002xxxx  REQ-LFR-SRS-5535\_Ed1 |  |
|  | End | | | |  | | | | | |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE** | | | | | | | | | | | | | |
| **Test procedure identifier** | | | | | | | **Date** | | | | **Assessment** | | |
| SVS-0057\_Ed1 | | | | | | | DD/MM/YYYY | | | | NT / NA / NOK / POK / OK | | |
| **TS Tested requirements** | | | | | | | **RB Tested requirements** | | | | | | |
| N/A | | | | | | | REQ-LFR-SRS-5533\_Ed1 REQ-LFR-SRS-5534\_Ed1  REQ-LFR-SRS-5535\_Ed1 REQ-LFR-SRS-5230\_Ed1  REQ-LFR-SRS-5214\_Ed1 REQ-LFR-SRS-5215\_Ed1  REQ-LFR-SRS-5216\_Ed1  REQ-LFR-SRS-6002\_Ed1 REQ-LFR-SRS-6102\_Ed1  REQ-LFR-SRS-6105\_Ed1 | | | | | | |
| **Component:** | | | | |  | | **Version:** | V1 /V3++ | | | | | |
| **Involved subsystems** | | | | | | | | | | | | | |
| SW: | | 3.2.0.24 | | | | | | | | | | | |
| HW: | | 1.1.91 StarDundee | | | | | | | | | | | |
| Start time: | | | (DD/MM/YYYY) hh:mm | | | | Test Operator: | | |  | | | |
| End Time: | | | (DD/MM/YYYY) hh:mm | | | | Test Duration: | | 5mn30 | | | | |
| **Purpose:** | | | Check the LFR FSW provides periodically a TM\_LFR\_HK packet (HK report) reflecting the status of the LFR.  Check the LFR FSW provides in HK report SW and HW status  Check the LFR FSW sends its HK report as a packet compliant to the PUS service n°3. | | | | | | | | | | |
| **General remarks about the test** | | | | | | | | | | | | | |
| Requirement Title | | | | Equipment HK reporting | | | | | | | | | |
| dependencies | | | | SSS-CP-EQS-040 SSS-CP-EQS-050 SSS-CP-EQS-060 SSS-CP-FS-520 SSS-CP-FS-150  SSS-CP-FS-160 SSS-CP-170  SSS-CP-EQS-752 SSS-CP-EQS-763 SSS-CP-EQS-766 | | | | | | | | | |
| restrictions | | | |  | | | | | | | | | |
| means | | | |  | | | | | | | | | |
| input data | | | |  | | | | | | | | | |
| prerequisite | | | | REQ-LFR-SRS-5215\_Ed1 (“The mode of housekeeping packet generation shall be the periodic mode.”) | | | | | | | | | |
| used test programs | | | |  | | | | | | | | | |
| test script | | | | periodical\_hk.py hk\_reporting (used by verif\_fields) for analysis.  3 scripts for timecode management: starting\_time\_set\_time.py, (step7) ,  starting\_time\_timecode\_start.py (step 8)  starting\_time\_timecode\_start\_after.py (step 9) | | | | | | | | | |
| output data | | | |  | | | | | | | | | |
| notes | | | | all tests verified these requirements | | | | | | | | | |
| **Procedure steps** | | | | | | | | | | | | | |
| **Step** | **Actions** | | | | | **Expected Results** | | | | | | **Comments** | **Status** |
|  | Launch LFR FSW and wait end of boot sequence  no automatic timecode | | | | | Verify the LFR FSW provides periodically a TM\_LFR\_HK packet (HK report) reflecting the status of the LFR after the boot sequence:  TIME = 0x80000002xxxx    HK\_LFR\_MODE = STANDBY  HK\_LFR\_DPU\_SPW\_ENABLED: ENABLED = 1, HK\_LFR\_DPU\_SPW\_LINK\_STATE: RUN = 5,  HK\_LFR\_SC\_POTENTIEL\_FLAG: ON = 1  SY\_LFR\_PAS\_FILTER\_ENABLED: DISABLED =0,  SY\_LFR\_WATCHDOG\_ENABLED: ENABLED = 1, HK\_LFR\_CALIB\_ENABLED: DISABLED = 0, HK\_LFR\_RESET\_CAUSE: POWER\_ON = 1 HK\_LFR\_SC\_POTENTIEL\_FLAG: ON = 1  Default parameters values:  SY\_LFR\_BW=1, SY\_LFR\_SP0=0, SY\_LFR\_SP1=0, SY\_LFR\_R0=0, SY\_LFR\_R1=0, SY\_LFR\_R2=0  Specific parameters  HK\_LFR\_VHDL\_SM=0, HK\_LFR\_VHDL\_IIR=0, HK\_LFR\_VHDL\_CAL=0  HK\_LFR\_SC\_V\_F3, HK\_LFR\_SC\_E1\_F3,  HK\_LFR\_SC\_E2\_F3  Error counters  HK\_LFR\_LE\_CNT = 1  Time counters :  HK\_LFR\_TIMECODE\_ERRONEOUS=0, **HK\_LFR\_TIMECODE\_MISSING=1,** HK\_LFR\_TIMECODE\_INVALID=0, HK\_LFR\_TIME\_TIMECODE\_IT=0, HK\_LFR\_TIME\_NOT\_SYNCHRO=0, HK\_LFR\_TIME\_TIMECODE\_CTR=0  HK\_LFR\_TIMECODE\_MISSING because no automatic timecode.  16 S/C reaction wheel frequencies are unavailable   * FLAG = 0 (DISABLED)   The fields are HK\_LFR\_SC\_RWi\_Fj\_FLAG with i=[1,4] and  j=[1,4]  A lot of fields to zero (HK\_LFR\_DPU\_SPW\_PKT\_RCV\_CNT=0, HK\_LFR\_DPU\_SPW\_PKT\_SENT\_CNT=0….  HK report as a packet compliant to the PUS service n°3 (PACKET\_CATEGORY=”HK\_ROUTINE = 4”, PUS\_VERSION=”PUS\_VERSION = 1”, SERVICE\_TYPE=”HOUSEKEEPING\_AND\_DIAGNOSTIC\_DATA\_REPORTING = 3)”. | | | | | |  |  |
|  | LFR is in **STANDBY** mode. Send all the TC. | | | | | Verify the LFR FSW provides periodically a TM\_LFR\_HK packet (HK report) reflecting the status of the LFR.  HK\_LFR\_MODE = standby  HK\_LFR\_CALIB\_ENABLED: ENABLED = 1 when  TC\_LFR\_ \_ENABLE\_CALIBRATION is sent  HK\_LFR\_CALIB\_ENABLED: DISABLED = 0, when TC\_LFR\_DISABLE\_CALIBRATION is sent  SY\_LFR\_PAS\_FILTER\_ENABLED: ENABLED = 1 when  TC\_LFR\_LOAD\_FILTER\_PAR is sent with SY\_LFR\_PAS\_FILTER\_ENABLED\_D =1  updated parameters values  SY\_LFR\_BW, SY\_LFR\_SP0, SY\_LFR\_SP1,SY\_LFR\_R0, SY\_LFR\_R1, SY\_LFR\_R2  if TC\_LFR\_LOAD\_COMMON\_PAR is sent  updated HK\_LFR\_SC\_RWi\_Fj\_FLAG with i=[1,4] and  j=[1,4]  if TC\_LFR\_UPDATE\_INFO is sent (CP\_RPW\_SC\_RWi\_Fj = Nan or float value)  FLAG=0 if Nan  FLAG=1 if float value  HK report as a packet compliant to the PUS service n°3 (PACKET\_CATEGORY=”HK\_ROUTINE = 4”, PUS\_VERSION=”PUS\_VERSION = 1”, SERVICE\_TYPE=”HOUSEKEEPING\_AND\_DIAGNOSTIC\_DATA\_REPORTING = 3)”. | | | | | |  |  |
|  | LFR is in **NORMAL** mode. Send all the TC. | | | | | Verify the LFR FSW provides periodically a TM\_LFR\_HK packet (HK report) reflecting the status of the LFR.  HK\_LFR\_MODE = NORMAL  HK\_LFR\_CALIB\_ENABLED: ENABLED = 1 when  TC\_LFR\_ \_ENABLE\_CALIBRATION is sent  HK\_LFR\_CALIB\_ENABLED: DISABLED = 0, when TC\_LFR\_DISABLE\_CALIBRATION is sent  SY\_LFR\_PAS\_FILTER\_ENABLED: ENABLED = 1 when  TC\_LFR\_LOAD\_FILTER\_PAR is sent with SY\_LFR\_PAS\_FILTER\_ENABLED\_D =1  updated parameters values  SY\_LFR\_BW, SY\_LFR\_SP0, SY\_LFR\_SP1,SY\_LFR\_R0, SY\_LFR\_R1, SY\_LFR\_R2  if TC\_LFR\_LOAD\_COMMON\_PAR is sent  updated HK\_LFR\_SC\_RWi\_Fj\_FLAG with i=[1,4] and  j=[1,4]  if TC\_LFR\_UPDATE\_INFO is sent (CP\_RPW\_SC\_RWi\_Fj = Nan or float value)  FLAG=0 if Nan  FLAG=1 if float value  HK report as a packet compliant to the PUS service n°3 (PACKET\_CATEGORY=”HK\_ROUTINE = 4”, PUS\_VERSION=”PUS\_VERSION = 1”, SERVICE\_TYPE=”HOUSEKEEPING\_AND\_DIAGNOSTIC\_DATA\_REPORTING = 3)”. | | | | | |  |  |
|  | LFR is in **BURST** mode. Send all the TC. | | | | | Verify the LFR FSW provides periodically a TM\_LFR\_HK packet (HK report) reflecting the status of the LFR.  HK\_LFR\_MODE = BURST  HK\_LFR\_CALIB\_ENABLED: ENABLED = 1 when  TC\_LFR\_ \_ENABLE\_CALIBRATION is sent  HK\_LFR\_CALIB\_ENABLED: DISABLED = 0, when TC\_LFR\_DISABLE\_CALIBRATION is sent  SY\_LFR\_PAS\_FILTER\_ENABLED: ENABLED = 1 when  TC\_LFR\_LOAD\_FILTER\_PAR is sent with SY\_LFR\_PAS\_FILTER\_ENABLED\_D =1  updated parameters values  SY\_LFR\_BW, SY\_LFR\_SP0, SY\_LFR\_SP1,SY\_LFR\_R0, SY\_LFR\_R1, SY\_LFR\_R2  if TC\_LFR\_LOAD\_COMMON\_PAR is sent  updated HK\_LFR\_SC\_RWi\_Fj\_FLAG with i=[1,4] and  j=[1,4]  if TC\_LFR\_UPDATE\_INFO is sent (CP\_RPW\_SC\_RWi\_Fj = Nan or float value)  FLAG=0 if Nan  FLAG=1 if float value  HK report as a packet compliant to the PUS service n°3 (PACKET\_CATEGORY=”HK\_ROUTINE = 4”, PUS\_VERSION=”PUS\_VERSION = 1”, SERVICE\_TYPE=”HOUSEKEEPING\_AND\_DIAGNOSTIC\_DATA\_REPORTING = 3)”. | | | | | |  |  |
|  | LFR is in **SBM1** mode. Send all the TC. | | | | | Verify the LFR FSW provides periodically a TM\_LFR\_HK packet (HK report) reflecting the status of the LFR.  HK\_LFR\_MODE = SBM1  HK\_LFR\_CALIB\_ENABLED: ENABLED = 1 when  TC\_LFR\_ \_ENABLE\_CALIBRATION is sent  HK\_LFR\_CALIB\_ENABLED: DISABLED = 0, when TC\_LFR\_DISABLE\_CALIBRATION is sent  SY\_LFR\_PAS\_FILTER\_ENABLED: ENABLED = 1 when  TC\_LFR\_LOAD\_FILTER\_PAR is sent with SY\_LFR\_PAS\_FILTER\_ENABLED\_D =1  updated parameters values  SY\_LFR\_BW, SY\_LFR\_SP0, SY\_LFR\_SP1,SY\_LFR\_R0, SY\_LFR\_R1, SY\_LFR\_R2  if TC\_LFR\_LOAD\_COMMON\_PAR is sent  updated HK\_LFR\_SC\_RWi\_Fj\_FLAG with i=[1,4] and  j=[1,4]  if TC\_LFR\_UPDATE\_INFO is sent (CP\_RPW\_SC\_RWi\_Fj = Nan or float value)  FLAG=0 if Nan  FLAG=1 if float value  HK report as a packet compliant to the PUS service n°3 (PACKET\_CATEGORY=”HK\_ROUTINE = 4”, PUS\_VERSION=”PUS\_VERSION = 1”, SERVICE\_TYPE=”HOUSEKEEPING\_AND\_DIAGNOSTIC\_DATA\_REPORTING = 3)”. | | | | | |  |  |
|  | LFR is in **SBM2** mode. Send all the TC. | | | | | Verify the LFR FSW provides periodically a TM\_LFR\_HK packet (HK report) reflecting the status of the LFR.  HK\_LFR\_MODE = SBM2  HK\_LFR\_CALIB\_ENABLED: ENABLED = 1 when  TC\_LFR\_ \_ENABLE\_CALIBRATION is sent  HK\_LFR\_CALIB\_ENABLED: DISABLED = 0, when TC\_LFR\_DISABLE\_CALIBRATION is sent  SY\_LFR\_PAS\_FILTER\_ENABLED: ENABLED = 1 when  TC\_LFR\_LOAD\_FILTER\_PAR is sent with SY\_LFR\_PAS\_FILTER\_ENABLED\_D =1  updated parameters values  SY\_LFR\_BW, SY\_LFR\_SP0, SY\_LFR\_SP1,SY\_LFR\_R0, SY\_LFR\_R1, SY\_LFR\_R2  if TC\_LFR\_LOAD\_COMMON\_PAR is sent  updated HK\_LFR\_SC\_RWi\_Fj\_FLAG with i=[1,4] and  j=[1,4]  if TC\_LFR\_UPDATE\_INFO is sent (CP\_RPW\_SC\_RWi\_Fj = Nan or float value)  FLAG=0 if Nan  FLAG=1 if float value  HK report as a packet compliant to the PUS service n°3 (PACKET\_CATEGORY=”HK\_ROUTINE = 4”, PUS\_VERSION=”PUS\_VERSION = 1”, SERVICE\_TYPE=”HOUSEKEEPING\_AND\_DIAGNOSTIC\_DATA\_REPORTING = 3)”. | | | | | |  |  |
|  | Run LFR FSW  Send a automatic timecode  Send a  TC\_LFR\_UPDATE\_TIME = PC current time  with a valid time code  Wait 10 s | | | | | Verify the LFR FSW provides periodically a TM\_LFR\_HK packet (HK report) reflecting the status of the LFR.  After the boot  Error counters  HK\_LFR\_LE\_CNT = 1  last error is HK\_LFR\_ERR\_RID=42119, HK\_LFR\_ER\_CODE= 24 , HK\_LFR\_ER\_TIME= time  Time counters :  HK\_LFR\_TIMECODE\_ERRONEOUS=0, **HK\_LFR\_TIMECODE\_MISSING=1,** HK\_LFR\_TIMECODE\_INVALID=0, HK\_LFR\_TIME\_TIMECODE\_IT=0, HK\_LFR\_TIME\_NOT\_SYNCHRO=0, HK\_LFR\_TIME\_TIMECODE\_CTR=0  HK\_LFR\_DPU\_SPW\_TICK\_OUT\_CNT is incremented  HK\_LFR\_DPU\_SPW\_LAST\_TIMC=timecode  timecode analyze will updated  HK\_LFR\_TIME\_TIMECODE\_ IT = 1  and  last error is HK\_LFR\_ERR\_RID=42119, HK\_LFR\_ER\_CODE= 24 , HK\_LFR\_ER\_TIME= time  HK\_LFR\_LE\_CNT is incremented  LFR is synchronized  HK\_LFR\_UPDATE\_TIME\_TC\_CNT is incremented  HK\_LFR\_DPU\_SPW\_TICK\_OUT\_CNT is incremented  HK\_LFR\_DPU\_SPW\_LAST\_TIMC=2  HK\_LFR\_TIME\_TIMECODE\_ IT = is incremented  and  last error is HK\_LFR\_ERR\_RID=42119, HK\_LFR\_ER\_CODE= 24 , HK\_LFR\_ER\_TIME= time  HK\_LFR\_LE\_CNT is incremented  HK\_LFR\_DPU\_SPW\_TICK\_OUT\_CNT is incremented  HK\_LFR\_DPU\_SPW\_LAST\_TIMC=timecode\_  HK\_LFR\_TIME\_TIMECODE\_ IT is incremented  HK\_LFR\_TIME\_TIMECODE\_CTR is incremented  last error is HK\_LFR\_ERR\_RID=42119, HK\_LFR\_ER\_CODE= 26 , HK\_LFR\_ER\_TIME= time  HK\_LFR\_LE\_CNT is incremented  End of test  HK\_LFR\_LE\_CNT=33  HK\_LFR\_DPU\_SPW\_TICK\_OUT\_CNT=17  HK\_LFR\_DPU\_SPW\_LAST\_TIMC=17  HK\_LFR\_TIMECODE\_MISSING=1  HK\_LFR\_TIME\_TIMECODE\_IT=17  HK\_LFR\_TIME\_TIMECODE\_CTR=15 | | | | | |  |  |
|  | Reset LFR  Run LFR FSW  and launch  automatic timecode during  boot sequence  wait 10s | | | | | Verify the LFR FSW provides periodically a TM\_LFR\_HK packet (HK report) reflecting the status of the LFR.  After the boot  Error counters  HK\_LFR\_LE\_CNT = 0  last error is HK\_LFR\_ERR\_RID=0, HK\_LFR\_ER\_CODE= 0 , HK\_LFR\_ER\_TIME= 0  Time counters :  HK\_LFR\_TIMECODE\_ERRONEOUS=0, HK\_LFR\_TIMECODE\_MISSING=0,HK\_LFR\_TIMECODE\_INVALID=0, HK\_LFR\_TIME\_TIMECODE\_IT=0, HK\_LFR\_TIME\_NOT\_SYNCHRO=0, HK\_LFR\_TIME\_TIMECODE\_CTR=0  End of test  No timecode error.  HK\_LFR\_DPU\_SPW\_TICK\_OUT\_CNT=16, HK\_LFR\_DPU\_SPW\_LAST\_TIMC=16 | | | | | |  |  |
|  | Reset LFR  Run LFR FSW  after boot sequence  launch automatic timecode  wait 10s | | | | | Verify the LFR FSW provides periodically a TM\_LFR\_HK packet (HK report) reflecting the status of the LFR.  After the boot  Error counters  HK\_LFR\_LE\_CNT = 1  last error is HK\_LFR\_ERR\_RID=42119, HK\_LFR\_ER\_CODE= 24 , HK\_LFR\_ER\_TIME= time  Time counters :  HK\_LFR\_TIMECODE\_ERRONEOUS=0, **HK\_LFR\_TIMECODE\_MISSING=1,** HK\_LFR\_TIMECODE\_INVALID=0, HK\_LFR\_TIME\_TIMECODE\_IT=0, HK\_LFR\_TIME\_NOT\_SYNCHRO=0, HK\_LFR\_TIME\_TIMECODE\_CTR=0  **HK\_LFR\_TIME\_TIMECODE\_IT** is incremented  last error is HK\_LFR\_ERR\_RID=42119, HK\_LFR\_ER\_CODE= 24 , HK\_LFR\_ER\_TIME= time  HK\_LFR\_LE\_CNT is incremented  HK\_LFR\_DPU\_SPW\_TICK\_OUT\_CNT is incremented  HK\_LFR\_DPU\_SPW\_LAST\_TIMC=timecode  End of test  HK\_LFR\_LE\_CNT=13  HK\_LFR\_DPU\_SPW\_TICK\_OUT\_CNT=12, HK\_LFR\_DPU\_SPW\_LAST\_TIMC=12  HK\_LFR\_TIMECODE\_MISSING=1  HK\_LFR\_TIME\_TIMECODE\_IT=12 | | | | | |  |  |
|  | Long tests (15h) | | | | | Verify the LFR FSW provides periodically a TM\_LFR\_HK packet (HK report) reflecting the status of the LFR.  HK report as a packet compliant to the PUS service n°3 (PACKET\_CATEGORY=”HK\_ROUTINE = 4”, PUS\_VERSION=”PUS\_VERSION = 1”, SERVICE\_TYPE=”HOUSEKEEPING\_AND\_DIAGNOSTIC\_DATA\_REPORTING = 3)”. | | | | | |  |  |
| 11 | End | | | | |  | | | | | |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE** | | | | | | | | | | | | |
| **Test procedure identifier** | | | | | | **Date** | | | | **Assessment** | | |
| SVS-0058\_Ed2 | | | | | | DD/MM/YYYY | | | | NT / NA / NOK / POK / OK | | |
| **TS Tested requirements** | | | | | | **RB Tested requirements** | | | | | | |
| N/A | | | | | | REQ-LFR-SRS-5232\_Ed1 | | | | | | |
| **Component:** | | | | |  | **Version:** | V3 | | | | | |
| **Involved subsystems** | | | | | | | | | | | | |
| SW: | | 3.2.0.24 | | | | | | | | | | |
| HW: | | 1.1.91 StarDundee | | | | | | | | | | |
| Start time: | | | (DD/MM/YYYY) hh:mm | | | Test Operator: | | |  | | | |
| End Time: | | | (DD/MM/YYYY) hh:mm | | | Test Duration: | | 24s | | | | |
| **Purpose:** | | | When the watchdog reaches 0, the LFR FSW will stop. | | | | | | | | | |
| **General remarks about the test** | | | | | | | | | | | | |
| Requirement Title | | | | Watchdog management | | | | | | | | |
| dependencies | | | | SSS-CP-FS-540 | | | | | | | | |
| restrictions | | | |  | | | | | | | | |
| means | | | |  | | | | | | | | |
| input data | | | |  | | | | | | | | |
| prerequisite | | | |  | | | | | | | | |
| used test programs | | | |  | | | | | | | | |
| test script | | | | test\_watchdog.py | | | | | | | | |
| output data | | | |  | | | | | | | | |
| notes | | | | LFR watchdog is implemented at software level and the only action is to exit (0). | | | | | | | | |
| **Procedure steps** | | | | | | | | | | | | |
| **Step** | **Actions** | | | | | **Expected Results** | | | | | **Comments** | **Status** |
|  | Run Fsw,  in standby mode rwrite 0 into the watchdog register | | | | | LFR FSW stops. | | | | |  |  |
|  | End | | | | |  | | | | |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE** | | | | | | | | | | | | |
| **Test procedure identifier** | | | | | | **Date** | | | | **Assessment** | | |
| SVS-0059\_Ed2 | | | | | | DD/MM/YYYY | | | | NT / NA / NOK / POK / OK | | |
| **TS Tested requirements** | | | | | | **RB Tested requirements** | | | | | | |
| N/A | | | | | | REQ-LFR-SRS-5544\_Ed2 REQ-LFR-SRS-5208\_Ed1  REQ-LFR-SRS-5210\_Ed1 REQ-LFR-SRS-5202\_Ed1  REQ-LFR-SRS-5545\_Ed1 | | | | | | |
| **Component:** | | | | |  | **Version:** | V2 V1 | | | | | |
| **Involved subsystems** | | | | | | | | | | | | |
| SW: | | 3.2.0.24 | | | | | | | | | | |
| HW: | | 1.1.91 StarDundee | | | | | | | | | | |
| Start time: | | | (DD/MM/YYYY) hh:mm | | | Test Operator: | | |  | | | |
| End Time: | | | (DD/MM/YYYY) hh:mm | | | Test Duration: | | 24s | | | | |
| **Purpose:** | | | Check the LFR FSW produces the command acknowledgment packets. | | | | | | | | | |
| **General remarks about the test** | | | | | | | | | | | | |
| Requirement Title | | | | Equipment command feedback | | | | | | | | |
| dependencies | | | | SSS-CP-EQS-150 SSS-CP-FS-085 SSS-CP-FS-095 SSS-CP-FS-100 SSS-CP-EQS-160 | | | | | | | | |
| restrictions | | | |  | | | | | | | | |
| means | | | |  | | | | | | | | |
| input data | | | |  | | | | | | | | |
| prerequisite | | | |  | | | | | | | | |
| used test programs | | | |  | | | | | | | | |
| test script | | | | ack.py | | | | | | | | |
| output data | | | |  | | | | | | | | |
| notes | | | | Command acknowledgment packets are compliant to the PUS service n°1.  See REQ-LFR-SRS-5543\_Ed2  TM\_LFR\_TC\_EXE\_ERROR command is validated by design  All LFR TC are implemented | | | | | | | | |
| **Procedure steps** | | | | | | | | | | | | |
| **Step** | **Actions** | | | | | **Expected Results** | | | | | **Comments** | **Status** |
|  | In each mode, transmit each command to LFR in case of execution success and not executable TC. | | | | | Verify the LFR FSW produces the TM\_LFR\_TC\_EXE\_SUCCESS, TM\_LFR\_TC\_EXE\_ NOT\_EXECUTABLE, command acknowledgment packet. | | | | |  |  |
|  | Transmit a command to LFR in case of error detected during the acceptance stage verifications (Illegal APID, Illegal packet type, Illegal packet subtype, wrong CRC, wrong or incomplete length). | | | | | Verify the LFR FSW produces the TM\_LFR\_TC\_EXE\_CORRUPTED command acknowledgment packet. | | | | |  |  |
|  | Transmit a command to LFR in case of wrong or inconsistent field (in the header fields or in the data fields. | | | | | Verify the LFR FSW produces the TM\_LFR\_TC\_EXE\_INCONSISTENT command acknowledgment packet. | | | | |  |  |
|  | End | | | | |  | | | | |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE** | | | | | | | | | | | | |
| **Test procedure identifier** | | | | | | **Date** | | | | **Assessment** | | |
| SVS-0060\_Ed1 | | | | | | DD/MM/YYYY | | | | NT / NA / NOK / POK / OK | | |
| **TS Tested requirements** | | | | | | **RB Tested requirements** | | | | | | |
| N/A | | | | | | REQ-LFR-SRS-5536\_Ed1 | | | | | | |
| **Component:** | | | | |  | **Version:** | V1 | | | | | |
| **Involved subsystems** | | | | | | | | | | | | |
| SW: | | 3.2.0.24 | | | | | | | | | | |
| HW: | | 1.1.91 StarDundee | | | | | | | | | | |
| Start time: | | | (DD/MM/YYYY) hh:mm | | | Test Operator: | | |  | | | |
| End Time: | | | (DD/MM/YYYY) hh:mm | | | Test Duration: | | 39h | | | | |
| **Purpose:** | | | The generation period of the HK equipment report packets is fixed to SY\_LFR\_HK\_SAMPLING\_PER = 1 second. | | | | | | | | | |
| **General remarks about the test** | | | | | | | | | | | | |
| Requirement Title | | | | Equipment HK reporting | | | | | | | | |
| dependencies | | | | SSS-CP-EQS-070 | | | | | | | | |
| restrictions | | | |  | | | | | | | | |
| means | | | |  | | | | | | | | |
| input data | | | |  | | | | | | | | |
| prerequisite | | | |  | | | | | | | | |
| used test programs | | | |  | | | | | | | | |
| test script | | | | test\_mode\_burst.py  test\_mode\_normal.py  test\_mode\_SBMx\_12h.py | | | | | | | | |
| output data | | | |  | | | | | | | | |
| notes | | | | periodicity.py checks the generation period of HK report fixed to SY\_LFR\_HK\_SAMPLING\_PER = 1  verif\_fields.py checks all TM\_LFR\_HK in all validation tests. | | | | | | | | |
| **Procedure steps** | | | | | | | | | | | | |
| **Step** | **Actions** | | | | | **Expected Results** | | | | | **Comments** | **Status** |
| 1 | LFR is in NORMAL Mode  during 13:40 hours | | | | | Verify the periodicity of TM\_LFR\_HK  mean of all HK delta times = 0.99999443s  median of all HK delta times : 0.999985s  **52339 TM\_LFR\_HK generated** | | | | |  |  |
| 2 | LFR is BURST Mode  during 12hours | | | | | Verify the periodicity of TM\_LFR\_HK  Mean of all HK delta times : 0.9999928759 s  Median of all HK delta times : 1.0.s  **43200 TM\_LFR\_HK generated** | | | | |  |  |
| 3 | LFR is SBM1 Mode  during hours  then  SBM2 during 6hours | | | | | Verify the periodicity of TM\_LFR\_HK  mean of all HK = 0.99999308s  median of all HK delta times : 1.0s  **44425 TM\_LFR\_HK generated** | | | | |  |  |
