Critical Software

Test Case Specification

Spacecraft thermal control system

CONTRACT REFERENCE: n/a

Date: 2024-07-24

Project Code: N/A

Doc. Ref.: n/a

Status: DRAFT

Pages: 10

Information Classification: INTERNAL

Version: 01

|  |  |  |  |
| --- | --- | --- | --- |
| Authors and Contributors | | | |
| Name | Description | Date |  |
|  | Author |  |  |
| Gustavo Dinis | Author |  |  |

|  |
| --- |
| COPYRIGHT |
| The contents of this document are under copyright of Critical Software, released on condition that it shall not be copied in whole, in part or otherwise reproduced (whether by photographic or any other method) and therefore shall not be divulged to any person other than the addressee (save to other authorized offices of his organization having the need to know such contents, for the purpose for which disclosure is made) without prior written consent of the CSW Quality Department. |

|  |  |  |  |
| --- | --- | --- | --- |
| Revision History | | | |
| Version | Date | Description | Author |
| 0.1 |  | Initial draft of the document. | Tomás Silva |

Table of Contents

[1. Introduction 4](#_Toc173241171)

[1.1. Objective 4](#_Toc173241172)

[1.2. Scope 4](#_Toc173241173)

[1.3. Audience 4](#_Toc173241174)

[1.4. Document Structure 4](#_Toc173241175)

[1.5. Applicable Documents 4](#_Toc173241176)

[1.6. Reference Documents 4](#_Toc173241177)

[2. Test Plan 5](#_Toc173241178)

[2.1. Features to be Tested 5](#_Toc173241179)

[2.2. Features not to be Tested 5](#_Toc173241180)

[2.3. Approach 5](#_Toc173241181)

[2.4. Testing Tasks 5](#_Toc173241182)

[2.5. Environment Needs 5](#_Toc173241183)

[2.6. Test Case Identification 5](#_Toc173241184)

[3. Test Cases 7](#_Toc173241185)

[3.1. Usage Scneario <1..> - <Module name> 7](#_Toc173241186)

[3.1.1. <Test ID> - <Test Title> 7](#_Toc173241187)

[Annex A. Test Specification table format 9](#_Toc173241188)

Table of Tables

Table of Figures

# Introduction

## Objective

The objective of this Test Case Specification document is to define the detailed test cases that will be used to validate the functionality