| 4 | LFR is in NORMAL MODE (3630s)  then  LFR is in SBM1 MODE (1800s)  then  LFR is in NORMAL MODE (3600s)  then  LFR is in SBM2 MODE (1800s)  then  LFR is in NORMAL MODE (3600s)  then  LFR is in SBM1 MODE (1800s)  then  LFR is in SBM2 MODE (1800s)  then  LFR is in SBM1 MODE (1830s)  then  LFR is in NORMAL MODE (3600s) | | | | | Verify the periodicity of TM\_LFR\_HK  mean of all HK = 0.99999450  median of all HK delta times : 0.999985s  **23459 TM\_LFR\_HK generated** | | | | |  |  |
| 5 | End | | | | |  | | | | |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE** | | | | | | | | | | | | |
| **Test procedure identifier** | | | | | | **Date** | | | | **Assessment** | | |
| SVS-0061\_Ed1 | | | | | | DD/MM/YYYY | | | | NT / NA / NOK / POK / OK | | |
| **TS Tested requirements** | | | | | | **RB Tested requirements** | | | | | | |
| N/A | | | | | | REQ-LFR-SRS-5540\_Ed1 | | | | | | |
| **Component:** | | | | |  | **Version:** | V2 | | | | | |
| **Involved subsystems** | | | | | | | | | | | | |
| SW: | | 3.2.0.24 | | | | | | | | | | |
| HW: | | 1.1.91 StarDundee | | | | | | | | | | |
| Start time: | | | (DD/MM/YYYY) hh:mm | | | Test Operator: | | |  | | | |
| End Time: | | | (DD/MM/YYYY) hh:mm | | | Test Duration: | | N/A | | | | |
| **Purpose:** | | | The LFR FSW shall report the normal progress of operations and activities having an operational significance by updating the value of the suitable status parameter in its periodic HK report. | | | | | | | | | |
| **General remarks about the test** | | | | | | | | | | | | |
| Requirement Title | | | | Equipment Event reporting | | | | | | | | |
| dependencies | | | | SSS-CP-EQS-110 | | | | | | | | |
| restrictions | | | |  | | | | | | | | |
| means | | | |  | | | | | | | | |
| input data | | | |  | | | | | | | | |
| prerequisite | | | |  | | | | | | | | |
| used test programs | | | |  | | | | | | | | |
| test script | | | | all tests | | | | | | | | |
| output data | | | |  | | | | | | | | |
| notes | | | | Only dynamic fields are described and this list is not exhaustive.  verif\_fields.py checks all TM\_LFR\_HK in all validation tests. | | | | | | | | |
| **Procedure steps** | | | | | | | | | | | | |
| **Step** | **Actions** | | | | | **Expected Results** | | | | | **Comments** | **Status** |
| 1 | Common parameters sent by TC\_LFR\_LOAD\_COMMON\_PAR | | | | | in TM\_LFR\_HK,  SY\_LFR\_BW, SY\_LFR\_SP0, SY\_LFR\_SP1,SY\_LFR\_R0, SY\_LFR\_R1,SY\_LFR\_R2 | | | | | See  SVS-0095 |  |
| 2 | Calibration mode activity sent by  TC\_LFR\_ENABLE\_CALIBRATION  or  TC\_LFR\_DISABLE\_CALIBRATION | | | | | in TM\_LFR\_HK  HK\_LFR\_CALIB\_ENABLED | | | | | See  SVS-0053 |  |
| 3 | LFR mode sent by  TC\_LFR\_ENTER\_MODE | | | | | in TM\_LFR\_HK  HK\_LFR\_MODE = current working mode | | | | | See  SVS-0034 |  |
| 4 | TC\_LFR\_LOAD\_FILTER\_PAR | | | | | in TM\_LFR\_HK  SY\_LFR\_PAS\_FILTER\_ENABLED\_D | | | | | See  SVS-0096  SVS-1202 |  |
| 5 | Specific LFR status | | | | | in TM\_LFR\_HK  HK\_LFR\_VHDL\_SM  HK\_LFR\_VHDL\_IIR  HK\_LFR\_VHDL\_CAL  HK\_LFR\_SC\_V\_F3  HK\_LFR\_SC\_E1\_F3  HK\_LFR\_SC\_E2\_F3 | | | | | all tests  CTC-800 |  |
| 6 | TC management  ( acceptance, execution, statistics) | | | | | in TM\_LFR\_HK  TC\_LFR\_UPDATE\_xxx\_CNT (TIME, INFO)  execution TC  HK\_LFR\_EXE\_xxxx  Reject TC  HK\_LFR\_REJ\_xxxx | | | | | See  SVS-0003  to  SVS-0010 |  |
| 7 | Errors management  (statistics anomalies)  TC are managed in step 5 | | | | | in TM\_LFR\_HK  Timing errors  Spacewire errors  Anomaly\_statistics in HK\_LFR\_LE\_CNT and  HK\_LFR\_ME\_CNT | | | | | SVS-0012 to  SVS-0013  SVS-0026  SVS-0064 |  |
| 8 | resource statistics | | | | | in TM\_LFR\_HK  HK\_LFR\_CPU\_LOAD  HK\_LFR\_CPU\_LOAD\_MAX  HK\_LFR\_CPU\_LOAD\_AVE  HK\_LFR\_Q\_SD\_FIFO\_SIZE\_MAX,  HK\_LFR\_Q\_RV\_FIFO\_SIZE\_MAX, HK\_LFR\_Q\_P0\_FIFO\_SIZE\_MAX, HK\_LFR\_Q\_P1\_FIFO\_SIZE\_MAX  HK\_LFR\_Q\_P2\_FIFO\_SIZE\_MAX  about spacewire interface  HK\_LFR\_DPU\_SPW\_PKT\_RCV\_CNT, HK\_LFR\_DPU\_SPW\_PKT\_SENT\_CNT  HK\_LFR\_DPU\_SPW\_TICK\_OUT\_CNT  HK\_LFR\_DPU\_SPW\_LAST\_TIMC | | | | | All tests |  |
| 9 | Reaction Wheels Frequencies  TC\_LFR\_UPDATE\_INFO  field CP\_RPW\_SC\_RWi\_Fi i=[1,4] and j=[1,4]  either 16 frequencies | | | | | In TM\_LFR\_HK  HK\_LFR\_SC\_RWi\_Fj\_FLAG i=[1,4] and j=[1,4] | | | | | SVS-1100 |  |
| 10 | End | | | | |  | | | | |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE** | | | | | | | | | | | | | |
| **Test procedure identifier** | | | | | | | **Date** | | | | **Assessment** | | |
| SVS-0062\_Ed1 | | | | | | | DD/MM/YYYY | | | | NT / NA / NOK / POK / OK | | |
| **TS Tested requirements** | | | | | | | **RB Tested requirements** | | | | | | |
| N/A | | | | | | | REQ-LFR-SRS-5561\_Ed1 | | | | | | |
| **Component:** | | | | |  | | **Version:** | V2 | | | | | |
| **Involved subsystems** | | | | | | | | | | | | | |
| SW: | | 3.2.0.24 | | | | | | | | | | | |
| HW: | | 1.1.91 StarDundee | | | | | | | | | | | |
| Start time: | | | (DD/MM/YYYY) hh:mm | | | | Test Operator: | | |  | | | |
| End Time: | | | (DD/MM/YYYY) hh:mm | | | | Test Duration: | | N/A | | | | |
| **Purpose:** | | | Check the LFR procedure after the loss of the SpW connection, if the analyzer has failed to (re-)establish the connection within a SY\_LFR\_DPU\_CONNECT\_TIMEOUT timeout period. | | | | | | | | | | |
| **General remarks about the test** | | | | | | | | | | | | | |
| Requirement Title | | | | SpaceWire link monitoring | | | | | | | | | |
| dependencies | | | | SSS-CP-EQS-153 | | | | | | | | | |
| restrictions | | | |  | | | | | | | | | |
| means | | | |  | | | | | | | | | |
| input data | | | |  | | | | | | | | | |
| prerequisite | | | |  | | | | | | | | | |
| used test programs | | | |  | | | | | | | | | |
| test script | | | | Manually tested. | | | | | | | | | |
| output data | | | |  | | | | | | | | | |
| notes | | | | Inspection is better to valid this requirement | | | | | | | | | |
| **Procedure steps** | | | | | | | | | | | | | |
| **Step** | **Actions** | | | | | **Expected Results** | | | | | | **Comments** | **Status** |
|  | Realize a loss of the SpW connection (disconnect the SpW).  Fail the LFR analyzer to establish the connection within a SY\_LFR\_DPU\_CONNECT\_TIMEOUT timeout period. | | | | | Verify the LFR FSW:  -Resets the SpW interface.  -Resets the connection timeout period.  -Starts again the connection process. | | | | | |  |  |
|  | Fail the LFR analyzer to re-establish the connection within a SY\_LFR\_DPU\_CONNECT\_TIMEOUT timeout period. | | | | | Verify the LFR FSW:  -Resets the SpW interface.  -Resets the connection timeout period.  -Starts again the connection process. | | | | | |  |  |
|  | End | | | | |  | | | | | |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE** | | | | | | | | | | | | | |
| **Test procedure identifier** | | | | | | | **Date** | | | | **Assessment** | | |
| SVS-0063\_Ed1 | | | | | | | DD/MM/YYYY | | | | NT / NA / NOK / POK / OK | | |
| **TS Tested requirements** | | | | | | | **RB Tested requirements** | | | | | | |
| N/A | | | | | | | REQ-LFR-SRS-5562\_Ed1 | | | | | | |
| **Component:** | | | | |  | | **Version:** | V2 | | | | | |
| **Involved subsystems** | | | | | | | | | | | | | |
| SW: | | 3.2.0.24 | | | | | | | | | | | |
| HW: | | 1.1.91 StarDundee | | | | | | | | | | | |
| Start time: | | | (DD/MM/YYYY) hh:mm | | | | Test Operator: | | |  | | | |
| End Time: | | | (DD/MM/YYYY) hh:mm | | | | Test Duration: | | N/A | | | | |
| **Purpose:** | | | Check the LFR FSW performs up to SY\_LFR\_DPU\_CONNECT\_ATTEMPT attempts of connection. | | | | | | | | | | |
| **General remarks about the test** | | | | | | | | | | | | | |
| Requirement Title | | | | SpaceWire link monitoring | | | | | | | | | |
| dependencies | | | | SSS-CP-EQS-154 | | | | | | | | | |
| restrictions | | | |  | | | | | | | | | |
| means | | | |  | | | | | | | | | |
| input data | | | |  | | | | | | | | | |
| prerequisite | | | |  | | | | | | | | | |
| used test programs | | | |  | | | | | | | | | |
| test script | | | | Manually | | | | | | | | | |
| output data | | | |  | | | | | | | | | |
| notes | | | | Minicom tool shows the attemps of connection | | | | | | | | | |
| **Procedure steps** | | | | | | | | | | | | | |
| **Step** | **Actions** | | | | | **Expected Results** | | | | | | **Comments** | **Status** |
|  | The LFR FSW realizes SY\_LFR\_DPU\_CONNECT\_ATTEMPT-1 unsuccessful attempts of connection. | | | | | The LFR FSW tries one more time a connection. | | | | | | Used Minicom |  |
|  | End | | | | |  | | | | | |  |  |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE** | | | | | | | | | | |
| **Test procedure identifier** | | | | **Date** | | | | **Assessment** | | |
| SVS-0064\_Ed1 | | | | DD/MM/YYYY | | | | NT / NA / NOK / POK / OK | | |
| **TS Tested requirements** | | | | **RB Tested requirements** | | | | | | |
| N/A | | | | REQ-LFR-SRS-5563\_Ed1 | | | | | | |
| **Component:** | | |  | **Version:** | | V2 | | | | |
| **Involved subsystems** | | | | | | | | | | |
| SW: | 3.2.0.24 | | | | | | | | | |
| HW: | 1.1.91 StarDundee | | | | | | | | | |
| Start time: | | (DD/MM/YYYY) hh:mm | | Test Operator: | | |  | | | |
| End Time: | | (DD/MM/YYYY) hh:mm | | Test Duration: | | | 2mn | | | |
| **Purpose:** | | Check, after SY\_LFR\_DPU\_CONNECT\_ATTEMPT unsuccessful attempts of connection, the LFR FSW enters into STANDBY. | | | | | | | | |
| **General remarks about the test** | | | | | | | | | | |
| Requirement Title | | | SpaceWire link monitoring | | | | | | | |
| dependencies | | | SSS-CP-EQS-155 | | | | | | | |
| restrictions | | |  | | | | | | | |
| means | | |  | | | | | | | |
| input data | | |  | | | | | | | |
| prerequisite | | |  | | | | | | | |
| used test programs | | |  | | | | | | | |
| test script | | | spw\_failure.py | | | | | | | |
| output data | | |  | | | | | | | |
| notes | | | WARNING: We disconnect the spacewire plug manually so  it ‘s difficult to obtain always the same SPW errors. | | | | | | | |
| **Procedure steps** | | | | | | | | | | |
| **Step** | **Actions** | | | | **Expected Results** | | | | **Comments** | **Status** |
|  | Run the LFR FSW with automatic timecodes  For each mode  The LFR FSW realizes SY\_LFR\_DPU\_CONNECT\_ATTEMPT unsuccessful attempts of connection. | | | | Verify after SY\_LFR\_DPU\_CONNECT\_ATTEMPT unsuccessful attempts of connection, the LFR FSW enters into STANDBY.  We observed spacewire errors in the TM\_LFR\_HK  HK\_LFR\_DPU\_SPW\_PARITYHK\_LFR\_DPU\_SPW\_DISCONNECT  HK\_LFR\_DPU\_SPW\_ ESCAPE  and HK\_LFR\_LE\_CNT is updated.  Sometimes  HK\_LFR\_DPU\_SPW\_EARLY\_EOP and HK\_LFR\_ME\_CNT is updated.  HK\_LFR\_TIMECODE\_ERRONEOUS and HK\_LFR\_TIMECODE\_INVALID are incremented.  The last error kept into  HK\_LFR\_ER\_RID  HK\_LFR\_ER\_CODE  HK\_LFR\_ER\_TIME | | | |  |  |
|  | End | | | |  | | | |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE** | | | | | | | | | | | |
| **Test procedure identifier** | | | | | | **Date** | | | **Assessment** | | |
| SVS-0065\_Ed1 | | | | | | DD/MM/YYYY | | | NT / NA / N / POK / OK | | |
| **TS Tested requirements** | | | | | | **RB Tested requirements** | | | | | |
| N/A | | | | | | REQ-LFR-SRS-5551\_Ed1 | | | | | |
| **Component:** | | | |  | | **Version:** | V2 | | | | |
| **Involved subsystems** | | | | | | | | | | | |
| SW: | | 3.2.0.24 | | | | | | | | | |
| HW: | | 1.1.91 StarDundee | | | | | | | | | |
| Start time: | | | (DD/MM/YYYY) hh:mm | | | Test Operator: | |  | | | |
| End Time: | | | (DD/MM/YYYY) hh:mm | | | Test Duration: | | 27mn | | | |
| **Purpose:** | | | Check the LFR FSW allows dumping in TM\_LFR PARAMETER\_DUMP packets, upon the reception of TC\_LFR\_DUMP\_PAR command, all their functional and operational configuration parameters. | | | | | | | | |
| **General remarks about the test** | | | | | | | | | | | |
| Requirement Title | | | | Equipment parameter dump | | | | | | | |
| dependencies | | | | SSS-CP-EQS-215 | | | | | | | |
| restrictions | | | |  | | | | | | | |
| means | | | |  | | | | | | | |
| input data | | | | Reference\_values\_for\_diff.asc files to diff with extract.py output | | | | | | | |
| prerequisite | | | |  | | | | | | | |
| used test programs | | | |  | | | | | | | |
| test script | | | | Send\_nominal\_tc\_dump\_all\_modes.py + extract.py (step1)  Send\_modified\_tc\_dump\_all\_modes.py + extract.py (step2)  Send\_bad\_tc\_dump\_all\_modes.py + extract.py (step3 step 4) | | | | | | | |
| output data | | | |  | | | | | | | |
| notes | | | | TC created for R3 and R3+ are tested | | | | | | | |
| **Procedure steps** | | | | | | | | | | | |
| **Step** | **Actions** | | | | **Expected Results** | | | | | **Comments** | **Status** |
|  | This step must be launched after boot or reboot of fsw to ensure nominal mode and SEQUENCE\_CNT verification.  Send a TC\_LFR\_DUMP\_PAR to valid default parameters after the boot  Default parameters:  In each LFR modes,LFR receipts TC\_LFR\_DUMP\_PAR command  send  send a TC\_LFR\_UPDATE\_INFO  send a TC\_LFR\_DUMP\_PAR  send a TC\_LFR\_LOAD\_FBINS\_MASKS  send a TC\_LFR\_DUMP\_PAR  send a TC\_LFR\_LOAD\_FILTER\_PAR  send a TC\_LFR\_DUMP\_PAR | | | | After the boot  Verify the LFR FSW dumps in TM\_LFR PARAMETER\_DUMP packets all their functional and operational configuration parameters (software and hardware) for each mode and with default parameters.  Default values for section  SY\_LFR\_COMMON\_PARAMETERS  SY\_LFR\_NORMAL\_PARAMETERS  SY\_LFR\_BURST\_PARAMETERS  SY\_LFR\_SBM1\_PARAMETERS  SY\_LFR\_SBM2\_PARAMETERS  SY\_LFR\_RW\_MASK  SY\_LFR\_FBINS\_MASKS  SY\_LFR\_FILTER\_PARAMETERS  if a TC\_LFR\_UPDATE\_INFO is sent with default values (CP\_RPW\_SC\_RWi\_Fi = Nan with i=[1,4] ans j =[1,4]  On TM\_LFR\_PARAMETER\_DUMP  no changed on section SY\_LFR\_RW\_MASK (PA\_LFR\_RW\_MASK\_Fi\_WORDj with i=[1,4] ans j =[1,4]  if a TC\_LFR\_LOAD\_FBINS\_MASK is sent with default values  (FBINS\_Fi\_WORDj with i=[1,4] ans j =[1,4] (bit=1)  On TM\_LFR\_PARAMETER\_DUMP  no changed on section SY\_LFR\_RW\_MASK (SY\_LFR\_FBINS\_Fi\_WORDj with i=[1,4] ans j =[1,4] (bit=1)  if a TC\_LFR\_LOAD\_FILTER\_PAR is sent with default values  SY\_LFR\_PAS\_FILTER\_ENABLED\_D = ENABLED =1  On TM\_LFR\_PARAMETER\_DUMP  Changed on SY\_LFR\_PAS\_FILTER\_ENABLED\_D field = ENABLED.  After the boot SY\_LFR\_PAS\_FILTER\_ENABLED\_D = DISABLED until a TC\_LFR\_LOAD\_FILTER should be sent. | | | | | We use diff between trace extract and a reference file with values expected |  |
|  | Modified parameters :  This step must be launched after boot or reboot of fsw  For each mode  Send a TC\_LFR\_COMMON\_MODE with  SY\_LFR\_BW=0  SY\_LFR\_SP0=1  SY\_LFR\_SP1=1  SY\_LFR\_R0=1  SY\_LFR\_R1=1  SY\_LFR\_R2=1  Send TC\_LFR\_DUMP\_PAR  Send a TC\_LFR\_COMMON\_MODE with default values  Send TC\_LFR\_DUMP\_PAR  For each mode send the allowed TC\_LFR\_LOAD\_\*\_PAR commands with coherent sample values (non default) and TC\_LFR\_DUMP\_PAR  For each mode send a  TC\_LFR\_UPDATE\_INFO with a floating value (not a Nan) CP\_RPW\_RW1\_F1 and CP\_RPW\_RW1\_F2 and a TC\_LFR\_DUMP\_PAR  For each mode send a  TC\_LFR\_LOAD\_FBINS\_MASK with FBINS\_F0\_WORDi = 0  i = [1,4]  and a TC\_LFR\_DUMP\_PAR  For each mode send a  TC\_LFR\_LOAD\_FILTER\_PAR with  SY\_LFR\_PAS\_FILTER\_ENABLED = ENABLED  SY\_LFR\_PAS\_FILTER\_TBAD = 3.0  SY\_LFR\_PAS\_FILTER\_OFFSET = 1  SY\_LFR\_PAS\_FILTER\_SHIFT=0  and a TC\_LFR\_DUMP\_PAR | | | | Verify the LFR FSW dumps in TM\_LFR PARAMETER\_DUMP packets all their functional and operational configuration parameters (software and hardware) for each mode and that they are equal to injected ones.  Updated parameters for section  SY\_LFR\_COMMON\_PARAMETERS  SY\_LFR\_NORMAL\_PARAMETERS  SY\_LFR\_BURST\_PARAMETERS  SY\_LFR\_SBM1\_PARAMETERS  SY\_LFR\_SBM2\_PARAMETERS  SY\_LFR\_RW\_MASK  SY\_LFR\_FBINS\_MASKS  SY\_LFR\_FILTER\_PARAMETERS | | | | | We use diff between trace extract and a reference file with values expected |  |
|  | Bad parameters : This step must be launched after boot or reboot of fsw  Go to Standby mode  we first load non default parameters and  then try to load non authorized values(out of range or impossible) with all TC\_LFR\_LOAD\_\*\_PAR commands and  we send TC\_LFR\_DUMP\_PAR for each.  send a  TC\_LFR\_UPDATE\_INFO  with a negative rw frequency (CP\_RPW\_SC\_RWi\_Fi with i=[1,4] and j=[1,4]  and send a TC\_LFR\_DUMP\_PAR  send a TC\_LFR\_LOAD\_FBINS\_MASK  and send a TC\_LFR\_DUMP\_PAR  send a  TC\_LFR\_LOAD\_KCOEFFICIENTS  with one KCOEFF with negative value  and send a TC-LFR\_DUMP\_PAR  send  a TC\_LFR\_LOAD\_FILTER\_PAR with a bad value for each parameter (Modulus, Tbad, shift, offset , inconsistent Modulus vs shift + offset)  TC\_LFR\_DUMP\_PAR | | | | Verify that parameters contained in the TM\_LFR PARAMETER\_DUMP packets have not been updated to bad ones and not forced to default.  Each Parameter of the TC\_LFR\_LOAD\_\*\_PAR is ent with a bad value and coherence between parameters is also tested.  TC\_LFR\_LOAD\_COMMON\_PAR hasn’t bad parameter value.  TC\_LFR\_LOAD\_FBINS\_MASK hasn’t bad parameter value. | | | | | We use diff between trace extract and a reference file used in step 3 |  |
|  | For each scientific mode we first load non default parameters and then try to load non authorized values(out of range or impossible) with all TC\_LFR\_LOAD\_\*\_PAR commands and we send TC\_LFR\_DUMP\_PAR for each.  send  a TC\_LFR\_UPDATE\_INFO  with a negative rw frequency (CP\_RPW\_SC\_RWi\_Fi with i=[1,4] and j=[1,4]  and send a TC\_LFR\_DUMP\_PAR  send a TC\_LFR\_LOAD\_FBINS\_MASK  and send a TC\_LFR\_DUMP\_PAR  send a TC\_LFR\_LOAD\_FILTER\_PAR inconsistent Modulus vs shift + offset)  and send a TC\_LFR\_DUMP\_PAR | | | | Verify that parameters contained in the TM\_LFR PARAMETER\_DUMP packets have not been updated to bad ones and not forced to default  TC\_LFR\_LOAD\_COMMON\_PAR hasn’t bad parameter value.  TC\_LFR\_LOAD\_FBINS\_MASK hasn’t bad parameter value. | | | | |  |  |
|  | End | | | |  | | | | |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE** | | | | | | | | | | | |
| **Test procedure identifier** | | | | | | **Date** | | | **Assessment** | | |
| SVS-0066\_Ed1 | | | | | | DD/MM/YYYY | | | NT / NA / NOK / POK / OK | | |
| **TS Tested requirements** | | | | | | **RB Tested requirements** | | | | | |
| N/A | | | | | | REQ-LFR-SRS-5552\_Ed1 | | | | | |
| **Component:** | | | |  | | **Version:** | V2 | | | | |
| **Involved subsystems** | | | | | | | | | | | |
| SW: | | 3.2.0.24 | | | | | | | | | |
| HW: | | 1.1.91 StarDundee | | | | | | | | | |
| Start time: | | | (DD/MM/YYYY) hh:mm | | | Test Operator: | |  | | | |
| End Time: | | | (DD/MM/YYYY) hh:mm | | | Test Duration: | | 13s | | | |
| **Purpose:** | | | Check, upon reception of a TC\_LFR\_RESET command, the LFR FSW forces the reboot of the LFR: the initial state after a reset or after a boot is then the STANDBY mode. | | | | | | | | |
| **General remarks about the test** | | | | | | | | | | | |
| Requirement Title | | | | Software reset | | | | | | | |
| dependencies | | | | SSS-CP-EQS-220 | | | | | | | |
| restrictions | | | |  | | | | | | | |
| means | | | |  | | | | | | | |
| input data | | | |  | | | | | | | |
| prerequisite | | | |  | | | | | | | |
| used test programs | | | |  | | | | | | | |
| test script | | | | just\_tc\_lfr\_reset.py | | | | | | | |
| output data | | | |  | | | | | | | |
| notes | | | | Tc\_lfr\_reset performs exit(0) | | | | | | | |
| **Procedure steps** | | | | | | | | | | | |
| **Step** | **Actions** | | | | **Expected Results** | | | | | **Comments** | **Status** |
|  | LFR FSW receipts a TC\_LFR\_RESET command. | | | | This TC is not acknowledged because exit (0) is performed.  Environment test displays  A TM\_LFR\_TC\_EXE\_\* is not seen after time-out (4s). | | | | |  |  |
|  | Boot the LFR. | | | | Verify the initial state after a boot is the STANDBY mode. | | | | |  |  |
|  | End | | | |  | | | | |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE** | | | | | | | | | | | | |
| **Test procedure identifier** | | | | | | | **Date** | | | **Assessment** | | |
| SVS-0067\_Ed1 | | | | | | | DD/MM/YYYY | | | NT / NA / NOK / POK / OK | | |
| **TS Tested requirements** | | | | | | | **RB Tested requirements** | | | | | |
| N/A | | | | | | | REQ-LFR-SRS-5517\_Ed1 | | | | | |
| **Component:** | | | | |  | | **Version:** | V2 | | | | |
| **Involved subsystems** | | | | | | | | | | | | |
| SW: | | | 3.2.0.24 | | | | | | | | | |
| HW: | | | 1.1.91 StarDundee | | | | | | | | | |
| Start time: | | | | (DD/MM/YYYY) hh:mm | | | Test Operator: | |  | | | |
| End Time: | | | | (DD/MM/YYYY) hh:mm | | | Test Duration: | | 4mn5 | | | |
| **Purpose:** | Check the LFR FSW uses its internal time to time-stamp all the packets. | | | | | | | | | | | |
| **General remarks about the test** | | | | | | | | | | | | |
| Requirement Title | | | | | Software reset (NOT IMPLEMENTED) | | | | | | | |
| dependencies | | | | | SSS-CP-EQS-400 | | | | | | | |
| restrictions | | | | |  | | | | | | | |
| means | | | | |  | | | | | | | |
| input data | | | | |  | | | | | | | |
| prerequisite | | | | |  | | | | | | | |
| used test programs | | | | |  | | | | | | | |
| test script | | | | | activeLfrinternaltime.py | | | | | | | |
| output data | | | | |  | | | | | | | |
| notes | | | | |  | | | | | | | |
| **Procedure steps** | | | | | | | | | | | | |
| **Step** | | **Actions** | | | | **Expected Results** | | | | | **Comments** | **Status** |
|  | | Begin in SBM2 mode during  6s after boot  Reach each mode  LFR FSW receipts a S/C time by TC\_LFR\_UPDATE\_TIME+ time code  Wait 5s | | | | Verify the LFR FSW time-stamps all the HK packets.  Verify the LFR FSW time-stamps all the science data packets.  MSB time = 1  Verify the LFR FSW time-stamps all the HK packets  Verify the LFR FSW time-stamps all the science data packets.  MSB time = 0 | | | | |  |  |
|  | | Reach each mode  Wait 1.1\*SY\_RPW\_DELAY\_WITHOUT\_CTR. | | | | Verify the LFR FSW time-stamps all the HK packets.  The first TM\_LFR\_HK after the synchronization lost has the fields HK\_LFR\_TIME\_NOT\_SYNCHRO and HK\_LFR\_LE\_CNT incremented by 1.  Verify the LFR FSW time-stamps all the science data packets.  MSB time = 1 lost synchro | | | | |  |  |
|  | | End | | | |  | | | | |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE** | | | | | | | | | | | |
| **Test procedure identifier** | | | | | **Date** | | | | **Assessment** | | |
| SVS-0069\_Ed1 | | | | | DD/MM/YYYY | | | | NT / NA / NOK / POK / OK | | |
| **TS Tested requirements** | | | | | **RB Tested requirements** | | | | | | |
| N/A | | | | | REQ-LFR-SRS-5565\_Ed1 REQ-LFR-SRS-5592\_Ed1 | | | | | | |
| **Component:** | | | |  | **Version:** | | V2 | | | | |
| **Involved subsystems** | | | | | | | | | | | |
| SW: | 3.2.0.24 | | | | | | | | | | |
| HW: | 1.1.91 StarDUndee | | | | | | | | | | |
| Start time: | | | (DD/MM/YYYY) hh:mm | | Test Operator: | | |  | | | |
| End Time: | | | (DD/MM/YYYY) hh:mm | | Test Duration: | | | 3mn10 | | | |
| **Purpose:** | | | Check the LFR FSW executes the mode transition on the next sharp second compared to current LFR time (current coarse time +1) when the time given in parameter of the TC\_LFR\_ENTER\_MODE packet is equal to 0.  The equipment flight software shall ignore the synchronization bit of the CP\_LFR\_ENTER\_MODE\_TIME parameter (most significant bit) when they handle a TC\_LFR\_ENTER\_MODE command. | | | | | | | | |
| **General remarks about the test** | | | | | | | | | | | |
| Requirement Title | | | | Equipment mode management | | | | | | | |
| dependencies | | | | SSS-CP-EQS-321 SSS-CPEQS-328 | | | | | | | |
| restrictions | | | |  | | | | | | | |
| means | | | |  | | | | | | | |
| input data | | | |  | | | | | | | |
| prerequisite | | | |  | | | | | | | |
| used test programs | | | |  | | | | | | | |
| test script | | | | activateLfrMode0.py | | | | | | | |
| output data | | | |  | | | | | | | |
| notes | | | |  | | | | | | | |
| **Procedure steps** | | | | | | | | | | | |
| **Step** | | **Actions** | | | | **Expected Results** | | | | **Comments** | **Status** |
|  | | For each transition modes, send TC with CP\_LFR\_ENTER\_MODE\_TIME =0. | | | | Verify LFR FSW executes the mode transition immediately, without waiting for time synchronization.  HK\_LFR\_MODE of TM\_HK\_LFR is equal to the asked mode.  PA\_LFR\_ACQUISITION\_TIME of TM\_LFR\_SCIENCE\_CWF\_Fx allows to valid this transition mode | | | |  |  |
|  | | For each transition modes, send TC with CP\_LFR\_ENTER\_MODE\_TIME= 0 with msb=1. | | | | Verify LFR SWF executes the mode transition immediately, without waiting for time synchronization.  HK\_LFR\_MODE of TM\_HK\_LFR is equal to the asked mode.  PA\_LFR\_ACQUISITION\_TIME of TM\_LFR\_SCIENCE\_CWF\_Fx allows to valid this transition mode  The MSB bit of the CP\_LFR\_ENTER\_MODE\_TIME  has no-effect level.  LFR FSW executes the mode transition immediately.  HK\_LFR\_MODE of TM\_LFR\_TC\_EXE\_SUCCESS is equal to the asked mode. | | | | REQ-LFR-SRS-5592\_Ed1 |  |
|  | | End | | | |  | | | |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE** | | | | | | | | | | | | | |
| **Test procedure identifier** | | | | | | | **Date** | | | | **Assessment** | | |
| SVS-0070\_Ed1 | | | | | | | DD/MM/YYYY | | | | NT / NA / NOK / POK / OK | | |
| **TS Tested requirements** | | | | | | | **RB Tested requirements** | | | | | | |
| N/A | | | | | | | REQ-LFR-SRS-5566\_Ed1 | | | | | | |
| **Component:** | | | | |  | | **Version:** | V2 | | | | | |
| **Involved subsystems** | | | | | | | | | | | | | |
| SW: | | 3.2.0.24 | | | | | | | | | | | |
| HW: | | 1.1.91 StarDundee | | | | | | | | | | | |
| Start time: | | | (DD/MM/YYYY) hh:mm | | | | Test Operator: | |  | | | | |
| End Time: | | | (DD/MM/YYYY) hh:mm | | | | Test Duration: | | 6mn5 | | | | |
| **Purpose:** | | | | Check the LFR FSW rejects the TC\_LFR\_ENTER\_MODE packet if the CP\_LFR\_ENTER\_MODE\_TIME parameter is lower than the current time. | | | | | | | | | |
| **General remarks about the test** | | | | | | | | | | | | | |
| Requirement Title | | | | | | Equipment mode management | | | | | | | |
| dependencies | | | | | | SSS-CP-EQS-322 | | | | | | | |
| restrictions | | | | | |  | | | | | | | |
| means | | | | | |  | | | | | | | |
| input data | | | | | |  | | | | | | | |
| prerequisite | | | | | |  | | | | | | | |
| used test programs | | | | | |  | | | | | | | |
| test script | | | | | | activateLfrModeRetroactive.py (steps 1,2), | | | | | | | |
| output data | | | | | |  | | | | | | | |
| Notes | | | | | |  | | | | | | | |
| **Procedure steps** | | | | | | | | | | | | | |
| **Step** | **Actions** | | | | | | | | | **Expected Results** | | **Comments** | **Status** |
|  | For each transition modes, send TC with CP\_LFR\_ENTER\_MODE\_TIME-current TIME=-1(fine time forced to 0), -1/0x10000. | | | | | | | | | Verify LFR FSW rejects the TC\_LFR\_ENTER\_MODE packet.  and one  TM\_LFR\_EXE\_NOT\_EXECUTABLE is generated.  Field HK\_LFR\_MODE of TM\_LFR\_EXE\_NOT\_EXECUTABLE showns the current working mode . | |  |  |
|  | For each transition modes, send TC with CP\_LFR\_ENTER\_MODE\_TIME-current TIME=-1(fine time forced to 0), -1/0x10000. with msb=1. | | | | | | | | | Verify LFR FSW rejects the TC\_LFR\_ENTER\_MODE packet.  and one  TM\_LFR\_EXE\_NOT\_EXECUTABLE is generated.  Field HK\_LFR\_MODE of TM\_LFR\_EXE\_NOT\_EXECUTABLE showns the current working mode . | |  |  |
|  | End | | | | | | | | |  | |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE** | | | | | | | | | | | | |
| **Test procedure identifier** | | | | | | **Date** | | | | **Assessment** | | |
| SVS-0071\_Ed1 | | | | | | DD/MM/YYYY | | | | NT / NA / NOK / POK / OK | | |
| **TS Tested requirements** | | | | | | **RB Tested requirements** | | | | | | |
| N/A | | | | | | REQ-LFR-SRS-5567\_Ed1 | | | | | | |
| **Component:** | | | | |  | **Version:** | V2 | | | | | |
| **Involved subsystems** | | | | | | | | | | | | |
| SW: | | 3.2.0.24 | | | | | | | | | | |
| HW: | | 1.1.91 StarDundee | | | | | | | | | | |
| Start time: | | | (DD/MM/YYYY) hh:mm | | | Test Operator: | |  | | | | |
| End Time: | | | (DD/MM/YYYY) hh:mm | | | Test Duration: | | 5mn10 | | | | |
| **Purpose:** | | | Verify the LFR FSW accepts all the transition between modes excepting if the destination mode is equal to the source mode. | | | | | | | | | |
| **General remarks about the test** | | | | | | | | | | | | |
| Requirement Title | | | | Equipment mode management | | | | | | | | |
| dependencies | | | | SSS-CP-EQS-325 | | | | | | | | |
| restrictions | | | |  | | | | | | | | |
| means | | | |  | | | | | | | | |
| input data | | | |  | | | | | | | | |
| prerequisite | | | |  | | | | | | | | |
| used test programs | | | |  | | | | | | | | |
| test script | | | | activateLfrMode.py | | | | | | | | |
| output data | | | |  | | | | | | | | |
| notes | | | |  | | | | | | | | |
| **Procedure steps** | | | | | | | | | | | | |
| **Step** | **Actions** | | | | | | | | **Expected Results** | | **Comments** | **Status** |
|  | Realize all the mode transitions by the complete sequence:  4->0->0->1->0->2->0->3->0->4->1->1->2->1->3->1->4->2->2->3->2->4->3->3->4->4->0.  Notice:  -Sends comply with timing constraints.  -For the four Science sub-modes: don’t send the TC\_LFR\_LOAD\_xxx\_PAR before the TC\_LFR\_ENTER\_MODE. | | | | | | | | Verify the LFR FSW:  -accepts all the transition between modes excepting if the destination mode is equal to the source mode, in this case TM\_LFR\_EXE\_NOT\_EXECUTABLE.  -the HK\_LFR\_MODE field from TM\_LFR\_HK is consistent  -the TM\_LFR\_SCIENCE\_\* are consistent. | |  |  |
|  | End | | | | | | | |  | |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE** | | | | | | | | | | | |
| **Test procedure identifier** | | | | | | **Date** | | | **Assessment** | | |
| SVS-0073\_Ed1 | | | | | | DD/MM/YYYY | | | NT / NA / NOK / POK / OK | | |
| **TS Tested requirements** | | | | | | **RB Tested requirements** | | | | | |
| N/A | | | | | | REQ-LFR-SRS-5575\_Ed1 | | | | | |
| **Component:** | | | |  | | **Version:** | V2 | | | | |
| **Involved subsystems** | | | | | | | | | | | |
| SW: | | 3.2.0.24 | | | | | | | | | |
| HW: | | 1.1.91 StarDundee | | | | | | | | | |
| Start time: | | | (DD/MM/YYYY) hh:mm | | | Test Operator: | |  | | | |
| End Time: | | | (DD/MM/YYYY) hh:mm | | | Test Duration: | | 1h10 | | | |
| **Purpose:** | | | Check the LFR FSW is able to change the value of SY\_LFR\_N\_SWF\_L and SY\_LFR\_N\_SWF\_P parameters upon reception of the TC\_LFR\_LOAD\_NORMAL\_PAR packet. | | | | | | | | |
| **General remarks about the test** | | | | | | | | | | | |
| Requirement Title | | | |  | | | | | | | |
| dependencies | | | | SSS-CP-EQS-431 | | | | | | | |
| restrictions | | | | SY\_LFR\_N\_SWF\_L ne peut etre changer. Reste à sa valeur par défaut 2048 | | | | | | | |
| means | | | |  | | | | | | | |
| input data | | | |  | | | | | | | |
| prerequisite | | | |  | | | | | | | |
| used test programs | | | |  | | | | | | | |
| test script | | | | loadNorm.py | | | | | | | |
| output data | | | |  | | | | | | | |
| notes | | | | Realize the steps below for each initial mode without SCIENCE\_NORMAL stream, and for each final mode with SCIENCE\_NORMAL stream. | | | | | | | |
| **Procedure steps** | | | | | | | | | | | |
| **Step** | **Actions** | | | | **Expected Results** | | | | | **Comments** | **Status** |
|  | Standby mode:  We try to change SY\_LFR\_N\_SWF\_L to value <> 2048 ( we try 1024). | | | | Verify that TM\_LFR\_TC\_EXE\_INCONSISTENT is received with PA\_RPW\_BYTE\_POSITION = 10 and PA\_RPW\_RCV\_VALUE = 0. Also, we have to see by TM\_LFR\_PARAM\_DUMP that SY\_LFR\_N\_SWF\_L is still 2048 | | | | |  |  |
| 2 | For mode NORMAL SBM1 SBM2  Change only SY\_LFR\_N\_SWF\_P (22s) and default value for SY\_LFR\_N\_SWF\_L  Stay in the mode during  1.1\* SY\_LFR\_N\_SWF\_P seconds | | | | Verify LFR has changed the value of SY\_LFR\_N\_SWF\_P parameters: by the TM\_LFR\_PARAM\_DUMP, by the SCIENCE\_NORMAL stream analysis.  We observe one sequence of 7 packets for TM\_LFR\_SCIENCE\_SWF\_F0 around 41s after the transition mode.  Same thing for TM\_LFR\_SCIENCE\_SWF\_F1  Same thing for TM\_LFR\_SCIENCE\_SWF\_F2 | | | | |  |  |
| 3 | Change SY\_LFR\_N\_SWF\_P with default  Value.  default value for SY\_LFR\_N\_SWF\_L) | | | | Verify LFR has changed the value of SY\_LFR\_N\_SWF\_P parameters: by the TM\_LFR\_PARAM\_DUMP, by the SCIENCE\_NORMAL stream analysis. | | | | |  |  |
| 4 | End | | | |  | | | | |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE** | | | | | | | | | | | |
| **Test procedure identifier** | | | | | **Date** | | | | **Assessment** | | |
| SVS-0074\_Ed1 | | | | | DD/MM/YYYY | | | | NT / NA / NOK / POK / OK | | |
| **TS Tested requirements** | | | | | **RB Tested requirements** | | | | | | |
| N/A | | | | | REQ-LFR-SRS-5571\_Ed1 | | | | | | |
| **Component:** | | | |  | **Version:** | | V2 | | | | |
| **Involved subsystems** | | | | | | | | | | | |
| SW: | | 3.2.0.24 | | | | | | | | | |
| HW: | | 1.1.91 StarDundee | | | | | | | | | |
| Start time: | | | (DD/MM/YYYY) hh:mm | | Test Operator: | | |  | | | |
| End Time: | | | (DD/MM/YYYY) hh:mm | | Test Duration: | | | 6mn | | | |
| **Purpose:** | | | Check the LFR FSW rejects the TC\_LFR\_ENTER\_MODE packet if the CP\_LFR\_ENTER\_MODE\_TIME parameter is greater than the current time plus 3 seconds. | | | | | | | | |
| **General remarks about the test** | | | | | | | | | | | |
| Requirement Title | | | | Equipment mode management | | | | | | | |
| dependencies | | | | SSS-CP-EQS-323 | | | | | | | |
| restrictions | | | |  | | | | | | | |
| means | | | |  | | | | | | | |
| input data | | | |  | | | | | | | |
| prerequisite | | | |  | | | | | | | |
| used test programs | | | |  | | | | | | | |
| test script | | | | activateLfrModeOver.py | | | | | | | |
| output data | | | |  | | | | | | | |
| notes | | | | CP\_LFR\_ENTER\_MODE\_TIME = Due date in seconds (fine\_time= 0) | | | | | | | |
| **Procedure steps** | | | | | | | | | | | |
| **Step** | **Actions** | | | | | **Expected Results** | | | | **Comments** | **Status** |
|  | For each transition modes, send TC with CP\_LFR\_ENTER\_MODE\_TIME=current TIME+3s. | | | | | This TC is validated by a  TM\_LFR\_TC\_EXE\_SUCCESS.  HK\_LFR\_MODE of TM\_LFR\_TC\_EXE\_SUCCESS is equal to the asked mode.  Transition at due date is realized. | | | |  |  |
|  | For each transition modes, send TC with CP\_LFR\_ENTER\_MODE\_TIME= current TIME+4s. | | | | | Verify the LFR FSW rejects the TC\_LFR\_ENTER\_MODE packet.  and one  TM\_LFR\_EXE\_NOT\_EXECUTABLE  is generated.  Field HK\_LFR\_MODE showns the current working mode . | | | |  |  |
|  | For each transition modes, send TC with CP\_LFR\_ENTER\_MODE\_TIME=current TIME+3s, and . with msb=1. | | | | | This TC is validated by a  TM\_LFR\_TC\_EXE\_SUCCESS.  HK\_LFR\_MODE of TM\_LFR\_TC\_EXE\_SUCCESS is equal to the asked mode.  Transition at due date is realized.  MSB bit of TIME has no-effect level. | | | |  |  |
|  | For each transition modes, send TC with CP\_LFR\_ENTER\_MODE\_TIME= current TIME+4s and . with msb=1. | | | | | Verify the LFR FSW rejects the TC\_LFR\_ENTER\_MODE packet.  and one  TM\_LFR\_EXE\_NOT\_EXECUTABLE  is generated.  Field HK\_LFR\_MODE showns the current working mode .  MSB bit of TIME has no-effect level. | | | |  |  |
|  | End | | | | |  | | | |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE** | | | | | | | | | | | | |
| **Test procedure identifier** | | | | | | | **Date** | | | **Assessment** | | |
| SVS-0076\_Ed1 | | | | | | | DD/MM/YYYY | | | NT / NA / NOK / POK / OK | | |
| **TS Tested requirements** | | | | | | | **RB Tested requirements** | | | | | |
| N/A | | | | | | | REQ-LFR-SRS-5222\_Ed1 | | | | | |
| **Component:** | | | | |  | | **Version:** | V2 | | | | |
| **Involved subsystems** | | | | | | | | | | | | |
| SW: | | 3.2.0.24 | | | | | | | | | | |
| HW: | | StarDundee | | | | | | | | | | |
| Start time: | | | (DD/MM/YYYY) hh:mm | | | | Test Operator: | |  | | | |
| End Time: | | | (DD/MM/YYYY) hh:mm | | | | Test Duration: | | 5mn40 | | | |
| **Purpose:** | | | Check if no SpW time code is received by the LFR FSW for a delay greater than SY\_RPW\_DELAY\_WITHOUT\_CTR, the LFR FSW indicates this by setting the MSB of the time field of each packet to 1. | | | | | | | | | |
| **General remarks about the test** | | | | | | | | | | | | |
| Requirement Title | | | | Time management | | | | | | | | |
| dependencies | | | | SSS-CP-FS-410; AD06 R-789 | | | | | | | | |
| restrictions | | | |  | | | | | | | | |
| means | | | |  | | | | | | | | |
| input data | | | |  | | | | | | | | |
| prerequisite | | | |  | | | | | | | | |
| used test programs | | | |  | | | | | | | | |
| test script | | | | time\_msb1.py | | | | | | | | |
| output data | | | |  | | | | | | | | |
| notes | | | |  | | | | | | | | |
| **Procedure steps** | | | | | | | | | | | | |
| **Step** | **Actions** | | | | | **Expected Results** | | | | | **Comments** | **Status** |
| 1 | Launch LFR FSW,  after the boot sequence,  no timecode and no TC\_LFR\_UPDATE\_TIME | | | | | in TM\_LFR\_HK  Only one timing error is traced,  HK\_LFR\_ TIMECODE\_MISSING at this end of boot sequence  So  HK\_LFR\_LE\_CNT=1,  HK\_LFR\_LAST\_ER\_RID: LE\_LFR\_TIMEC = 42129, HK\_LFR\_LAST\_ER\_CODE: MISSING = 21, HK\_LFR\_LAST\_ER\_TIME=0x800000022f12,  HK\_LFR\_DPU\_SPW\_TICK\_OUT\_CNT=0, HK\_LFR\_DPU\_SPW\_LAST\_TIMC=0, HK\_LFR\_TIMECODE\_ERRONEOUS=0, HK\_LFR\_TIMECODE\_MISSING=1, HK\_LFR\_TIMECODE\_INVALID=0, HK\_LFR\_TIME\_TIMECODE\_IT=0, HK\_LFR\_TIME\_NOT\_SYNCHRO=0, HK\_LFR\_TIME\_TIMECODE\_CTR=0  TIME of TM\_LFR\_HK has its MSB bit set to 1 | | | | |  |  |
| 2 | Realize, for each mode:  LFR is synchronized at t0  (TC\_LFR\_UPDATE\_TIME + valid time-code)  LFR receives neither time codes, neither CTR for a delay greater than SY\_RPW\_DELAY\_WITHOUT\_CTR. | | | | | Verify TIME field of all TMs:  -coarse and fine part are consistent with the clock of the PC.  -MSB is 0 before t0+SY\_RPW\_DELAY\_WITHOUT\_CTR.  HK\_LFR\_DPU\_SPW\_TICK\_OUT\_CNT and HK\_LFR\_DPU\_SPW\_LAST\_TIMC are updated  but  -MSB is 1 after t0+SY\_RPW\_DELAY\_WITHOUT\_CTR.  in TM\_LFR\_HK  HK\_LFR\_LE\_CNT is incremented of 1  HK\_LFR\_TIME\_NOT\_SYNCHRO=1 | | | | |  |  |
| 3 | Go to standby | | | | | in TM\_HK\_LFR  5 valid timecodes sent  HK\_LFR\_DPU\_SPW\_TICK\_OUT\_CNT=5, HK\_LFR\_DPU\_SPW\_LAST\_TIMC=5,  HK\_LFR\_TIME\_NOT\_SYNCHRO=5  HK\_LFR\_TIMECODE\_MISSING=6 (one after the boot sequence + 5)  the last error is  HK\_LFR\_LAST\_ER\_RID: LE\_LFR\_TIME = 42119, HK\_LFR\_LAST\_ER\_CODE: NOT\_SYNCHRO = 25, HK\_LFR\_LAST\_ER\_TIME=0x8000003d64f3 | | | | |  |  |
| 3 | End | | | | |  | | | | |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE** | | | | | | | | | | | |
| **Test procedure identifier** | | | | | | **Date** | | | **Assessment** | | |
| SVS-0077\_Ed1 | | | | | | DD/MM/YYYY | | | NT / NA / NOK / POK / OK | | |
| **TS Tested requirements** | | | | | | **RB Tested requirements** | | | | | |
| N/A | | | | | | REQ-LFR-SRS-5574\_Ed1 REQ-LFR-SRS-5564\_Ed1 | | | | | |
| **Component:** | | | |  | | **Version:** | V2 | | | | |
| **Involved subsystems** | | | | | | | | | | | |
| SW: | | 3.2.0.24 | | | | | | | | | |
| HW: | | 1.1.91 StarDundee | | | | | | | | | |
| Start time: | | | (DD/MM/YYYY) hh:mm | | | Test Operator: | |  | | | |
| End Time: | | | (DD/MM/YYYY) hh:mm | | | Test Duration: | | 6mn | | | |
| **Purpose:** | | | Check upon reception of a TC\_LFR\_UPDATE\_INFO packet, the LFR FSW increments the HK\_LFR\_UPDATE\_INFO\_TC\_CNT counter only if the packet is correct and has been accepted.  Check the LFR FSW does not acknowledge the TC\_LFR\_UPDATE\_INFO packets | | | | | | | | |
| **General remarks about the test** | | | | | | | | | | | |
| Requirement Title | | | | Equipment command feedback | | | | | | | |
| dependencies | | | | SSS-CP-EQS-141 SSS-CP-EQS-354 | | | | | | | |
| restrictions | | | |  | | | | | | | |
| means | | | |  | | | | | | | |
| input data | | | |  | | | | | | | |
| prerequisite | | | |  | | | | | | | |
| used test programs | | | |  | | | | | | | |
| test script | | | | update\_info\_cnt.py + + update\_time\_and\_info\_cnt\_wrap.py (step3) | | | | | | | |
| output data | | | |  | | | | | | | |
| notes | | | | Fully tested by REQ-LFR-SRS-5543\_Ed1 (SVS-0081) | | | | | | | |
| **Procedure steps** | | | | | | | | | | | |
| **Step** | **Actions** | | | | **Expected Results** | | | | | **Comments** | **Status** |
| 1 | After boot, all rw frequencies are set to Nan  Send a TC\_LFR\_DUMP\_PAR and  wait TM\_LFR\_PARAMETER\_DUMP | | | | All rw frequencies are set to Nan  (all PA\_LFR\_RW\_MASK\_Fi\_WORDj are set to  0xFFFFFFFF. I=[1,4] and j=[1,4]  On TM\_LFR\_HK, HK\_LFR\_SC\_RWi\_Fj are DISABLED (=0). I=[1,4] and j=[1,4] | | | | |  |  |
| 2 | In each mode  send the sequence  Send nominal TC\_LFR\_UPDATE\_INFO | | | | On TM\_LFR\_HK  Verify the LFR FSW increments the HK\_LFR\_UPDATE\_INFO\_TC\_CNT counter. | | | | |  |  |
| 3 | Send erroneous TC\_LFR\_UPDATE\_INFO  (inconsistent parameters)  CP\_LFR\_MODE\_COPY = [5,15]  CP\_TDS\_MODE\_COPY =[6,15]  CP\_THR\_MODE\_COPY =[[4,15] | | | | On TM\_LFR\_HK  Verify the LFR FSW does not increment the HK\_LFR\_UPDATE\_INFO\_TC\_CNT counter.  HK\_LFR\_LAST\_REJ\_x Fields of the TM\_LFR\_HK not updated. | | | | |  |  |
|  | Send erroneous TC\_LFR\_UPDATE\_INFO  (inconsistent parameters)  a negative rw frequency  CP\_RPW\_SC\_RW1\_F1 = -1.0  send a TC\_LFR\_DUMP\_PAR | | | | Verify the LFR FSW does not increment the HK\_LFR\_UPDATE\_INFO\_TC\_CNT counter.  HK\_LFR\_LAST\_REJ\_x Fields of the TM\_LFR\_HK not updated.  On TM\_LFR\_PARAMETER\_DUMP  All rw frequencies are set to Nan  (all PA\_LFR\_RW\_MASK\_Fi\_WORDj are set to  0xFFFFFFFF. I=[1,4] and j=[1,4]  On TM\_LFR\_HK TM\_LFR\_HK, HK\_LFR\_SC\_RWi\_Fj are DISABLED (=0). I=[1,4] and j=[1,4] | | | | |  |  |
| 4 | Send erroneous TC\_LFR\_UPDATE\_INFO (for instance: wrong .CRC)). | | | | Verify the LFR FSW does not increment the HK\_LFR\_UPDATE\_INFO\_TC\_CNT counter.  HK\_LFR\_LAST\_REJ\_x Fields of the TM\_LFR\_HK not updated. | | | | | Accurate tests in SVS-00081 |  |
| 5 | **Use update\_time\_and\_info\_cnt\_wrap.py**  Reach the HK\_LFR\_UPDATE\_INFO\_TC\_CNT maximal value.  Send nominal TC\_LFR\_UPDATE\_INFO | | | | Verify the LFR FSW increments the HK\_LFR\_UPDATE\_INFO\_TC\_CNT counter by wrapping around 0 (after 65535). | | | | | See SVS-0078  (step3) |  |
| 6 | End | | | |  | | | | |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE** | | | | | | | | | | | |
| **Test procedure identifier** | | | | | **Date** | | | | **Assessment** | | |
| SVS-0078\_Ed1 | | | | | DD/MM/YYYY | | | | NT / NA / NOK / POK / OK | | |
| **TS Tested requirements** | | | | | **RB Tested requirements** | | | | | | |
| N/A | | | | | REQ-LFR-SRS-5572\_Ed1 REQ-LFR-SRS-5241\_Ed1 | | | | | | |
| **Component:** | | | |  | **Version:** | | V2 | | | | |
| **Involved subsystems** | | | | | | | | | | | |
| SW: | | 3.2.0.24 | | | | | | | | | |
| HW: | | 1.1.91 Stardundee | | | | | | | | | |
| Start time: | | | (DD/MM/YYYY) hh:mm | | Test Operator: | | |  | | | |
| End Time: | | | (DD/MM/YYYY) hh:mm | | Test Duration: | | | 50s + 3h50 | | | |
| **Purpose:** | | | Check upon reception of a TC\_LFR\_UPDATE\_TIME packet, the LFR FSW increment s the HK\_LFR\_UPDATE\_TIME\_TC\_CNT counter only if the packet is correct and has been accepted. | | | | | | | | |
| **General remarks about the test** | | | | | | | | | | | |
| Requirement Title | | | | Equipment command feedback | | | | | | | |
| dependencies | | | | SSS-CP-EQS-142 SSS-CP-FS-201 | | | | | | | |
| restrictions | | | |  | | | | | | | |
| means | | | |  | | | | | | | |
| input data | | | |  | | | | | | | |
| prerequisite | | | |  | | | | | | | |
| used test programs | | | |  | | | | | | | |
| test script | | | | update\_time\_cnt.py + update\_time\_and\_info\_cnt\_wrap.py (step3) | | | | | | | |
| output data | | | |  | | | | | | | |
| notes | | | |  | | | | | | | |
| **Procedure steps** | | | | | | | | | | | |
| **Step** | **Actions** | | | | | **Expected Results** | | | | **Comments** | **Status** |
| 1 | In each mode:  Send nominal TC\_LFR\_UPDATE\_TIME without time code. | | | | | Verify the LFR FSW increments the HK\_LFR\_UPDATE\_TIME\_TC\_CNT counter. | | | |  |  |
| 2 | In each mode:  Send erroneous TC\_LFR\_UPDATE\_TIME (for instance: wrong .CRC), with nominal time codes. | | | | | Verify the LFR FSW does not increment the HK\_LFR\_UPDATE\_TIME\_TC\_CNT counter.  HK\_LFR\_LAST\_REJ\_x Fields of the TM\_LFR\_HK not updated. | | | |  |  |
| 3 | **Use update\_time\_and\_info\_cnt\_wrap.py**  Reach the HK\_LFR\_UPDATE\_TIME\_TC\_CNT maximal value  Send nominal TC\_LFR\_UPDATE\_TIME  ( no time code)  Reach the HK\_LFR\_UPDATE\_INFO\_TC\_CNT maximal value | | | | | Verify the LFR FSW increments the HK\_LFR\_UPDATE\_TIME\_TC\_CNT counter by wrapping around 0,  and  HK\_LFR\_UPDATE\_INFO\_TC\_CNT counter by wrapping around 0 | | | | 0x10004 TC ~3h50  REQ-LFR-SRS-5241\_Ed1 |  |
| 4 | End | | | | |  | | | |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE** | | | | | | | | | | | |
| **Test procedure identifier** | | | | | **Date** | | | | **Assessment** | | |
| SVS-0079 \_Ed1 | | | | | DD/MM/YYYY | | | | NT / NA / NOK / POK / OK | | |
| **TS Tested requirements** | | | | | **RB Tested requirements** | | | | | | |
| N/A | | | | | REQ-LFR-SRS-5568\_Ed1 | | | | | | |
| **Component:** | | | |  | **Version:** | | V2 | | | | |
| **Involved subsystems** | | | | | | | | | | | |
| SW: | | 3.2.0.24 | | | | | | | | | |
| HW: | | 1.1.91 StarDundee | | | | | | | | | |
| Start time: | | | (DD/MM/YYYY) hh:mm | | Test Operator: | | |  | | | |
| End Time: | | | (DD/MM/YYYY) hh:mm | | Test Duration: | | | 49mn | | | |
| **Purpose:** | | | Check, when it performs a NORMAL / BURST mode transition, the LFR FSW stops the acquisition processing which is in progress if any. | | | | | | | | |
| **General remarks about the test** | | | | | | | | | | | |
| Requirement Title | | | | LFR NORMAL / BURST mode transition | | | | | | | |
| dependencies | | | | SSS-CP-EQS-525 | | | | | | | |
| restrictions | | | |  | | | | | | | |
| means | | | |  | | | | | | | |
| input data | | | |  | | | | | | | |
| prerequisite | | | |  | | | | | | | |
| used test programs | | | |  | | | | | | | |
| test script | | | | Normal2Burst \_swf\_22s\_asm\_4s.py + extract.py | | | | | | | |
| output data | | | |  | | | | | | | |
| notes | | | |  | | | | | | | |
| **Procedure steps** | | | | | | | | | | | |
| **Step** | **Actions** | | | | | **Expected Results** | | | | **Comments** | **Status** |
| 1 | Realize the below scenario:  t1: TC\_LFR\_ENTER\_MODE immediat to NORMAL. Ensure LFR processes acquisitions.  (Wait SWF or CWF products)  t2: TC\_LFR\_ENTER\_MODE immediat to BURST as short as possible.  t3: TC\_LFR\_ENTER\_MODE immediat to NORMAL. Ensure LFR processes acquisitions. | | | | | Using the TIME\_ACQUISITION of TMs emetted after t3, verify the LFR FSW has stopped the previous acquisition processing | | | |  |  |
| 3 | End | | | | |  | | | |  |  |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE** | | | | | | | | | | |
| **Test procedure identifier** | | | | | | | **Date** | **Assessment** | | |
| SVS-0081\_Ed1 | | | | | | | DD/MM/YYYY | NT / NA / NOK / POK / OK | | |
| **TS Tested requirements** | | | | | | | **RB Tested requirements** | | | |
| N/A | | | | | | | REQ-LFR-SRS-5543\_Ed3 REQ-LFR-SRS-5564\_Ed3 | | | |
| **Component:** | | | | |  | | **Version:** V2 |  | | |
| **Involved subsystems** | | | | | | | | | | |
| SW: | | | | | 3.2.0.24 | | | | | |
| HW: | | | | | 1.1.91 StarDundee | | | | | |
| Start time: | | | | | (DD/MM/YYYY) hh:mm | | Test Operator: |  | | |
| End Time: | | | | | (DD/MM/YYYY) hh:mm | | Test Duration: 35s |  | | |
| **Purpose:** | | The equipment flight software shall acknowledge all the commands it receives from the DPU expected the both following commands for which no TM\_LFR\_TC\_EXE\_yyy packets shall be generated**:** TC\_LFR\_UPDATE\_INFO, TC\_LFR\_UPDATE\_TIME.  - The parameters HK\_LFR\_EXE\_TC\_CNT to HK\_LFR\_LAST\_REJ\_TC\_TIME shall not be updated upon reception of TC\_LFR\_UPDATE\_INFO and TC\_LFR\_UPDATE\_TIME. | | | | | | | | |
| **General remarks about the test** | | | | | | | | | | |
| Requirement Title | | | | Command Feedback | | | | | | |
| dependencies | | | | AD05, SSS-CP-EQS-140 SSS-CP-EQS-354 | | | | | | |
| restrictions | | | |  | | | | | | |
| means | | | |  | | | | | | |
| input data | | | |  | | | | | | |
| prerequisite | | | | Acceptance of the command fails. | | | | | | |
| used test programs | | | |  | | | | | | |
| test script | | | | loop\_update.py | | | | | | |
| output data | | | |  | | | | | | |
| notes | | | -criterias of acceptance failed are listed in REQ-LFR-SRS-5204  -CRC should not add extra inconsistency; so, it should be recalculated | | | | | | | |
| **Procedure steps** | | | | | | | | | | |
| **Step** | **Actions** | | | | | **Expected Results** | | | **Comments** | **Status** |
| 1 | For each mode send a  TC\_LFR\_UPDATE\_INFO nominal. | | | | | Verify:  -FSW does not reply TM\_LFR\_TC\_EXE\_xxx.  -The parameters HK\_LFR\_EXE\_TC\_CNT to HK\_ LFR \_LAST\_REJ\_TC\_TIME are not updated.  HK\_LFR\_UPDATE\_INFO\_TC\_CNT is incremented | | |  |  |
| 2 | TC\_LFR\_UPDATE\_INFO with error detected during the acceptance stage verifications.  ( bad CRC, bad PID, bad CAT, bad type, bad SUB-TYPE, bad length packet, bad source\_id , bad length vs type/sub-type). | | | | | Verify:  -FSW does not reply TM\_LFR\_TC\_EXE\_xxx.  -The parameters HK\_LFR\_EXE\_TC\_CNT to HK\_ LFR \_LAST\_REJ\_TC\_TIME are not updated.  Except when type et sub-type are bad, LFR FSW doen’t identify a TC\_UPDATE\_INFO and in this case, LFR FSW sends a TM\_LFR\_TC\_EXE\_CORRUPTED. | | |  |  |
| 3 | Send a TC\_LFR\_UPDATE\_TIME nominal | | | | | Verify:  -FSW does not reply TM\_LFR\_TC\_EXE\_xxx.  -The parameters HK\_LFR\_EXE\_TC\_CNT to HK\_ LFR \_LAST\_REJ\_TC\_TIME are not updated.  HK\_LFR\_UPDATE\_TIME\_TC\_CNT is incremented | | |  |  |
| 4 | TC\_LFR\_UPDATE\_TIME with error detected during the acceptance stage verifications  ( bad CRC, bad PID, bad CAT, bad type, bad SUB-TYPE, bad length packet, bad source\_id , bad length vs type/sub-type). | | | | | Verify:  -FSW does not reply TM\_LFR\_TC\_EXE\_xxx.  -The parameters HK\_LFR\_EXE\_TC\_CNT to HK\_ LFR \_LAST\_REJ\_TC\_TIME are not updated.  Except when type et sub-type are bad, LFR FSW doen’t identify a TC\_UPDATE\_TIME and in this case, LFR FSW sends a TM\_LFR\_TC\_EXE\_CORRUPTED. | | |  |  |
| 5 | End | | | | |  | | |  |  |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE** | | | | | | | | | | |
| **Test procedure identifier** | | | | | | | **Date** | **Assessment** | | |
| SVS-0082\_Ed1 | | | | | | | DD/MM/YYYY | NT / NA / NOK / POK / OK | | |
| **TS Tested requirements** | | | | | | | **RB Tested requirements** | | | |
| N/A | | | | | | | REQ-LFR-SRS-5580\_Ed1 REQ-LFR-SRS-5569\_Ed1 | | | |
| **Component:** | | | | |  | | **Version:** V3 |  | | |
| **Involved subsystems** | | | | | | | | | | |
| SW: | | | | | 3.2.0.24 | | | | | |
| HW: | | | | | 1.1.91 StarDundee | | | | | |
| Start time: | | | | | (DD/MM/YYYY) hh:mm | | Test Operator: |  | | |
| End Time: | | | | | (DD/MM/YYYY) hh:mm | | Test Duration: 16s |  | | |
| **Purpose:** | | The LFR flight software shall report in its periodic HK packet (TM\_LFR\_HK) the availability of the electric field components sampled at f3.  VHDL starts acquisition immediately after power-on and hardware init sequence. After flight software boot sequence and as soon as the filters are nominally working (stable) : electrical values written by VHDL in dedicated registers are read by LFR Flight software and following actions are performed:   * HK\_LFR\_SC\_POTENTIAL\_FLAG is set to 1 * HK\_LFR\_SC\_V\_F3, HK\_LFR\_SC\_E1\_F3, HK\_LFR\_SC\_E2\_F3 are filled   This implies that even in STANDBY mode, flag is set to 1 and fields are filled with valid values.  Check each second, the LFR FSW puts in its TM\_LFR\_HK packet a mean of the 16 last values of the electric field components sampled at f3: V\_f3, E1\_f3, E2\_f3. | | | | | | | | |
| **General remarks about the test** | | | | | | | | | | |
| Requirement Title | | | | LFR data for S/C potential computation | | | | | | |
| dependencies | | | | SSS-CP-EQS-533 | | | | | | |
| restrictions | | | |  | | | | | | |
| means | | | |  | | | | | | |
| input data | | | |  | | | | | | |
| prerequisite | | | |  | | | | | | |
| used test programs | | | |  | | | | | | |
| test script | | | | survey\_potential\_computation.py + CTC-800 | | | | | | |
| output data | | | |  | | | | | | |
| notes | | | Never observed HK\_LFR\_SC\_POTENTIAL\_FLAG = 0 because, this case is during some cycles in the boot sequence.  Some plots of results can be seen in Calibration report  (RPW-MEB-LFR-RPT-00168-LPP-01-02-LFR\_Calibration\_report) | | | | | | | |
| **Procedure steps** | | | | | | | | | | |
| **Step** | **Actions** | | | | | **Expected Results** | | | **Comments** | **Status** |
| 1 | Launch LFR FSW  For each mode  Read the 3 VHDL registers F3\_V,F3\_E1,F3\_E2  wait 10s TM\_LFR\_HK packets. | | | | | After the boot sequence, LFR is in STANDBY,  Verify the first one TM\_LFR\_HK  HK\_LFR\_SC\_POTENTIAL\_FLAG is set to 1  HK\_LFR\_SC\_V\_F3,HK\_LFR\_SC\_E1\_F3, HK\_LFR\_SC\_E2\_F3 are filled with the registers values .  Registers addresses are 0x80000fe4,0x 80000fe8, 0x80000fec  Verify each second, the LFR FSW puts in its TM\_LFR\_HK packet a mean of the 16 last values of the electric field components sampled at f3:  V\_f3, E1\_f3,E2\_f2. | | |  |  |
| 2 | End | | | | |  | | |  |  |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE** | | | | | | | | | | |
| **Test procedure identifier** | | | | | | | **Date** | **Assessment** | | |
| SVS-0083\_Ed1 | | | | | | | DD/MM/YYYY | NT / NA / NOK / POK / OK | | |
| **TS Tested requirements** | | | | | | | **RB Tested requirements** | | | |
| N/A | | | | | | | REQ-LFR-SRS-5581\_Ed1 REQ-LFR-SRS-5582\_Ed1 | | | |
| **Component:** | | | | |  | | **Version:** V3 |  | | |
| **Involved subsystems** | | | | | | | | | | |
| SW: | | | | | 3.2.0.24 | | | | | |
| HW: | | | | | 1.1.91 StarDundee | | | | | |
| Start time: | | | | | (DD/MM/YYYY) hh:mm | | Test Operator: |  | | |
| End Time: | | | | | (DD/MM/YYYY) hh:mm | | Test Duration: 35s |  | | |
| **Purpose:** | | Upon reception of a TC\_LFR\_LOAD\_FBINS\_MASK, the LFR flight software shall be able to update the internal mask used for avoiding some frequency bins in the computation of the basic parameters:   * 1 mask f0 , 128 bits = 16 bytes * 1 mask f1 , 16 bytes * 1 mask f2 , 16 bytes   The LFR flight software shall be able to dump in the TM\_LFR\_PARAMETER\_DUMP packet the internal mask of frequency bins. | | | | | | | | |
| **General remarks about the test** | | | | | | | | | | |
| Requirement Title | | | | LFR frequency bins internal mask | | | | | | |
| dependencies | | | | SSS-CP-EQS-527 SSS-CP-EQS-528 | | | | | | |
| restrictions | | | |  | | | | | | |
| means | | | |  | | | | | | |
| input data | | | |  | | | | | | |
| prerequisite | | | |  | | | | | | |
| used test programs | | | |  | | | | | | |
| test script | | | | load\_fbins\_with\_dump.py | | | | | | |
| output data | | | |  | | | | | | |
| notes | | | Covered by Calbration tests CTC-610 to CTC-612 | | | | | | | |
| **Procedure steps** | | | | | | | | | | |
| **Step** | **Actions** | | | | | **Expected Results** | | | **Comments** | **Status** |
| 1 | Launch LFR FSW  For each mode  Send a TC\_LFR\_LOAD\_FBINS\_MASK with default values  Send a TC\_LFR\_DUMP\_PAR | | | | | TM\_LFR\_PARAMETER\_DUMP contains  FBINS\_Fi\_WORD1=0xffffffff  FBINS\_Fi\_WORD2=0xffffffff  FBINS\_Fi\_WORD3=0xffffffff  FBINS\_Fi\_WORD4=0xffffffff  Fi = F0,F1,F2  The default values are set to 1 | | | No mode constraint |  |
| 2 | For each mode  Send a TC\_LFR\_LOAD\_FBINS\_MASK with all bits set to 0  Send a TC\_LFR\_DUMP\_PAR | | | | | TM\_LFR\_PARAMETER\_DUMP contains  FBINS\_Fi\_WORD1=0x  FBINS\_Fi\_WORD2=0Xffffffff  FBINS\_Fi\_WORD3=0Xffffffff  FBINS\_Fi\_WORD4=0Xffffffff  Fi = F0,F1,F2 | | | No mode constraint |  |
| 3 | Use 3 calibration tests CTC-610,CTC-611 and CTC-612 to validate the effect of the internal mask on science data (BP2 products).  For each frequency  internal mask = 0  then  Internal mask= 1 | | | | |  | | |  |  |
| 4 | acceptance stage tests see  SVS-0003, SVS-0005, SVS-0007  SVS-0008, SVS-0009 | | | | | no inconsistent parameter for this TC | | |  |  |
| 5 | End | | | | |  | | |  |  |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE** | | | | | | | | | | |
| **Test procedure identifier** | | | | | | **Date** | | **Assessment** | | |
| SVS-0086\_Ed1 | | | | | | DD/MM/YYYY | | NT / NA / NOK / POK / OK | | |
| **TS Tested requirements** | | | | | | **RB Tested requirements** | | | | |
| N/A | | | | | | REQ-LFR-SRS-5583\_Ed1 REQ-LFR-SRS-5584\_Ed1 | | | | |
| **Component:** | | | | |  | **Version:** V3 | |  | | |
| **Involved subsystems** | | | | | | | | | | |
| SW: | | | | | 3.2.0.24 | | | | | |
| HW: | | | | | 1.1.91 StarDundee | | | | | |
| Start time: | | | | | (DD/MM/YYYY) hh:mm | Test Operator: | |  | | |
| End Time: | | | | | (DD/MM/YYYY) hh:mm | Test Duration: 20mn | |  | | |
| **Purpose:** | | Upon reception of several TC\_LFR\_LOAD\_KCOEFFICIENTS packets, the LFR flight software shall be able to update the inter-calibration factors (k-coefficients) used for computing the poynting flux and phase velocity estimators:   * 32 coefficients (coded on 4 bytes) for 36 frequencies.   Upon reception of a TC\_LFR\_DUMP\_KCOEFFICIENTS packet, the LFR flight software shall be able to dump in two TM\_LFR\_KCOEFFICENTS\_DUMP packets the inter-calibration factors (k-coefficients). | | | | | | | | |
| **General remarks about the test** | | | | | | | | | | |
| Requirement Title | | | | LFR inter-calibration factors | | | | | | |
| dependencies | | | | SSS-CP-EQS-529 SSS-CP-EQS-531 | | | | | | |
| restrictions | | | |  | | | | | | |
| means | | | |  | | | | | | |
| input data | | | |  | | | | | | |
| prerequisite | | | |  | | | | | | |
| used test programs | | | |  | | | | | | |
| test script | | | | load\_kcoeff\_with\_dump.py | | | | | | |
| output data | | | |  | | | | | | |
| notes | | | Covered by Calibration tests CTC-700 | | | | | | | |
| **Procedure steps** | | | | | | | | | | |
| **Step** | **Actions** | | | | | | **Expected Results** | | **Comments** | **Status** |
| 1 | Launch LFR FSW  Send a TC\_LFR\_DUMP\_KCOEFFICIENTS | | | | | | 2 TM\_LFR\_KCOEFFICIENTS\_DUMP expected  First TM\_LFR\_KCOEFFICIENTS\_DUMP contains  from SY\_LFR\_KCOEFF\_FREQUENCY = 0  to SY\_LFR\_KCOEFF\_FREQUENCY = 29  for each one, 32 KCOEFFICIENTS with default values = 1  field SY\_LFR\_KCOEFF\_i (1 to 32)  Second TM\_LFR\_KCOEFFICIENTS\_DUMP contains  from SY\_LFR\_KCOEFF\_FREQUENCY =30  to SY\_LFR\_KCOEFF\_FREQUENCY = 35  for each one, 32 KCOEFFICIENTS with default values = 1  field SY\_LFR\_KCOEFF\_i (1 to 32) | | No mode constraint |  |
| 2 | For each mode  Send a TC\_LFR\_LOAD\_KCOEFFICIENTS with default values  KCOEFF\_FREQ = 0  Send a TC\_LFR\_DUMP\_KCOEFFICIENTS | | | | | | 2 TM\_LFR\_KCOEFFICIENTS\_DUMP expected  First TM\_LFR\_KCOEFFICIENTS\_DUMP contains  from SY\_LFR\_KCOEFF\_FREQUENCY = 0  to SY\_LFR\_KCOEFF\_FREQUENCY = 29  for each one, 32 KCOEFFICIENTS with default values = 1  field SY\_LFR\_KCOEFF\_i (1 to 32)  Second TM\_LFR\_KCOEFFICIENTS\_DUMP contains  from SY\_LFR\_KCOEFF\_FREQUENCY =30  to SY\_LFR\_KCOEFF\_FREQUENCY = 35  for each one, 32 KCOEFFICIENTS with default values = 1  field SY\_LFR\_KCOEFF\_i (1 to 32) | | No mode constraint |  |
| 3 | For each mode  for each freq [1,35]  Send a TC\_LFR\_LOAD\_KCOEFFICIENTS with 19 Kcoeffs changed with representative values.  Send a  TC\_LFR\_DUMP\_KCOEFFICIENTS | | | | | | 2 TM\_LFR\_KCOEFFICIENTS\_PAR , only the 19 Kcoefficients of the tested frequency are changed.  At this test end, 19 kcoefficients of all frequencies are changed. | | No mode constraint |  |
| 4 | Use one calibration test CTC-710 to validate the effect inter-calibration factors on science products.  For each frequency  32 factors changed | | | | | |  | |  |  |
| 5 | to acceptance stage see  SVS-0003, SVS-0005, SVS-0007  SVS-0008, SVS-0009 | | | | | | KCOEFF\_FREQ is the only parameter of this TC.  Interval definition is [0,35]. | |  |  |
| 6 | End | | | | | |  | |  |  |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE** | | | | | | | | | | |
| **Test procedure identifier** | | | | | | **Date** | | **Assessment** | | |
| SVS-0088\_Ed1 | | | | | | DD/MM/YYYY | | NT / NA / NOK / POK / OK | | |
| **TS Tested requirements** | | | | | | **RB Tested requirements** | | | | |
| N/A | | | | | | REQ-LFR-SRS-5590\_Ed1 | | | | |
| **Component:** | | | | |  | **Version:** V2 | |  | | |
| **Involved subsystems** | | | | | | | | | | |
| SW: | | | | | 3.2.0.24 | | | | | |
| HW: | | | | | 1.1.91 StarDundee | | | | | |
| Start time: | | | | | (DD/MM/YYYY) hh:mm | Test Operator: | |  | | |
| End Time: | | | | | (DD/MM/YYYY) hh:mm | Test Duration: 1h30 | |  | | |
| **Purpose:** | | |  | | --- | | Upon reception of a TC\_LFR\_ENTER\_MODE(SBM1) / TC\_LFR\_ENTER\_MODE(SBM2) / TC\_LFR\_ENTER\_MODE(NORMAL), the equipment flight software shall **not** re-initialize the NORMAL data flow if this one was already active. | |  | | | | | | | | | |
| **General remarks about the test** | | | | | | | | | | |
| Requirement Title | | | | Equipment mode management | | | | | | |
| dependencies | | | | SSS-CP-EQS-326 | | | | | | |
| Restrictions | | | | For this test SWF period is set to 22s and ASM period to 4s | | | | | | |
| means | | | |  | | | | | | |
| input data | | | |  | | | | | | |
| prerequisite | | | |  | | | | | | |
| used test programs | | | | LFR\_packet\_decom-linux-x86\_64 (C++ LPP decom program) | | | | | | |
| test script | | | | test\_modes-noTimegen\_allTransitions.py | | | | | | |
| output data | | | | All TM are logged in log\_data file and All dataflows are in record\_data file (can be decommuted with LFR\_packet\_decom-linux-x86\_64) | | | | | | |
| notes | | |
| **Procedure steps** | | | | | | | | | | |
| **Step** | **Actions** | | | | | | **Expected Results** | | **Comments** | **Status** |
| 1 | LFR goes from STANDBY mode to NORMAL during 600s | | | | | | We should see all products of NORMAL dataflow starting with expected periodicity. | |  |  |
| 2 | LFR goes from NORMAL mode to SBM1 during 900s | | | | | | We should see that waveform products of NORMAL dataflow periodicities are unchanged since step 1 e.g. not affected by mode transition. | |  |  |
| 3 | LFR goes from SBM1 mode to SBM2 during 900s | | | | | | We should see that waveform products of NORMAL dataflow periodicities are unchanged since step 1 e.g. not affected by mode transition. | |  |  |
| 4 | LFR goes from SBM2 mode to NORMAL during 600s | | | | | | We should see that waveform products of NORMAL dataflow periodicities are unchanged since step 1 e.g. not affected by mode transition. | |  |  |
| 5 | LFR goes from NORMAL mode to BURST during 600s | | | | | | We should observe the termination of NORMAL dataflow started at step 1 e.g. no more NORMAL products. | |  |  |
| 6 | LFR goes from BURST mode to NORMAL during 600s | | | | | | We should see all products of NORMAL dataflow starting with expected periodicity e.g. not the continuation of NORMAL dataflow stopped at step 5. | |  |  |
| 7 | LFR goes from NORMAL mode to SBM2 during 900s | | | | | | We should see that waveform products of NORMAL dataflow periodicities are unchanged e.g. not affected by mode transition. | |  |  |
| 8 | LFR goes from SBM2 mode to STANDBY | | | | | | We should observe termination of all dataflows. | |  |  |
|  | End | | | | | |  | |  |  |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE** | | | | | | | | | | |
| **Test procedure identifier** | | | | | | **Date** | | **Assessment** | | |
| SVS-0089\_Ed1 | | | | | | DD/MM/YYYY | | NT / NA / NOK / POK / OK | | |
| **TS Tested requirements** | | | | | | **RB Tested requirements** | | | | |
| N/A | | | | | | REQ-LFR-SRS-5591\_Ed1 REQ-LFR-SRS-5501\_Ed1 | | | | |
| **Component:** | | | | |  | **Version:** V3 | |  | | |
| **Involved subsystems** | | | | | | | | | | |
| SW: | | | | | 3.2.0.24 | | | | | |
| HW: | | | | | 1.1.91 StarDundee | | | | | |
| Start time: | | | | | (DD/MM/YYYY) hh:mm | Test Operator: | |  | | |
| End Time: | | | | | (DD/MM/YYYY) hh:mm | Test Duration: 50mn | |  | | |
| **Purpose:** | | |  | | --- | | The LFR FSW shall stop to produce science packets as soon as they have sent the acknowledgment packet related to the mode transition to STANDBY.  When the LFR FSW enters in the standby mode, it shall be stop the acquisitions if they are active | |  | | | | | | | | | |
| **General remarks about the test** | | | | | | | | | | |
| Requirement Title | | | | Equipment mode management | | | | | | |
| dependencies | | | | SSS-CP-EQS-327 / SSS-CP-EQS-240 | | | | | | |
| restrictions | | | |  | | | | | | |
| means | | | |  | | | | | | |
| input data | | | |  | | | | | | |
| prerequisite | | | |  | | | | | | |
| used test programs | | | |  | | | | | | |
| test script | | | | transition\_to\_standby.py | | | | | | |
| output data | | | |  | | | | | | |
| notes | | | Use ASM=4s for quick test | | | | | | | |
| **Procedure steps** | | | | | | | | | | |
| **Step** | **Actions** | | | | | | **Expected Results** | | **Comments** | **Status** |
| 1 | Launch LFR FSW, mode is standby.  Send a TC\_LOAD\_NORMAL\_PAR  with SY\_N\_ASM\_P = 4s  Reach Normal mode  Send a TC\_LFR\_DUMP\_PAR  Wait t\_n\_aux  Reach Standby and wait t\_n\_aux  t\_n\_aux=360.s | | | | | | After transition NORMAL -> STANDBY  17:21:21.719005, /!\Timeout=360.0s: ['TM\_LFR\_SCIENCE\_NORMAL\_SWF\_F0', 'TM\_LFR\_SCIENCE\_NORMAL\_SWF\_F1', 'TM\_LFR\_SCIENCE\_NORMAL\_SWF\_F2', 'TM\_LFR\_SCIENCE\_NORMAL\_CWF\_F3' 'TM\_LFR\_SCIENCE\_NORMAL\_ASM\_F0', 'TM\_LFR\_SCIENCE\_NORMAL\_ASM\_F1', 'TM\_LFR\_SCIENCE\_NORMAL\_ASM\_F2', 'TM\_LFR\_SCIENCE\_NORMAL\_BP1\_F0', 'TM\_LFR\_SCIENCE\_NORMAL\_BP1\_F1', 'TM\_LFR\_SCIENCE\_NORMAL\_BP1\_F2', 'TM\_LFR\_SCIENCE\_NORMAL\_BP2\_F0', 'TM\_LFR\_SCIENCE\_NORMAL\_BP2\_F1', 'TM\_LFR\_SCIENCE\_NORMAL\_BP2\_F2'] not seen | |  |  |
| 2 | Reach BURST mode  Send a TC\_LFR\_DUMP\_PAR  Wait t\_b\_aux  Reach Standby and wait t\_b\_aux  t\_b\_aux=12.6s | | | | | | After transition BURST -> STANDBY  17:21:54.839855, /!\Timeout=12.6s: ['TM\_LFR\_SCIENCE\_BURST\_CWF\_F2', 'TM\_LFR\_SCIENCE\_BURST\_BP1\_F0', 'TM\_LFR\_SCIENCE\_BURST\_BP2\_F0', 'TM\_LFR\_SCIENCE\_BURST\_BP1\_F1', 'TM\_LFR\_SCIENCE\_BURST\_BP2\_F1'] not seen. | |  |  |
| 3 | Reach SBM1 mode  Send a TC\_LFR\_DUMP\_PAR  Wait t\_1\_aux  Reach Standby and wait t\_1\_aux  t\_1\_aux=360s | | | | | | After transition SBM1 -> STANDBY  17:38:01.793427, /!\Timeout=360.0s: ['TM\_LFR\_SCIENCE\_NORMAL\_SWF\_F0', 'TM\_LFR\_SCIENCE\_NORMAL\_SWF\_F1', 'TM\_LFR\_SCIENCE\_NORMAL\_SWF\_F2', 'TM\_LFR\_SCIENCE\_NORMAL\_CWF\_F3', 'TM\_LFR\_SCIENCE\_NORMAL\_ASM\_F0', 'TM\_LFR\_SCIENCE\_NORMAL\_ASM\_F1', 'TM\_LFR\_SCIENCE\_NORMAL\_ASM\_F2', 'TM\_LFR\_SCIENCE\_NORMAL\_BP1\_F0', 'TM\_LFR\_SCIENCE\_NORMAL\_BP1\_F1', 'TM\_LFR\_SCIENCE\_NORMAL\_BP1\_F2', 'TM\_LFR\_SCIENCE\_NORMAL\_BP2\_F0', 'TM\_LFR\_SCIENCE\_NORMAL\_BP2\_F1', 'TM\_LFR\_SCIENCE\_NORMAL\_BP2\_F2', 'TM\_LFR\_SCIENCE\_SBM1\_CWF\_F1', 'TM\_LFR\_SCIENCE\_SBM1\_BP1\_F0', 'TM\_LFR\_SCIENCE\_SBM1\_BP2\_F0'] not seen. | |  |  |
| 4 | Reach SBM2 mode  Send a TC\_LFR\_DUMP\_PAR  Wait t\_2\_aux  Reach Standby and wait t\_2\_aux  t\_2\_aux=360s | | | | | | After transition SBM2 -> STANDBY  17:54:21.179829, /!\Timeout=360.0s: ['TM\_LFR\_SCIENCE\_NORMAL\_SWF\_F0', 'TM\_LFR\_SCIENCE\_NORMAL\_SWF\_F1', 'TM\_LFR\_SCIENCE\_NORMAL\_SWF\_F2', 'TM\_LFR\_SCIENCE\_NORMAL\_CWF\_F3', 'TM\_LFR\_SCIENCE\_NORMAL\_ASM\_F0', 'TM\_LFR\_SCIENCE\_NORMAL\_ASM\_F1', 'TM\_LFR\_SCIENCE\_NORMAL\_ASM\_F2', 'TM\_LFR\_SCIENCE\_NORMAL\_BP1\_F0', 'TM\_LFR\_SCIENCE\_NORMAL\_BP1\_F1', 'TM\_LFR\_SCIENCE\_NORMAL\_BP1\_F2', 'TM\_LFR\_SCIENCE\_NORMAL\_BP2\_F0', 'TM\_LFR\_SCIENCE\_NORMAL\_BP2\_F1', 'TM\_LFR\_SCIENCE\_NORMAL\_BP2\_F2', 'TM\_LFR\_SCIENCE\_SBM2\_CWF\_F2', 'TM\_LFR\_SCIENCE\_SBM2\_BP1\_F0', 'TM\_LFR\_SCIENCE\_SBM2\_BP2\_F0', 'TM\_LFR\_SCIENCE\_SBM2\_BP1\_F1', 'TM\_LFR\_SCIENCE\_SBM2\_BP2\_F1'] not seen. | |  |  |
| 5 | End | | | | | |  | |  |  |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE** | | | | | | | | | | |
| **Test procedure identifier** | | | | | | | **Date** | **Assessment** | | |
| SVS-0090\_Ed1 | | | | | | | DD/MM/YYYY | NT / NA / NOK / POK / OK | | |
| **TS Tested requirements** | | | | | | | **RB Tested requirements** | | | |
| N/A | | | | | | | REQ-LFR-SRS-5518\_Ed2 REQ-LFR-SRS-5407\_Ed2 | | | |
| **Component:** | | | | |  | | **Version:** V2 V1 |  | | |
| **Involved subsystems** | | | | | | | | | | |
| SW: | | | | | 3.2.0.24 | | | | | |
| HW: | | | | | 1.1.91 StarDundee | | | | | |
| Start time: | | | | | (DD/MM/YYYY) hh:mm | | Test Operator: |  | | |
| End Time: | | | | | (DD/MM/YYYY) hh:mm | | Test Duration: 2h5mn |  | | |
| **Purpose:** | | Each data packet generated by the equipment flight software shall contain the absolute time (SCET) of the first sample contained in the packet (acquisition time). The time of the other samples are deduced from the time of the first sample.   The absolute time value shall be copied twice: 1) in the time field of the TM packet data field header (PUS header) and 2) in the TM packet source data auxiliary header (data fields).   If needed, some relative timestamps can be added in the data packet to tag data blocks. | | | | | | | | |
| **General remarks about the test** | | | | | | | | | | |
| Requirement Title | | | | Science data acquisition and processing | | | | | | |
| dependencies | | | | SSS-CP-EQS-410 SSS-IF-DPS-EQ-160 | | | | | | |
| restrictions | | | |  | | | | | | |
| means | | | |  | | | | | | |
| input data | | | |  | | | | | | |
| prerequisite | | | |  | | | | | | |
| used test programs | | | |  | | | | | | |
| test script | | | | generate\_science\_all\_modes.py + extract\_v1.bash | | | | | | |
| output data | | | |  | | | | | | |
| notes | | |  | | | | | | | |
| **Procedure steps** | | | | | | | | | | |
| **Step** | **Actions** | | | | | **Expected Results** | | | **Comments** | **Status** |
| **1** | We generated all  TM\_LFR\_SCIENCE\_\* in all science modes.  MSB=1 | | | | | Verify:  Absolute time is copied with the same value in time field of TM data auxiliary header AND in PUS header (PA\_LFR\_ACQUISITION\_TIME).  No relative timestamps are added. | | | extract\_v1.bash does the verification |  |
| **2** | End | | | | |  | | |  |  |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE** | | | | | | | | | | |
| **Test procedure identifier** | | | | | | | **Date** | **Assessment** | | |
| SVS-0091\_Ed1 | | | | | | | DD/MM/YYYY | NT / NA / NOK / POK / OK | | |
| **TS Tested requirements** | | | | | | | **RB Tested requirements** | | | |
| N/A | | | | | | | REQ-LFR-SRS-5289\_Ed1 | | | |
| **Component:** | | | | |  | | **Version:** V3 |  | | |
| **Involved subsystems** | | | | | | | | | | |
| SW: | | | | | 3.2.0.24 | | | | | |
| HW: | | | | | 1.1.91 StarDundee | | | | | |
| Start time: | | | | | (DD/MM/YYYY) hh:mm | | Test Operator: |  | | |
| End Time: | | | | | (DD/MM/YYYY) hh:mm | | Test Duration: 10s |  | | |
| **Purpose:** | | The RPW Flight Software shall explicitly configure the data and instruction caches at startup. | | | | | | | | |
| **General remarks about the test** | | | | | | | | | | |
| Requirement Title | | | | Cache configuration | | | | | | |
| dependencies | | | | SSS-CP-FS-610 | | | | | | |
| restrictions | | | |  | | | | | | |
| means | | | |  | | | | | | |
| input data | | | |  | | | | | | |
| prerequisite | | | |  | | | | | | |
| used test programs | | | |  | | | | | | |
| test script | | | | test\_cache\_state.py | | | | | | |
| output data | | | |  | | | | | | |
| notes | | | Reset the LFR | | | | | | | |
| **Procedure steps** | | | | | | | | | | |
| **Step** | **Actions** | | | | | **Expected Results** | | | **Comments** | **Status** |
|  | Run socexplorer  don’t launch LFR SFW  Read the instruction cache status into the dedicated register  Read the data cache status  into the dedicated register | | | | | instruction cache status = Disabled  data cache status = Disabled | | |  |  |
|  | Load and run the LFR FSW  Read the instruction cache status into the dedicated register  Read the data cache status  into the dedicated register | | | | | instruction cache status = Enabled  data cache status = Enabled | | |  |  |
|  | End | | | | |  | | |  |  |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE** | | | | | | | | | | |
| **Test procedure identifier** | | | | | | | **Date** | **Assessment** | | |
| SVS-0095\_Ed1 | | | | | | | DD/MM/YYYY | NT / NA / NOK / POK / OK | | |
| **TS Tested requirements** | | | | | | | **RB Tested requirements** | | | |
| N/A | | | | | | | REQ-LFR-SRS-5550\_Ed1 | | | |
| **Component:** | | | | |  | | **Version:** V2 |  | | |
| **Involved subsystems** | | | | | | | | | | |
| SW: | | | | | 3.2.0.24 | | | | | |
| HW: | | | | | 1.1.91 StarDundee | | | | | |
| Start time: | | | | | (DD/MM/YYYY) hh:mm | | Test Operator: |  | | |
| End Time: | | | | | (DD/MM/YYYY) hh:mm | | Test Duration: 7h10 |  | | |
| **Purpose:** | | Additionally, the equipment flight software, depending on the features of the equipment, shall have one or several other commands for configuring the parameters not directly linked to the science sub-mode configuration.   Common parameters: TC\_LFR\_LOAD\_COMMON\_PAR | | | | | | | | |
| **General remarks about the test** | | | | | | | | | | |
| Requirement Title | | | | Equipment configuration management | | | | | | |
| dependencies | | | | SSS-CP-EQS-210 | | | | | | |
| restrictions | | | |  | | | | | | |
| means | | | |  | | | | | | |
| input data | | | |  | | | | | | |
| prerequisite | | | |  | | | | | | |
| used test programs | | | |  | | | | | | |
| test script | | | | send\_tc\_load\_common\_all\_modes.py | | | | | | |
| output data | | | |  | | | | | | |
| notes | | |  | | | | | | | |
| **Procedure steps** | | | | | | | | | | |
| **Step** | **Actions** | | | | | **Expected Results** | | | **Comments** | **Status** |
| 1 | Launch LFR FSW  Standby mode  configure normal mode  Send TC\_LFR\_NORMAL\_PAR  SY\_LFR\_N\_SWF\_P=22(s)  SY\_LFR\_N\_ASM\_P=4 (s)  Send a TC\_LFR\_DUMP\_PAR  DEFAULT\_COMMON\_PAR = 0x20  SY\_LFR\_BW = 1  SY\_LFR\_SP0 , SY\_LFR\_SP1, SY\_LFR\_R0, SY\_LFR\_R1, SY\_LFR\_R2 = 0  For each parameter combination  send a TC\_LFR\_COMMON\_PAR  send a TC\_DUMP\_PAR  wait 10s  send TC\_LFR\_COMMON\_PAR with DEFAULT\_COMMON\_PAR = 0x20  send a TC\_DUMP\_PAR | | | | | TM\_LFR\_DUMP\_PAR contains the modified parameter into  the dedicated field: SY\_LFR\_BW, SY\_LFR\_SP0 ,  SY\_LFR\_SP1, SY\_LFR\_R0, SY\_LFR\_R1, SY\_LFR\_R2  TM\_LFR\_HK contains the modified parameter into  the dedicated field: SY\_LFR\_BW, SY\_LFR\_SP0 ,  SY\_LFR\_SP1, SY\_LFR\_R0, SY\_LFR\_R1, SY\_LFR\_R2  No TM\_LFR\_SCIENCE\* products | | | No mode constraint |  |
| 2 | Reach Normal mode  DEFAULT\_COMMON\_PAR = 0x20  SY\_LFR\_BW = 1  SY\_LFR\_SP0 , SY\_LFR\_SP1, SY\_LFR\_R0, SY\_LFR\_R1, SY\_LFR\_R2 = 0  For each parameter combination  send a TC\_LFR\_COMMON\_PAR  send a TC\_DUMP\_PAR  wait 180s  send TC\_LFR\_COMMON\_PAR with DEFAULT\_COMMON\_PAR = 0x20  send a TC\_DUMP\_PAR | | | | | TM\_LFR\_DUMP\_PAR contains the modified parameter into  the dedicated field: SY\_LFR\_BW, SY\_LFR\_SP0 ,  SY\_LFR\_SP1, SY\_LFR\_R0, SY\_LFR\_R1, SY\_LFR\_R2  TM\_LFR\_HK contains the modified parameter into  the dedicated field: SY\_LFR\_BW, SY\_LFR\_SP0 ,  SY\_LFR\_SP1, SY\_LFR\_R0, SY\_LFR\_R1, SY\_LFR\_R2  TM\_LFR\_SCIENCE\_NORMAL\* products contains  the modified parameter into  the dedicated field: SY\_LFR\_BW, SY\_LFR\_SP0 ,  SY\_LFR\_SP1, SY\_LFR\_R0, SY\_LFR\_R1, SY\_LFR\_R2 | | | No mode constraint |  |
| 3 | Reach Burst mode  DEFAULT\_COMMON\_PAR = 0x20  SY\_LFR\_BW = 1  SY\_LFR\_SP0 , SY\_LFR\_SP1, SY\_LFR\_R0, SY\_LFR\_R1, SY\_LFR\_R2 = 0  For each parameter combination  send a TC\_LFR\_COMMON\_PAR  send a TC\_DUMP\_PAR  wait 30s  send TC\_LFR\_COMMON\_PAR with DEFAULT\_COMMON\_PAR = 0x20  send a TC\_DUMP\_PAR | | | | | TM\_LFR\_DUMP\_PAR contains the modified parameter into  the dedicated field: SY\_LFR\_BW, SY\_LFR\_SP0 ,  SY\_LFR\_SP1, SY\_LFR\_R0, SY\_LFR\_R1, SY\_LFR\_R2  TM\_LFR\_HK contains the modified parameter into  the dedicated field: SY\_LFR\_BW, SY\_LFR\_SP0 ,  SY\_LFR\_SP1, SY\_LFR\_R0, SY\_LFR\_R1, SY\_LFR\_R2  TM\_LFR\_SCIENCE\_BURST\* products contains  the modified parameter into  the dedicated field: SY\_LFR\_BW, SY\_LFR\_SP0 ,  SY\_LFR\_SP1, SY\_LFR\_R0, SY\_LFR\_R1, SY\_LFR\_R2  de | | | No mode constraint |  |
| 4 | Reach SBM1 mode  DEFAULT\_COMMON\_PAR = 0x20  SY\_LFR\_BW = 1  SY\_LFR\_SP0 , SY\_LFR\_SP1, SY\_LFR\_R0, SY\_LFR\_R1, SY\_LFR\_R2 = 0  For each parameter combination  send a TC\_LFR\_COMMON\_PAR  send a TC\_DUMP\_PAR  wait 180s  send TC\_LFR\_COMMON\_PAR with DEFAULT\_COMMON\_PAR = 0x20  send a TC\_DUMP\_PAR | | | | | TM\_LFR\_DUMP\_PAR contains the modified parameter into  the dedicated field: SY\_LFR\_BW, SY\_LFR\_SP0 ,  SY\_LFR\_SP1, SY\_LFR\_R0, SY\_LFR\_R1, SY\_LFR\_R2  TM\_LFR\_HK contains the modified parameter into  the dedicated field: SY\_LFR\_BW, SY\_LFR\_SP0 ,  SY\_LFR\_SP1, SY\_LFR\_R0, SY\_LFR\_R1, SY\_LFR\_R2  TM\_LFR\_SCIENCE\_NORMAL\* products contains  the modified parameter into  the dedicated field: SY\_LFR\_BW, SY\_LFR\_SP0 ,  SY\_LFR\_SP1, SY\_LFR\_R0, SY\_LFR\_R1, SY\_LFR\_R2  TM\_LFR\_SCIENCE\_SBM1\* products contains  the modified parameter into  the dedicated field: SY\_LFR\_BW, SY\_LFR\_SP0 ,  SY\_LFR\_SP1, SY\_LFR\_R0, SY\_LFR\_R1, SY\_LFR\_R2 | | | No mode constraint |  |
| 5 | Reach SBM2 mode  DEFAULT\_COMMON\_PAR = 0x20  SY\_LFR\_BW = 1  SY\_LFR\_SP0 , SY\_LFR\_SP1, SY\_LFR\_R0, SY\_LFR\_R1, SY\_LFR\_R2 = 0  For each parameter combination  send a TC\_LFR\_COMMON\_PAR  send a TC\_DUMP\_PAR  wait 180s  send TC\_LFR\_COMMON\_PAR with DEFAULT\_COMMON\_PAR = 0x20  send a TC\_DUMP\_PAR | | | | | TM\_LFR\_DUMP\_PAR contains the modified parameter into  the dedicated field: SY\_LFR\_BW, SY\_LFR\_SP0 ,  SY\_LFR\_SP1, SY\_LFR\_R0, SY\_LFR\_R1, SY\_LFR\_R2  TM\_LFR\_HK contains the modified parameter into  the dedicated field: SY\_LFR\_BW, SY\_LFR\_SP0 ,  SY\_LFR\_SP1, SY\_LFR\_R0, SY\_LFR\_R1, SY\_LFR\_R2  TM\_LFR\_SCIENCE\_NORMAL\* products contains  the modified parameter into  the dedicated field: SY\_LFR\_BW, SY\_LFR\_SP0 ,  SY\_LFR\_SP1, SY\_LFR\_R0, SY\_LFR\_R1, SY\_LFR\_R2  TM\_LFR\_SCIENCE\_SBM2\* products contains  the modified parameter into  the dedicated field: SY\_LFR\_BW, SY\_LFR\_SP0 ,  SY\_LFR\_SP1, SY\_LFR\_R0, SY\_LFR\_R1, SY\_LFR\_R2 | | | No mode constraint |  |
| 6 | to acceptance stage see  SVS-0003, SVS-0005, SVS-0007  SVS-0008, SVS-0009 | | | | | No inconsistent parameter because each one is coded in one bit.  No constraint mode | | |  |  |
| 7 | Calibration test CTC-100 to validate the effect of these 5 parameters. | | | | |  | | |  |  |
| 8 | End | | | | |  | | |  |  |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE** | | | | | | | | | | |
| **Test procedure identifier** | | | | | | | **Date** | **Assessment** | | |
| SVS-0096\_Ed1 | | | | | | | DD/MM/YYYY | NT / NA / NOK / POK / OK | | |
| **TS Tested requirements** | | | | | | | **RB Tested requirements** | | | |
| N/A | | | | | | | REQ-LFR-SRS-6003\_Ed1 REQ-LFR-SRS-6005\_Ed1  REQ-LFR-SRS-6104\_Ed1 | | | |
| **Component:** | | | | |  | | **Version:** V3++ |  | | |
| **Involved subsystems** | | | | | | | | | | |
| SW: | | | | | 3.2.0.24 | | | | | |
| HW: | | | | | 1.1.91 StarDundee | | | | | |
| Start time: | | | | | (DD/MM/YYYY) hh:mm | | Test Operator: |  | | |
| End Time: | | | | | (DD/MM/YYYY) hh:mm | | Test Duration: 1mn 50 |  | | |
| **Purpose:** | | Upon reception of a TC\_LFR\_LOAD\_FILTER\_PAR packet, the LFR FSW should be able to change   * SY\_LFR\_RW\_DELTA\_F * The 32 bit float values (SY\_LFR\_RWi\_Kj with i=[1,4] and j=[1,4]   The LFR FSW should be able to dump in the TM\_LFR\_PARAMETER\_DUMP   * SY\_LFR\_RW\_DELTA\_F * The 32 bit float values (SY\_LFR\_RWi\_Kj with i=[1,4] and j=[1,4]   The LFR FSW should be able to dump in the TM\_LFR\_PARAMETER\_DUMP   * SY\_LFR\_PAS\_FILTER\_ENABLED * SY\_LFR\_PAS\_FILTER\_MODULUS * SY\_LFR\_PAS\_FILTER\_TBAD * SY\_LFR\_PAS\_FILTER\_OFFSET * SY\_LFR\_PAS\_FILTER\_SHIFT | | | | | | | | |
| **General remarks about the test** | | | | | | | | | | |
| Requirement Title | | | | LFR Filtering of S/C reaction wheel emission frequencies | | | | | | |
| dependencies | | | | SSS-CP-EQS-753 – SSS-CP-EQS-755 – SSS-CP-EQS-756 | | | | | | |
| restrictions | | | |  | | | | | | |
| means | | | |  | | | | | | |
| input data | | | |  | | | | | | |
| prerequisite | | | |  | | | | | | |
| used test programs | | | |  | | | | | | |
| test script | | | | change\_param\_load\_filter\_par.py | | | | | | |
| output data | | | |  | | | | | | |
| notes | | |  | | | | | | | |
| **Procedure steps** | | | | | | | | | | |
| **Step** | **Actions** | | | | | **Expected Results** | | | **Comments** | **Status** |
| 1 | Launch LFR FSW  Standby mode  Send a TC\_LFR\_DUMP\_PAR | | | | | TM\_LFR\_DUMP\_PAR contains the default values  SY\_LFR\_PAS\_FILTER\_ENABLED\_D = DISABLED  SY\_LFR\_PAS\_FILTER\_MODULUS = 4  SY\_LFR\_PAS\_FILTER\_TBAD = 1.0  SY\_LFR\_PAS\_FILTER\_OFFSET = 0  SY\_LFR\_PAS\_FILTER\_SHIFT = 0.5  SY\_LFR\_SC\_RW\_DELTA\_F = 0.045  SY\_LFR\_RWi\_K1 = 1 with i=[1,4]  SY\_LFR\_RWi\_K2 = 8 with i=[1,4]  SY\_LFR\_RWi\_K3 = 24 with i=[1,4]  SY\_LFR\_RWi\_K4 = 48 with i=[1,4]  PA\_LFR\_RW\_MASK\_Fi\_WORDj = 1 (DISABLED)  with i=[0,2] and j=[1,4]  either 32 bits by frequency | | | No mode constraint |  |
| 2 | For each mode  send a TC\_LOAD\_FILTER\_PAR  with  SY\_LFR\_PAS\_FILTER\_ENABLED\_D = DISABLED  SY\_LFR\_SC\_RW\_DELTA\_F  = 1.0  SY\_LFR\_RWi\_K1 = 2 with i=[1,4]  SY\_LFR\_RWi\_K2 = 16 with i=[1,4]  SY\_LFR\_RWi\_K3 = 48 with i=[1,4]  SY\_LFR\_RWi\_K4 = 96 with i=[1,4]  send a TC\_LFR\_DUMP\_PAR  send a TC\_LOAD\_FILTER\_PAR  with  SY\_LFR\_PAS\_FILTER\_ENABLED\_D = ENABLED  and previous values  send a TC\_LFR\_DUMP\_PAR  send a TC\_LOAD\_FILTER\_PAR  with  SY\_LFR\_PAS\_FILTER\_ENABLED\_D =DISABLED  and default values of others parameters  send a TC\_LFR\_DUMP\_PAR | | | | | TM\_LFR\_DUMP\_PAR contains the modified parameter into  the dedicated field:  SY\_LFR\_SC\_RW\_DELTA\_F  SY\_LFR\_RWi\_K1 = 1 with i=[1,4]  SY\_LFR\_RWi\_K2 = 8 with i=[1,4]  SY\_LFR\_RWi\_K3 = 24 with i=[1,4]  SY\_LFR\_RWi\_K4 = 48 with i=[1,4]  These fields are not updated  PA\_LFR\_RW\_MASK\_Fi\_WORDj = 1 (DISABLED)  with i=[0,2] and j=[1,4]  either 32 bits by frequency  because the SY\_LFR\_PAS\_FILTER\_ENABLED\_D = DISABLED | | | No mode constraint |  |
| 3 | For each mode  send a TC\_LOAD\_FILTER\_PAR  with  SY\_LFR\_PAS\_FILTER\_ENABLED\_D = ENABLED  SY\_LFR\_PAS\_FILTER\_MODULUS = 6  SY\_LFR\_PAS\_FILTER\_TBAD = 4.0  SY\_LFR\_PAS\_FILTER\_OFFSET = 4  SY\_LFR\_PAS\_FILTER\_SHIFT = 1.0  send a TC\_DUMP\_PAR  send a TC\_LOAD\_FILTER\_PAR  SY\_LFR\_PAS\_FILTER\_ENABLED\_D = DISABLED  with default values for others parameters  send a TC\_DUMP\_PAR | | | | | TM\_LFR\_DUMP\_PAR contains the default values  SY\_LFR\_PAS\_FILTER\_ENABLED\_D = ENABLED  SY\_LFR\_PAS\_FILTER\_MODULUS = 6  SY\_LFR\_PAS\_FILTER\_TBAD = 4.0  SY\_LFR\_PAS\_FILTER\_OFFSET = 4  SY\_LFR\_PAS\_FILTER\_SHIFT = 1.0  TM\_LFR\_DUMP\_PAR contains the default values  SY\_LFR\_PAS\_FILTER\_ENABLED\_D = ENABLED  SY\_LFR\_PAS\_FILTER\_MODULUS = 4  SY\_LFR\_PAS\_FILTER\_TBAD = 1.0  SY\_LFR\_PAS\_FILTER\_OFFSET = 0  SY\_LFR\_PAS\_FILTER\_SHIFT = 0.5 | | |  |  |
| 4 | End | | | | |  | | |  |  |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE** | | | | | | | | | | |
| **Test procedure identifier** | | | | | | | **Date** | **Assessment** | | |
| SVS-1100\_Ed1 | | | | | | | DD/MM/YYYY | NT / NA / NOK / POK / OK | | |
| **TS Tested requirements** | | | | | | | **RB Tested requirements** | | | |
| N/A | | | | | | | REQ-LFR-SRS-6000\_Ed1 (NORMAL mode part) + REQ-LFR-SRS-6020\_Ed1 | | | |
| **Component:** | | | | |  | | **Version:**V3++ |  | | |
| **Involved subsystems** | | | | | | | | | | |
| SW: | | | | | 3.2.0.24 | | | | | |
| HW: | | | | | 1.1.91 + StarDundee | | | | | |
| Start time: | | | | | (DD/MM/YYYY) hh:mm | | Test Operator: |  | | |
| End Time: | | | | | (DD/MM/YYYY) hh:mm | | Test Duration: N/A |  | | |
| **Purpose:** | | * Verify S/C RW filtering efficiency on Basic Parameters (BP2) computation for NORMAL mode | | | | | | | | |
| **General remarks about the test** | | | | | | | | | | |
| Requirement Title | | | | LFR filtering of S/C reaction wheel emission frequencies | | | | | | |
| dependencies | | | | SSS-CP-EQS-750 | | | | | | |
| restrictions | | | |  | | | | | | |
| means | | | |  | | | | | | |
| input data | | | | SVS\_1100.input | | | | | | |
| prerequisite | | | | This test uses lfrcompliance Python notebooks framework | | | | | | |
| used test programs | | | |  | | | | | | |
| test script | | | | SVS\_1100.py | | | | | | |
| output data | | | | SVS\_1100.output | | | | | | |
| notes | | | Automatic analysis is provided through lfrcompliance framework : SVS\_1100\_analysis.ipynb (Python notebook) | | | | | | | |
| **Procedure steps** | | | | | | | | | | |
| **Step** | **Actions** | | | | | **Expected Results** | | | **Comments** | **Status** |
| 1 | * Boot LFR * Activate output to file of the LFR binary stream * Switch to NORMAL mode | | | | |  | | |  |  |
| 2 | For each of the 88 ASM bins frequencies of F0 ( [1632Hz,9984Hz] with df=96Hz) :   * Disable RW filtering * Start generation of the considered frequency on E1, E2, B1, B2 and B3 (1Vpp) * Wait 60 s (to get 3 F0 BP packets) * Activate RW filtering by setting all RWx\_freqs to the injected frequency value. * Wait 60s (to get 3 F0 BP packets) | | | | | * We should get 6 F0 BP packets for each frequency. * We consider to check the 2nd and the 5th because they reflect a stabilized period of acquisition (not impacted by a recent change of parameters). * We should see in the field AUTO of 2nd BP packet a important count rate (> 8x105)reflecting the DISABLE state of RW filtering. * We should see in the field AUTO of 5th BP packet a low count rate (< 5x104)reflecting the ENABLE state of RW filtering. | | | Most filtered values shoud be around 0. Because of some spreading effect over the masked bandwidth, some higher values should be observed but always below 5x104 |  |
| 3 | For each of the 104 ASM bins frequencies of F1 ( [96Hz,1744Hz] with df=16Hz) :   * Disable RW filtering * Start generation of the considered frequency on E1, E2, B1, B2 and B3 (1Vpp) * Wait 60 s (to get 3 F1 BP packets) * Activate RW filtering by setting all RWx\_freqs to the injected frequency value. * Wait 60s (to get 3 F1 BP packets) | | | | | * We should get 6 F1 BP packets for each frequency. * We consider to check the 2nd and the 5th because they reflect a stabilized period of acquisition (not impacted by a recent change of parameters). * We should see in the field AUTO of 2nd BP packet a important count rate (> 8x105)reflecting the DISABLE state of RW filtering. * We should see in the field AUTO of 5th BP packet a low count rate (< 5x104)reflecting the ENABLE state of RW filtering. | | | Most filtered values shoud be around 0. Because of some spreading effect over the masked bandwidth, some higher values should be observed but always below 5x104 |  |
| 4 | For each of the 96 ASM bins frequencies of F2 ( [7Hz,102Hz] with df=1Hz) :   * Disable RW filtering * Start generation of the considered frequency on E1, E2, B1, B2 and B3 (1Vpp) * Wait 60 s (to get 3 F2 BP packets) * Activate RW filtering by setting all RWx\_freqs to the injected frequency value. * Wait 60s (to get 3 F2 BP packets) | | | | | * We should get 6 F2 BP packets for each frequency. * We consider to check the 2nd and the 5th because they reflect a stabilized period of acquisition (not impacted by a recent change of parameters). * We should see in the field AUTO of 2nd BP packet a important count rate (> 8x105)reflecting the DISABLE state of RW filtering. * We should see in the field AUTO of 5th BP packet a low count rate (< 5x104)reflecting the ENABLE state of RW filtering. | | | Most filtered values shoud be around 0. Because of some spreading effect over the masked bandwidth, some higher values should be observed but always below 5x104 |  |
| 5 | End | | | | |  | | |  |  |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE** | | | | | | | | | | |
| **Test procedure identifier** | | | | | | | **Date** | **Assessment** | | |
| SVS-1110\_Ed1 | | | | | | | DD/MM/YYYY | NT / NA / NOK / POK / OK | | |
| **TS Tested requirements** | | | | | | | **RB Tested requirements** | | | |
| N/A | | | | | | | REQ-LFR-SRS-6000\_Ed1 (SBM1 mode part) + REQ-LFR-SRS-6020\_Ed1 | | | |
| **Component:** | | | | |  | | **Version:**V3++ |  | | |
| **Involved subsystems** | | | | | | | | | | |
| SW: | | | | | 3.2.0.24 | | | | | |
| HW: | | | | | 1.1.91 + StarDundee | | | | | |
| Start time: | | | | | (DD/MM/YYYY) hh:mm | | Test Operator: |  | | |
| End Time: | | | | | (DD/MM/YYYY) hh:mm | | Test Duration: N/A |  | | |
| **Purpose:** | | * Verify S/C RW filtering efficiency on Basic Parameters (BP2) computation for SBM1 mode | | | | | | | | |
| **General remarks about the test** | | | | | | | | | | |
| Requirement Title | | | | LFR filtering of S/C reaction wheel emission frequencies | | | | | | |
| dependencies | | | | SSS-CP-EQS-750 | | | | | | |
| restrictions | | | |  | | | | | | |
| means | | | |  | | | | | | |
| input data | | | | SVS\_1110.input | | | | | | |
| prerequisite | | | | This test uses lfrcompliance Python notebooks framework | | | | | | |
| used test programs | | | |  | | | | | | |
| test script | | | | SVS\_1110.py | | | | | | |
| output data | | | | SVS\_1110.output | | | | | | |
| notes | | | In SBM1 we only have BP2@F0.  Automatic analysis is provided through lfrcompliance framework : SVS\_1110\_analysis.ipynb (Python notebook) | | | | | | | |
| **Procedure steps** | | | | | | | | | | |
| **Step** | **Actions** | | | | | **Expected Results** | | | **Comments** | **Status** |
| 1 | * Boot LFR * Activate output to file of the LFR binary stream * Switch to SBM1 mode | | | | |  | | |  |  |
| 2 | For each of the 88 ASM bins frequencies of F0 ( [1632Hz,9984Hz] with df=96Hz) :   * Disable RW filtering * Start generation of the considered frequency on E1, E2, B1, B2 and B3 (1Vpp) * Wait 3 s (to get 3 F0 BP packets from SBM dataflow) * Activate RW filtering by setting all RWx\_freqs to the injected frequency value. * Wait 3s (to get 3 F0 BP packets from SBM dataflow) | | | | | * We should get 6 F0 BP packets for each frequency. * We consider to check the 2nd and the 5th because they reflect a stabilized period of acquisition (not impacted by a recent change of parameters). * We should see in the field AUTO of 2nd BP packet a important count rate (> 8x105)reflecting the DISABLE state of RW filtering. * We should see in the field AUTO of 5th BP packet a low count rate (< 5x104)reflecting the ENABLE state of RW filtering. | | | Most filtered values shoud be around 0. Because of some spreading effect over the masked bandwidth, some higher values should be observed but always below 5x104 |  |
| 3 | End | | | | |  | | |  |  |
| **TEST CASE** | | | | | | | | | | |
| **Test procedure identifier** | | | | | | | **Date** | **Assessment** | | |
| SVS-1102\_Ed1 | | | | | | | DD/MM/YYYY | NT / NA / NOK / POK / OK | | |
| **TS Tested requirements** | | | | | | | **RB Tested requirements** | | | |
| N/A | | | | | | | REQ-LFR-SRS-6006\_Ed1 | | | |
| **Component:** | | | | |  | | **Version:**V3++ |  | | |
| **Involved subsystems** | | | | | | | | | | |
| SW: | | | | | 3.2.0.24 | | | | | |
| HW: | | | | | 1.1.91 + StarDundee | | | | | |
| Start time: | | | | | (DD/MM/YYYY) hh:mm | | Test Operator: |  | | |
| End Time: | | | | | (DD/MM/YYYY) hh:mm | | Test Duration: N/A |  | | |
| **Purpose:** | | * Verify combination of S/C RW filtering masks AND FBINS\_MASKS efficiency on Basic Parameters (BP2) computation | | | | | | | | |
| **General remarks about the test** | | | | | | | | | | |
| Requirement Title | | | | LFR filtering of S/C reaction wheel emission frequencies | | | | | | |
| dependencies | | | | SSS-CP-EQS-527 and SSS-CP-EQS-750 | | | | | | |
| restrictions | | | |  | | | | | | |
| means | | | |  | | | | | | |
| input data | | | | SVS\_1102.input | | | | | | |
| prerequisite | | | | This test uses lfrcompliance Python notebooks framework | | | | | | |
| used test programs | | | |  | | | | | | |
| test script | | | | SVS\_1102.py | | | | | | |
| output data | | | | SVS\_1102.output | | | | | | |
| notes | | | Only sample case tests (combinations) are performed. | | | | | | | |
| **Procedure steps** | | | | | | | | | | |
| **Step** | **Actions** | | | | | **Expected Results** | | | **Comments** | **Status** |
| 1 | * Boot LFR * Activate output to file of the LFR binary stream * Switch to NORMAL mode | | | | |  | | |  |  |
| 2 | **We test here 2 frequencies that are far one from another:**   * Start generation of a sum of sinus (8Hz and 101Hz) on E1, E2, B1, B2 and B3 (1Vpp) * Activate RW filtering by setting all RWx\_freqs to 8Hz * Load a FBINS\_MASK corresponding to 101Hz * Wait 60 s (to get 3 F0 BP packets) | | | | | * We should get 3 F0 BP packets.We consider to check the 2nd because it reflects a stabilized period of acquisition (not impacted by a recent change of parameters). * We should see in the fields AUTO of 2nd BP packet a low count rate for bins corresponding to 8Hz and 101Hz reflecting the action of both RW and FBINS masks combined. | | | Most filtered values shoud be around 0. Because of some spreading effect over the masked bandwidth, some higher values should be observed but always below 5x104 |  |
| 3 | **We test here the same frequency for RW and FBINS:**   * Start generation of a sinus (8Hz ) on E1, E2, B1, B2 and B3 (1Vpp) * Activate RW filtering by setting all RWx\_freqs to 8Hz * Load a FBINS\_MASK corresponding to 8Hz * Wait 60 s (to get 3 F0 BP packets) | | | | | * We should get 3 F0 BP packets.We consider to check the 2nd because it reflects a stabilized period of acquisition (not impacted by a recent change of parameters). * We should see in the fields AUTO of 2nd BP packet a low count rate for bins corresponding to 8Hz reflecting the action of both RW and FBINS masks combined. | | | Most filtered values shoud be around 0. Because of some spreading effect over the masked bandwidth, some higher values should be observed but always below 5x104 |  |
| 4 | **We test here 2 frequencies that are that are contiguous:**   * Start generation of a sum of sinus (8Hz and 10Hz) on E1, E2, B1, B2 and B3 (1Vpp) * Activate RW filtering by setting all RWx\_freqs to 8Hz * Load a FBINS\_MASK corresponding to 10Hz * Wait 60 s (to get 3 F0 BP packets) | | | | | * We should get 3 F0 BP packets.We consider to check the 2nd because it reflects a stabilized period of acquisition (not impacted by a recent change of parameters). * We should see in the fields AUTO of 2nd BP packet a low count rate for bins corresponding to 8Hz and 10Hz reflecting the action of both RW and FBINS masks combined. | | | Most filtered values shoud be around 0. Because of some spreading effect over the masked bandwidth, some higher values should be observed but always below 5x104 |  |
| 5 | End | | | | |  | | |  |  |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE** | | | | | | | | | | |
| **Test procedure identifier** | | | | | | | **Date** | **Assessment** | | |
| SVS-1103\_Ed1 | | | | | | | DD/MM/YYYY | NT / NA / NOK / POK / OK | | |
| **TS Tested requirements** | | | | | | | **RB Tested requirements** | | | |
| N/A | | | | | | | REQ-LFR-SRS-6001\_Ed1 | | | |
| **Component:** | | | | |  | | **Version:**V3++ |  | | |
| **Involved subsystems** | | | | | | | | | | |
| SW: | | | | | 3.2.0.24 | | | | | |
| HW: | | | | | 1.1.91 + StarDundee | | | | | |
| Start time: | | | | | (DD/MM/YYYY) hh:mm | | Test Operator: |  | | |
| End Time: | | | | | (DD/MM/YYYY) hh:mm | | Test Duration: N/A |  | | |
| **Purpose:** | | * Verify correct reporting in TM\_LFR\_HK of the available/unavailable state for each of the 16 S/C reaction wheel conveyed in TC\_LFR\_UPDATE\_INFO | | | | | | | | |
| **General remarks about the test** | | | | | | | | | | |
| Requirement Title | | | | LFR filtering of S/C reaction wheel emission frequencies | | | | | | |
| dependencies | | | | SSS-CP-EQS-751 | | | | | | |
| restrictions | | | |  | | | | | | |
| means | | | |  | | | | | | |
| input data | | | | SVS\_1103.input | | | | | | |
| prerequisite | | | | This test uses lfrcompliance Python notebooks framework | | | | | | |
| used test programs | | | |  | | | | | | |
| test script | | | | SVS\_1103.py | | | | | | |
| output data | | | | SVS\_1103.output | | | | | | |
| notes | | | Automatic analysis is provided through lfrcompliance framework : SVS\_1103\_analysis.ipynb (Python notebook) | | | | | | | |
| **Procedure steps** | | | | | | | | | | |
| **Step** | **Actions** | | | | | **Expected Results** | | | **Comments** | **Status** |
| 1 | * Boot LFR * Activate output to file of the LFR binary stream * Switch to NORMAL mode | | | | |  | | |  |  |
| 2 | **For each of the 65536 combinations possible for the 16 S/C available state :**   * We load a combination of 16 values (nan or 1.0) through TC\_LFR\_UPDATE\_INFO * We wait 2s (to get an HK packet) | | | | | * We should retrieve in the HK packet received after each TC\_LFR\_UPDATE\_INFO the same values of states that were loaded. | | |  |  |
| 3 | End | | | | |  | | |  |  |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE** | | | | | | | | | | |
| **Test procedure identifier** | | | | | | | **Date** | **Assessment** | | |
| SVS-1109\_Ed1 | | | | | | | DD/MM/YYYY | NT / NA / NOK / POK / OK | | |
| **TS Tested requirements** | | | | | | | **RB Tested requirements** | | | |
| N/A | | | | | | | REQ-LFR-SRS-6004\_Ed1 | | | |
| **Component:** | | | | |  | | **Version:**V3++ |  | | |
| **Involved subsystems** | | | | | | | | | | |
| SW: | | | | | 3.2.0.24 | | | | | |
| HW: | | | | | 1.1.91 + StarDundee | | | | | |
| Start time: | | | | | (DD/MM/YYYY) hh:mm | | Test Operator: |  | | |
| End Time: | | | | | (DD/MM/YYYY) hh:mm | | Test Duration: N/A |  | | |
| **Purpose:** | | * Verify the good reporting of RW masks for F0, F1 and F2 in TM\_LFR\_PARAMETER\_DUMP | | | | | | | | |
| **General remarks about the test** | | | | | | | | | | |
| Requirement Title | | | | LFR filtering of S/C reaction wheel emission frequencies | | | | | | |
| dependencies | | | | SSS-CP-EQS-754 | | | | | | |
| restrictions | | | |  | | | | | | |
| means | | | |  | | | | | | |
| input data | | | | SVS\_1109.input | | | | | | |
| prerequisite | | | | This test uses lfrcompliance Python notebooks framework | | | | | | |
| used test programs | | | |  | | | | | | |
| test script | | | | SVS\_1109.py | | | | | | |
| output data | | | | SVS\_1109.output | | | | | | |
| notes | | | Because it would take too much time to check every possible combination of frequency, only sample test case are implemented : around 1200 frequencies have been carefully chosen to cover all ASM bandwidth and covering the maximum of different intersection cases for “center area” and “polluted bandwidth”.  Automatic analysis is provided through lfrcompliance framework : SVS\_1109\_analysis.ipynb (Python notebook) | | | | | | | |
| **Procedure steps** | | | | | | | | | | |
| **Step** | **Actions** | | | | | **Expected Results** | | | **Comments** | **Status** |
| 1 | * Boot LFR * Activate output to file of the LFR binary stream * Switch to NORMAL mode * Parameter sc\_rw\_delta\_f is set to default value 0.045Hz | | | | |  | | |  |  |
| 2 | * We load through several TC\_LFR\_UPDATE\_INFO frequencies chosen all over F0, F1 and F2 proper frequencies and out of bound frequencies (>10kHz) * After each TC\_LFR\_UPDATE\_INFO,we send a TC\_LFR\_DUMP\_PAR to get a TM\_LFR\_PARAMETER\_DUMP | | | | | We should see F0, F1 and F2 MASKS compliant with frequencies and delta\_f sent through TC\_LFR\_UPDATE\_INFO. | | |  |  |
| 3 | * We load through several TC\_LFR\_UPDATE\_INFO frequencies chosen to be included, intersecting or outside “center area” of ASM bins (for F0, F1 and F2) * After each TC\_LFR\_UPDATE\_INFO,we send a TC\_LFR\_DUMP\_PAR to get a TM\_LFR\_PARAMETER\_DUMP | | | | | We should see F0, F1 and F2 MASKS compliant with frequencies sent through TC\_LFR\_UPDATE\_INFO. | | |  |  |
| 4 | End | | | | |  | | |  |  |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE** | | | | | | | | | | |
| **Test procedure identifier** | | | | | | | **Date** | **Assessment** | | |
| SVS-1201\_Ed1 | | | | | | | DD/MM/YYYY | NT / NA / NOK / POK / OK | | |
| **TS Tested requirements** | | | | | | | **RB Tested requirements** | | | |
| N/A | | | | | | | REQ-LFR-SRS-6101\_Ed1 | | | |
| **Component:** | | | | |  | | **Version:**V3++ |  | | |
| **Involved subsystems** | | | | | | | | | | |
| SW: | | | | | 3.2.0.24 | | | | | |
| HW: | | | | | 1.1.91 + StarDundee | | | | | |
| Start time: | | | | | (DD/MM/YYYY) hh:mm | | Test Operator: |  | | |
| End Time: | | | | | (DD/MM/YYYY) hh:mm | | Test Duration: N/A |  | | |
| **Purpose:** | | * Verify that no filtering process is applied when SY\_LFR\_PAS\_FILTER\_ENABLED is set to DISBALED. | | | | | | | | |
| **General remarks about the test** | | | | | | | | | | |
| Requirement Title | | | | LFR REAL TIME FILTERING OF SWA/PAS PERTURBATIONS | | | | | | |
| dependencies | | | | SSS-CP-EQS-762 | | | | | | |
| restrictions | | | |  | | | | | | |
| means | | | |  | | | | | | |
| input data | | | | We use only noise at LFR inputs. | | | | | | |
| prerequisite | | | | This test uses lfrcompliance Python notebooks framework and LFR SGSE. | | | | | | |
| used test programs | | | | lfrcompliance/test\_engine\_R3pp/ManualTestsToolkit/ManualStarter.py  lfrcompliance/test\_engine\_R3pp/ManualTestsToolkit/ManualLoadFilterPar.py | | | | | | |
| test script | | | |  | | | | | | |
| output data | | | | Validation is made directly by monitoring ASM products on LFR SGSE | | | | | | |
| notes | | |  | | | | | | | |
| **Procedure steps** | | | | | | | | | | |
| **Step** | **Actions** | | | | | **Expected Results** | | | **Comments** | **Status** |
| 1 | * Boot LFR * ASM period is set to 4s * Switch to NORMAL mode | | | | |  | | |  |  |
| 2 | All LFR PAS filtering parameters are default (No TC\_LOAD\_FILTER\_PAR is sent) e.g. PAS filtering is disabled | | | | | We should see noise on all ASM products (F0, F1 and F2 every 4s) | | |  |  |
| 3 | * To compare with behavior when PAS filtering is enabled we load through TC\_LOAD\_FILTER\_PAR following parameters :   sy\_lfr\_pas\_filter\_enabled = 1  sy\_lfr\_pas\_filter\_modulus = 4  sy\_lfr\_pas\_filter\_tbad = 4  sy\_lfr\_pas\_filter\_offset = 0  sy\_lfr\_pas\_filter\_shift = 0  E.g. we consider a full perturbation of 4s | | | | | We should see a total flat response with no counts (amplitude 0) on all ASM products (F0, F1 and F2 every 4s) | | |  |  |
| 4 | End | | | | |  | | |  |  |
| **TEST CASE** | | | | | | | | | | |
| **Test procedure identifier** | | | | | | | **Date** | **Assessment** | | |
| SVS-1202\_Ed1 | | | | | | | DD/MM/YYYY | NT / NA / NOK / POK / OK | | |
| **TS Tested requirements** | | | | | | | **RB Tested requirements** | | | |
| N/A | | | | | | | F0 part of REQ-LFR-SRS-6100\_Ed1 | | | |
| **Component:** | | | | |  | | **Version:**V3++ |  | | |
| **Involved subsystems** | | | | | | | | | | |
| SW: | | | | | 3.2.0.24 | | | | | |
| HW: | | | | | 1.1.91 + StarDundee | | | | | |
| Start time: | | | | | (DD/MM/YYYY) hh:mm | | Test Operator: |  | | |
| End Time: | | | | | (DD/MM/YYYY) hh:mm | | Test Duration: N/A |  | | |
| **Purpose:** | | * Verify PAS filtering behavior and efficiency for F0 channel | | | | | | | | |
| **General remarks about the test** | | | | | | | | | | |
| Requirement Title | | | | LFR REAL TIME FILTERING OF SWA/PAS PERTURBATIONS | | | | | | |
| dependencies | | | | SSS-CP-EQS-761 | | | | | | |
| restrictions | | | |  | | | | | | |
| means | | | |  | | | | | | |
| input data | | | | 4s sweep signal repeated loop from 1536 Hz to 6144 Hz. | | | | | | |
| prerequisite | | | | This test uses lfrcompliance Python notebooks framework and LFR SGSE.  Also, we use specific synchronized start script to ensure that ASM production process starts on the first timecode of on MODULUS (we ensure that we go to NORMAL mode when current coarse time%MODULUS =0 ) AND sweep signal loop is triggered at the same time. | | | | | | |
| used test programs | | | | lfrcompliance/test\_engine\_R3pp/ManualTestsToolkit/ManualStarter.py  lfrcompliance/test\_engine\_R3pp/ManualTestsToolkit/ManualLoadFilterPar.py  lfrcompliance/test\_engine\_R3pp/ManualTestsToolkit/SynchroStartPASandTRIGGER.py | | | | | | |
| test script | | | |  | | | | | | |
| output data | | | | Validation is made directly by monitoring ASM products on LFR SGSE | | | | | | |
| notes | | |  | | | | | | | |
| **Procedure steps** | | | | | | | | | | |
| **Step** | **Actions** | | | | | **Expected Results** | | | **Comments** | **Status** |
| 1 | * Boot LFR * ASM period is set to 4s * Switch to NORMAL mode on the beginning of a MODULUS + sweep signal triggered at the same time. | | | | |  | | |  |  |
| 2 | All LFR PAS filtering parameters are default (No TC\_LOAD\_FILTER\_PAR is sent) e.g. PAS filtering is disabled | | | | | we see all the frequencies from 1536 Hz to 6144 Hz excited on the 4s ASM@F0 :  C:\home\[LPP]\SOLAR ORBITER\Validation\TRR R3pp\ress\noPasFiltering.png | | |  |  |
| 3 | PAS filtering is enabled with default parameters loaded through TC\_LOAD\_FILTER\_PAR:  sy\_lfr\_pas\_filter\_enabled = 1  sy\_lfr\_pas\_filter\_modulus = 4  sy\_lfr\_pas\_filter\_tbad = 1  sy\_lfr\_pas\_filter\_offset = 0  sy\_lfr\_pas\_filter\_shift = 0.5 | | | | | - TC\_LOAD\_FILTER\_PAR is accepted and acquitted : TM\_LFR\_TC\_EXE\_SUCCESS is emitted bu LFR FSW  - Considering the 4s sweep represents 4 time unit of a perturbation, we should see on ASM products at F0 :  a 1 unit hole (TBAD) of total flat response with no counts (amplitude 0) after ½ unit of signal (SHIFT):  C:\home\[LPP]\SOLAR ORBITER\Validation\TRR R3pp\ress\pasDefaultFiltering-annoted.png | | |  |  |
| 4 | PAS filtering is enabled with following parameters loaded through TC\_LOAD\_FILTER\_PAR:  sy\_lfr\_pas\_filter\_enabled = 1  sy\_lfr\_pas\_filter\_modulus = 4  sy\_lfr\_pas\_filter\_tbad = 1  sy\_lfr\_pas\_filter\_offset = 0  sy\_lfr\_pas\_filter\_shift = 0 | | | | | - TC\_LOAD\_FILTER\_PAR is accepted and acquitted : TM\_LFR\_TC\_EXE\_SUCCESS is emitted bu LFR FSW  - Considering the 4s sweep represents 4 time unit of a perturbation, we should see on ASM products at F0 :  a 1 unit hole (TBAD) of total flat response with no counts (amplitude 0) at the beginning:  C:\home\[LPP]\SOLAR ORBITER\Validation\TRR R3pp\ress\pasFilteringConf01-annoted.png | | |  |  |
| 5 | PAS filtering is enabled with following parameters loaded through TC\_LOAD\_FILTER\_PAR:  sy\_lfr\_pas\_filter\_enabled = 1  sy\_lfr\_pas\_filter\_modulus = 4  sy\_lfr\_pas\_filter\_tbad = 1  sy\_lfr\_pas\_filter\_offset = 1  sy\_lfr\_pas\_filter\_shift = 0.5 | | | | | - TC\_LOAD\_FILTER\_PAR is accepted and acquitted : TM\_LFR\_TC\_EXE\_SUCCESS is emitted bu LFR FSW  - Considering the 4s sweep represents 4 time unit of a perturbation, we should see on ASM products at F0 :  a 1 unit hole (TBAD) of total flat response with no counts (amplitude 0) after a 1 unit OFFSET and 1/2 unit of SHIFT:  C:\home\[LPP]\SOLAR ORBITER\Validation\TRR R3pp\ress\pasFilteringConf02-annoted.png | | |  |  |
| 6 | PAS filtering is enabled with following parameters loaded through TC\_LOAD\_FILTER\_PAR:  sy\_lfr\_pas\_filter\_enabled = 1  sy\_lfr\_pas\_filter\_modulus = 4  sy\_lfr\_pas\_filter\_tbad = 0.5  sy\_lfr\_pas\_filter\_offset = 0  sy\_lfr\_pas\_filter\_shift = 0.5 | | | | | - TC\_LOAD\_FILTER\_PAR is accepted and acquitted : TM\_LFR\_TC\_EXE\_SUCCESS is emitted bu LFR FSW  -Considering the 4s sweep represents 4 time unit of a perturbation, we should see on ASM products at F0 :  a 1/2 unit hole (TBAD) of total flat response with no counts (amplitude 0) after 1/2 unit of SHIFT:  C:\home\[LPP]\SOLAR ORBITER\Validation\TRR R3pp\ress\pasFilteringConf03-annoted.png | | |  |  |
| 7 | PAS filtering is enabled with following parameters loaded through TC\_LOAD\_FILTER\_PAR:  sy\_lfr\_pas\_filter\_enabled = 1  sy\_lfr\_pas\_filter\_modulus = 4  sy\_lfr\_pas\_filter\_tbad = 1.0  sy\_lfr\_pas\_filter\_offset = 3  sy\_lfr\_pas\_filter\_shift = 0.5  This step is dedicated to test limit case e.g. when a perturbation is considered to be spread over 2 modulus. | | | | | - TC\_LOAD\_FILTER\_PAR is accepted and acquitted : TM\_LFR\_TC\_EXE\_SUCCESS is emitted bu LFR FSW  - Considering the 4s sweep represents 4 time unit of a perturbation, we should see on ASM products at F0 : a 1 unit hole(TBAD) of total flat response with no counts (amplitude 0) but this hole should be splitted like 0.5 unit at beginning and 0.5 unit at the end because of a 3 unit OFFSET and 0.5 unit SHIFT :  C:\home\[LPP]\SOLAR ORBITER\Validation\TRR R3pp\ress\pasFilteringConf06-annoted.png | | |  |  |
| 8 | End | | | | |  | | |  |  |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE** | | | | | | | | | | |
| **Test procedure identifier** | | | | | | | **Date** | **Assessment** | | |
| SVS-1203\_Ed1 | | | | | | | DD/MM/YYYY | NT / NA / NOK / POK / OK | | |
| **TS Tested requirements** | | | | | | | **RB Tested requirements** | | | |
| N/A | | | | | | | F2 part of REQ-LFR-SRS-6100\_Ed1 | | | |
| **Component:** | | | | |  | | **Version:**V3++ |  | | |
| **Involved subsystems** | | | | | | | | | | |
| SW: | | | | | 3.2.0.24 | | | | | |
| HW: | | | | | 1.1.91 + StarDundee | | | | | |
| Start time: | | | | | (DD/MM/YYYY) hh:mm | | Test Operator: |  | | |
| End Time: | | | | | (DD/MM/YYYY) hh:mm | | Test Duration: N/A |  | | |
| **Purpose:** | | * Verify PAS filtering behavior and efficiency for F2 channel. | | | | | | | | |
| **General remarks about the test** | | | | | | | | | | |
| Requirement Title | | | | LFR REAL TIME FILTERING OF SWA/PAS PERTURBATIONS | | | | | | |
| dependencies | | | | SSS-CP-EQS-761 | | | | | | |
| restrictions | | | | For F2 channel, only 4 instant spectral matrices (e.g. 4 FFT) are used to compute a 4s ASM. | | | | | | |
| means | | | |  | | | | | | |
| input data | | | | 4s custom signal with 0.5s of 16Hz at the end of 1st second, 0.5s of 16Hz and 2.5s of 100hz : PAS\_perturbation\_0.5s-0Hz\_1s-16Hz\_2.5s-100Hz.csv @2048Hz sampling rate:  C:\home\[LPP]\SOLAR ORBITER\Validation\TRR R3pp\ress\custom01.png | | | | | | |
| prerequisite | | | | This test uses lfrcompliance Python notebooks framework and LFR SGSE.  Also, we use specific synchronized start script to ensure that ASM production process starts on the first timecode of on MODULUS (we ensure that we go to NORMAL mode when current coarse time%MODULUS =0 ) AND sweep signal loop is triggered at the same time. | | | | | | |
| used test programs | | | | lfrcompliance/test\_engine\_R3pp/ManualTestsToolkit/ManualStarter.py  lfrcompliance/test\_engine\_R3pp/ManualTestsToolkit/ManualLoadFilterPar.py  lfrcompliance/test\_engine\_R3pp/ManualTestsToolkit/SynchroStartPASandTRIGGER.py | | | | | | |
| test script | | | |  | | | | | | |
| output data | | | | Validation is made directly by monitoring ASM products on LFR SGSE | | | | | | |
| notes | | | Wave form generation configuration made with Digilent WaveForms2015 GUI | | | | | | | |
| **Procedure steps** | | | | | | | | | | |
| **Step** | **Actions** | | | | | **Expected Results** | | | **Comments** | **Status** |
| 1 | * Boot LFR * ASM period is set to 4s * Switch to NORMAL mode on the beginning of a MODULUS + custom signal triggered at the same time. | | | | |  | | |  |  |
| 2 | All LFR PAS filtering parameters are default (No TC\_LOAD\_FILTER\_PAR is sent) e.g. PAS filtering is disabled | | | | | We should see the 2 peaks at 16 Hz and 100 Hz on the 4s ASM@F2 with expected amplitudes e.g because of different signals length and integration periods : 100 Hz peak should be around 3 times stronger than 16Hz peak.C:\home\[LPP]\SOLAR ORBITER\Validation\TRR R3pp\ress\noPasFiltering-02.png | | |  |  |
| 3 | PAS filtering is enabled with default parameters loaded through TC\_LOAD\_FILTER\_PAR:  sy\_lfr\_pas\_filter\_enabled = 1  sy\_lfr\_pas\_filter\_modulus = 4  sy\_lfr\_pas\_filter\_tbad = 1  sy\_lfr\_pas\_filter\_offset = 0  sy\_lfr\_pas\_filter\_shift = 0.5 | | | | | - TC\_LOAD\_FILTER\_PAR is accepted and acquitted : TM\_LFR\_TC\_EXE\_SUCCESS is emitted bu LFR FSW  -We should see the filtering of the 16Hz peak AND an attenuation of the 100Hz Peak (~10%). This is because the 16Hz perturbation affects the 2 first SM so we should drop a part of the 100Hz signal.  C:\home\[LPP]\SOLAR ORBITER\Validation\TRR R3pp\ress\pasFilteringConf04.png | | |  |  |
| 4 | PAS filtering is enabled with following parameters loaded through TC\_LOAD\_FILTER\_PAR:  sy\_lfr\_pas\_filter\_enabled = 1  sy\_lfr\_pas\_filter\_modulus = 4  sy\_lfr\_pas\_filter\_tbad = 0.49  sy\_lfr\_pas\_filter\_offset = 0  sy\_lfr\_pas\_filter\_shift = 0.5 | | | | | - TC\_LOAD\_FILTER\_PAR is accepted and acquitted : TM\_LFR\_TC\_EXE\_SUCCESS is emitted bu LFR FSW  - We should see a part of the 16Hz peak perturbation (amplitude should be ~60% lower than unfiltered conf) due to the fact that TBAD does not cover all the perturbation (16Hz is emitted in the [1s, 1.5s] range):  C:\home\[LPP]\SOLAR ORBITER\Validation\TRR R3pp\ress\pasFilteringConf05.png | | |  |  |
| 5 | End | | | | |  | | |  |  |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE** | | | | | | | | | | |
| **Test procedure identifier** | | | | | | | **Date** | **Assessment** | | |
| SVS-1204\_Ed1 | | | | | | | DD/MM/YYYY | NT / NA / NOK / POK / OK | | |
| **TS Tested requirements** | | | | | | | **RB Tested requirements** | | | |
| N/A | | | | | | | F1 part of REQ-LFR-SRS-6100\_Ed1 | | | |
| **Component:** | | | | |  | | **Version:**V3++ |  | | |
| **Involved subsystems** | | | | | | | | | | |
| SW: | | | | | 3.2.0.24 | | | | | |
| HW: | | | | | 1.1.91 + StarDundee | | | | | |
| Start time: | | | | | (DD/MM/YYYY) hh:mm | | Test Operator: |  | | |
| End Time: | | | | | (DD/MM/YYYY) hh:mm | | Test Duration: N/A |  | | |
| **Purpose:** | | * Verify PAS filtering behavior and efficiency for F1 channel. | | | | | | | | |
| **General remarks about the test** | | | | | | | | | | |
| Requirement Title | | | | LFR REAL TIME FILTERING OF SWA/PAS PERTURBATIONS | | | | | | |
| dependencies | | | | SSS-CP-EQS-761 | | | | | | |
| restrictions | | | |  | | | | | | |
| means | | | |  | | | | | | |
| input data | | | | 4s custom sweep signal repeated loop with 1s of 96Hz then 1s of 144Hz then 1s of 192Hz then 1s of 240Hz.  PAS\_perturbation\_1s-96-144-192-240.csv @2048Hz sampling rate:C:\home\[LPP]\SOLAR ORBITER\Validation\TRR R3pp\ress\custom02.png | | | | | | |
| prerequisite | | | | This test uses lfrcompliance Python notebooks framework and LFR SGSE.  Also, we use specific synchronized start script to ensure that ASM production process starts on the first timecode of on MODULUS (we ensure that we go to NORMAL mode when current coarse time%MODULUS =0 ) AND sweep signal loop is triggered at the same time. | | | | | | |
| used test programs | | | | lfrcompliance/test\_engine\_R3pp/ManualTestsToolkit/ManualStarter.py  lfrcompliance/test\_engine\_R3pp/ManualTestsToolkit/ManualLoadFilterPar.py  lfrcompliance/test\_engine\_R3pp/ManualTestsToolkit/SynchroStartPASandTRIGGER.py | | | | | | |
| test script | | | |  | | | | | | |
| output data | | | | Validation is made directly by monitoring ASM products on LFR SGSE | | | | | | |
| notes | | | Wave form generation configuration made with Digilent WaveForms2015 GUI | | | | | | | |
| **Procedure steps** | | | | | | | | | | |
| **Step** | **Actions** | | | | | **Expected Results** | | | **Comments** | **Status** |
| 1 | * Boot LFR * ASM period is set to 4s * Switch to NORMAL mode on the beginning of a MODULUS + custom signal triggered at the same time. | | | | |  | | |  |  |
| 2 | All LFR PAS filtering parameters are default (No TC\_LOAD\_FILTER\_PAR is sent) e.g. PAS filtering is disabled | | | | | We should see the 4 peaks of injected frequencies : 96 Hz, 144Hz, 192Hz and 240Hz on the 4s ASM@F1 with approximately the same amplitude (1.2x10^6 counts):C:\home\[LPP]\SOLAR ORBITER\Validation\TRR R3pp\ress\noPasFiltering-03.png | | |  |  |
| 3 | PAS filtering is enabled with following parameters loaded through TC\_LOAD\_FILTER\_PAR:  sy\_lfr\_pas\_filter\_enabled = 1  sy\_lfr\_pas\_filter\_modulus = 4  sy\_lfr\_pas\_filter\_tbad = 1.0  sy\_lfr\_pas\_filter\_offset = 0  sy\_lfr\_pas\_filter\_shift = 0.0 | | | | | - TC\_LOAD\_FILTER\_PAR is accepted and acquitted : TM\_LFR\_TC\_EXE\_SUCCESS is emitted by LFR FSW  -We should see the filtering of the freq injected during the first second e.g. 96Hz peak should be filtered :  C:\home\[LPP]\SOLAR ORBITER\Validation\TRR R3pp\ress\pasFilteringConf10.png | | |  |  |
| 4 | PAS filtering is enabled with following parameters loaded through TC\_LOAD\_FILTER\_PAR:  sy\_lfr\_pas\_filter\_enabled = 1  sy\_lfr\_pas\_filter\_modulus = 4  sy\_lfr\_pas\_filter\_tbad = 1.0  sy\_lfr\_pas\_filter\_offset = 1  sy\_lfr\_pas\_filter\_shift = 0.0 | | | | | - TC\_LOAD\_FILTER\_PAR is accepted and acquitted : TM\_LFR\_TC\_EXE\_SUCCESS is emitted by LFR FSW  -We should see the filtering of the freq injected during the 2nd second e.g. 144Hz peak should be filtered :  C:\home\[LPP]\SOLAR ORBITER\Validation\TRR R3pp\ress\pasFilteringConf11.png | | |  |  |
| 5 | PAS filtering is enabled with following parameters loaded through TC\_LOAD\_FILTER\_PAR:  sy\_lfr\_pas\_filter\_enabled = 1  sy\_lfr\_pas\_filter\_modulus = 4  sy\_lfr\_pas\_filter\_tbad = 1.0  sy\_lfr\_pas\_filter\_offset = 2  sy\_lfr\_pas\_filter\_shift = 0.0 | | | | | - TC\_LOAD\_FILTER\_PAR is accepted and acquitted : TM\_LFR\_TC\_EXE\_SUCCESS is emitted by LFR FSW  -We should see the filtering of the freq injected during the 3rd second e.g. 192Hz peak should be filtered :  C:\home\[LPP]\SOLAR ORBITER\Validation\TRR R3pp\ress\pasFilteringConf12.png | | |  |  |
| 6 | PAS filtering is enabled with following parameters loaded through TC\_LOAD\_FILTER\_PAR:  sy\_lfr\_pas\_filter\_enabled = 1  sy\_lfr\_pas\_filter\_modulus = 4  sy\_lfr\_pas\_filter\_tbad = 1.0  sy\_lfr\_pas\_filter\_offset = 3  sy\_lfr\_pas\_filter\_shift = 0.0 | | | | | - TC\_LOAD\_FILTER\_PAR is accepted and acquitted : TM\_LFR\_TC\_EXE\_SUCCESS is emitted by LFR FSW  -We should see the filtering of the freq injected during the 4th second e.g. 240Hz peak should be filtered :  C:\home\[LPP]\SOLAR ORBITER\Validation\TRR R3pp\ress\pasFilteringConf13.png | | |  |  |
| 7 | PAS filtering is enabled with default parameters loaded through TC\_LOAD\_FILTER\_PAR:  sy\_lfr\_pas\_filter\_enabled = 1  sy\_lfr\_pas\_filter\_modulus = 4  sy\_lfr\_pas\_filter\_tbad = 1.0  sy\_lfr\_pas\_filter\_offset = 0  sy\_lfr\_pas\_filter\_shift = 0.5 | | | | | - TC\_LOAD\_FILTER\_PAR is accepted and acquitted : TM\_LFR\_TC\_EXE\_SUCCESS is emitted by LFR FSW  -We should see the filtering of half the freq injected during the 1st second and half the freq injected during the 2nd second e.g. amplitudes of 96Hz and 144Hz peaks should be lowered by ~50%:  C:\home\[LPP]\SOLAR ORBITER\Validation\TRR R3pp\ress\pasFilteringConf14.png | | |  |  |
| 8 | PAS filtering is enabled with following parameters loaded through TC\_LOAD\_FILTER\_PAR:  sy\_lfr\_pas\_filter\_enabled = 1  sy\_lfr\_pas\_filter\_modulus = 4  sy\_lfr\_pas\_filter\_tbad = 3.0  sy\_lfr\_pas\_filter\_offset = 0  sy\_lfr\_pas\_filter\_shift = 0.0 | | | | | - TC\_LOAD\_FILTER\_PAR is accepted and acquitted : TM\_LFR\_TC\_EXE\_SUCCESS is emitted by LFR FSW  -We should see the filtering of the freqs injected during the 3 first seconds : 96Hz, 144Hz and 192Hz:  C:\home\[LPP]\SOLAR ORBITER\Validation\TRR R3pp\ress\pasFilteringConf15.png | | |  |  |
| 9 | End | | | | |  | | |  |  |

# Validation test platform requirements

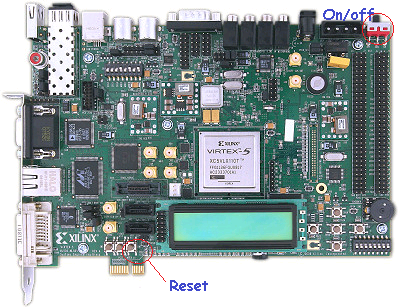
NB 1 : Since January 2015, validation test platform has been deeply upgraded : LPPMON has been replaced by SocExplorer. Also, RMAP Plugin has been split into LFRControlPlugin and SpwPlugin. Part of the following steps are impacted by those changes. Please report to SGSE user manual for up to date procedures ([RD07]).

NB 2 : A system clock simulator ([RD08]) has been developed to emulate the Central Time Reference behavior (a spacewire command packet coupled to a spacewire timecode sent to LFR analyzer). This clock simulator is used in all test cases related to time management.

This section describes the startup of the target.

The PC driving the SpW board is supposed to nominal state. Note lppmon soft is not launched.

The board powered on, and its switch is initially off. See figure below:



Led

The "SpaceWire-USB Brick" is disconnected from the PC.



Preliminary steps to run **without going too fas**t:

1. Ensure USB "SpaceWire-USB Brick" is connected to the Board via «Link 1”.
2. Ensure RS232 link of the board is connected to the PC.
3. Connect USB "SpaceWire-USB Brick" to the PC. Ensure (green or red) LEDs are lit (the S/N:32111332 brick may be used). Connect the brick to the board (Link 1)
4. Switch on the board.
5. Initialize by reset (possibly several times) the board. Note this reset unload any FSW out of the LEON.
6. Use LppMon to open the bridge.

Link 1nk 1

Link 2

1nk 1

2nk 1

3nk 1

4nk 1

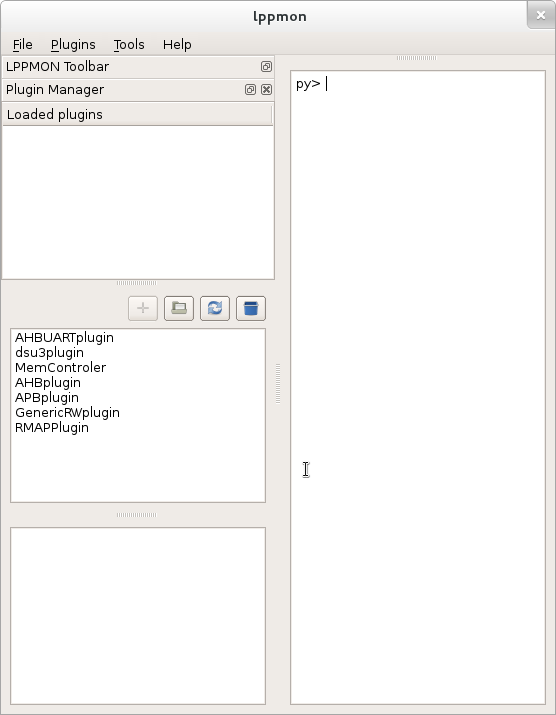
1. Usb is not connected
2. Usb connected; strart up in progress
3. Start up completed.
4. Bridge (link 1) opened; brick ready.
5. Power off
6. Begin of start up
7. Start up in progress (10s)
8. Start up completed
9. (Reset pushed)
10. Grmon launched
11. fsw load in progress
12. Soft loaded
13. Run of loaded fsw (bridge opened)

See below to begin the exchanges with the target. There are two independent ways: by commands (load FSW, search for information about HW …), or by a dedicated HMI.

1 LppMon HM:

Launch LppMon with associated icon:

In lppmon/LPPMON Toolbar/Plug-in Manager, plug-in are loadable by drag/move:

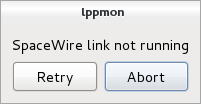


Instantiate the root plugging “RMAPPlugin0” (in “LPPMON Toolbar/Plugin Manager/Loaded plugins”).

In “RMAP and SPW Communication/connection” tab, select Bridge selection=”Star Dundee”

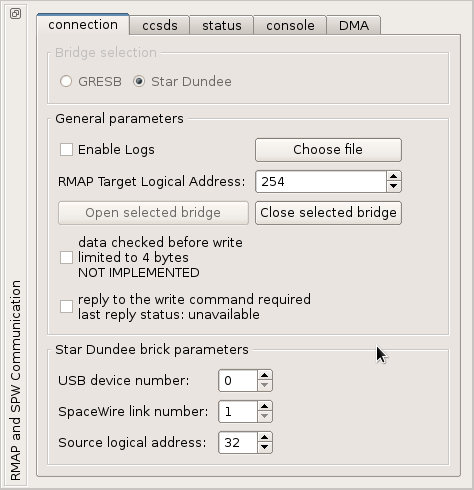
In “LPPMON Toolbar/Plugin Manager”, instantiate the useful child plug-in: GenericRWplugin (Generic Driver for the Memory editor), MemControler (Memctrlr Driver for the Memory Check) …

Activate SpW exchanges with loaded plug-in: in RMAP and SPW Communication/connection/General parameters, click on the “Open selected bridge” button (Plug-ins are activated after the “Open->Close” selected bridge transition).



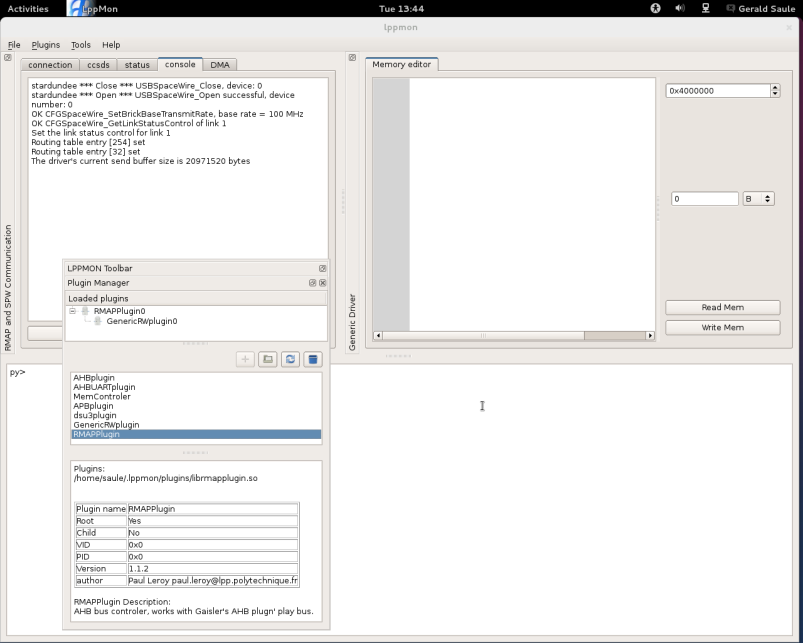
Note, in case of randomly non-synchronization, that box appears. Then, re-launch the brick until this problem is solved: Reset the brick, wait a few seconds, click “Retry”.

The final result is:



Ensure the LED on brick is green for the link used.

Ensure nominal logs are:



Instantiated objects are accessible by the LppMon Python console. List them with dir() command:

py> dir()

['BUTTON\_rmapCloseCommunication', 'BUTTON\_rmapOpenCommunication', 'BUTTON\_selectGRESB', 'BUTTON\_selectStarDundee', 'GRESB\_Bridge', 'GenericRWplugin0', 'Pylppmon', 'Qt', 'QtCore', 'QtGui', 'QtNetwork', 'QtOpenGL', 'QtSql', 'QtSvg', 'QtUiTools', 'QtWebKit', 'QtXml', 'RMAPPlugin0', '\_\_builtins\_\_', '\_\_doc\_\_', '\_\_name\_\_', '\_\_package\_\_', 'lppmonEngine', 'private', 'proxy']

Notice to type multi-line commands, use <SHIFT>+<ENTER> keys (ensure the Python indentation).

2 LFR FSW

In case of randomly non-synchronization, initialize by reset (possibly several times) the board. Note this reset unload any FSW out of the LEON.

Type grmon-eval, or grmon-eval -u, (use the auto completion features) in a console.

Note the port /dev/ttyS0 is used for displays.

Moreover, if the serial port uses a USB port, type “grmon-eval -uart /dev/ttyUSB0 –u”

Then, these instructions may be used: **load** (only if serial link is connected), **run**, **help**, **info sys**, **exit**. Example for a console session:

[saule@pc-alison ~]$ grmon-eval

GRMON LEON debug monitor v1.1.56 evaluation version

Copyright (C) 2004-2011 Aeroflex Gaisler - all rights reserved.

For latest updates, go to http://www.gaisler.com/

Comments or bug-reports to support@gaisler.com

This evaluation version will expire on 21/9/2013

using port /dev/ttyS0 @ 115200 baud

Device ID: : 0x509

GRLIB build version: 4098

initialising ................

detected frequency: 30 MHz

Component Vendor

LEON3 SPARC V8 Processor Gaisler Research

Unknown device Gaisler Research

AHB Debug UART Gaisler Research

SVGA Controller Gaisler Research

DDR2 Controller Gaisler Research

AHB/APB Bridge Gaisler Research

LEON3 Debug Support Unit Gaisler Research

LEON2 Memory Controller European Space Agency

System ACE I/F Controller Gaisler Research

Generic APB UART Gaisler Research

Multi-processor Interrupt Ctrl Gaisler Research

Modular Timer Unit Gaisler Research

PS/2 interface Gaisler Research

PS/2 interface Gaisler Research

General purpose I/O port Gaisler Research

AHB status register Gaisler Research

Use command 'info sys' to print a detailed report of attached cores

grlib> info sys

00.01:003 Gaisler Research LEON3 SPARC V8 Processor (ver 0x0)

ahb master 0

01.01:029 Gaisler Research Unknown device (ver 0x0)

ahb master 1, irq 10

apb: 80000900 - 80000a00

02.01:007 Gaisler Research AHB Debug UART (ver 0x0)

ahb master 2

apb: 80000700 - 80000800

baud rate 115200, ahb frequency 30.00

03.01:063 Gaisler Research SVGA Controller (ver 0x0)

ahb master 3

apb: 80000600 - 80000700

DCOM communication error, retrying ...

00.01:02e Gaisler Research DDR2 Controller (ver 0x0)

ahb: 40000000 - 60000000

ahb: fff00100 - fff00200

64-bit DDR2 : 1 \* 256 Mbyte @ 0x40000000, 4 internal banks

190 MHz, col 10, ref 7.8 us, trfc 131 ns

01.01:006 Gaisler Research AHB/APB Bridge (ver 0x0)

ahb: 80000000 - 80100000

02.01:004 Gaisler Research LEON3 Debug Support Unit (ver 0x1)

ahb: 90000000 - a0000000

AHB trace 128 lines, 32-bit bus, stack pointer 0x4ffffff0

CPU#0 win 8, hwbp 2, itrace 128, srmmu, lddel 1

icache 2 \* 8 kbyte, 32 byte/line lru

dcache 1 \* 8 kbyte, 16 byte/line

03.04:00f European Space Agency LEON2 Memory Controller (ver 0x1)

ahb: 00000000 - 20000000

ahb: 20000000 - 40000000

ahb: c0000000 - c2000000

apb: 80000000 - 80000100

16-bit prom @ 0x00000000

04.01:067 Gaisler Research System ACE I/F Controller (ver 0x0)

irq 13

ahb: fff00200 - fff00300

01.01:00c Gaisler Research Generic APB UART (ver 0x1)

irq 2

apb: 80000100 - 80000200

DCOM communication error, retrying ...

baud rate 3750000

02.01:00d Gaisler Research Multi-processor Interrupt Ctrl (ver 0x3)

apb: 80000200 - 80000300

03.01:011 Gaisler Research Modular Timer Unit (ver 0x0)

irq 8

apb: 80000300 - 80000400

8-bit scaler, 2 \* 32-bit timers, divisor 30

04.01:060 Gaisler Research PS/2 interface (ver 0x2)

irq 4

apb: 80000400 - 80000500

05.01:060 Gaisler Research PS/2 interface (ver 0x2)

irq 5

apb: 80000500 - 80000600

08.01:01a Gaisler Research General purpose I/O port (ver 0x0)

apb: 80000800 - 80000900

0f.01:052 Gaisler Research AHB status register (ver 0x0)

irq 7

apb: 80000f00 - 80001000

grlib>

Note the Component=”Unknown device” (Vendor=”Gaisler Research”) corresponds to the unit under test.

When the opened bridge is defined, the “run” command provides:

grlib> run

In configure\_spw\_link \*\*\* /dev/grspw0 opened and started successfully

In configure\_spw\_link \*\*\* /dev/grspw0 configured successfully

In SMIQ \*\*\*

In SPIQ \*\*\* Waiting for SPW\_LINKERR\_EVENT

In ACTN \*\*\*

In AVFO \*\*\*

In BPFO \*\*\*

In WFRM \*\*\*

In STAT \*\*\*

In RECV \*\*\*

Serial link lost

In this unfortunate case, typical message provided:

Cannot continue, processor not in debug mode

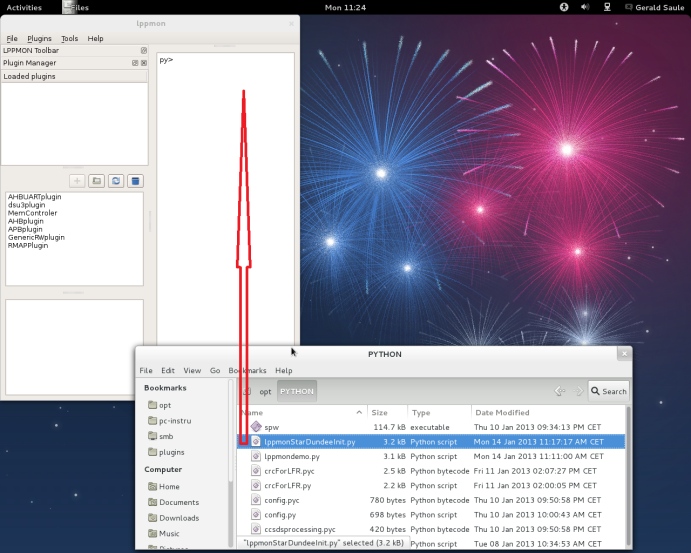
IU in error mode (tt = 0x00)

fff90688 05000000 sethi %hi(0x0), %g2

Leave (<CTRL>+<C>) and restart the GRMON session.

3.Scripts

Drag the \*.py script in the “py>” part of the lppmon console.



Note the explicit command exists:

execfile('/opt/PYTHON/lfrverif/LFR\_SVS/<path>/action.py')

Moreover, tests generate several log files:

-Raw-TC

-Raw-TM

-Synthetic sequence of TC/TM

-The chronology of TC/TM with the detail of all the fields.

(a control mechanism is implemented to tag any problem "/!\").

Note it’s possible to generate the synthesis and detailed files from raw TC/TM:

py> test\_launcher.test\_monitor.loadRecord("2013\_05\_29-11:07:09")

py> test\_launcher.test\_monitor.buildSynthesis("2013\_05\_29-11:07:09")

# Software validation specification additional information

All scripts, results, logs and backtraces of each SVS-xxx test case presented in this document is versioned on a mercurial deposit hosted under a Rhodecode server here :

<https://hephaistos.lpp.polytechnique.fr/rhodecode/HG_REPOSITORIES/LPP/INSTRUMENTATION/SOLO_LFR/VALIDATION_R3>

Each test case has his own directory identified by his SVS number.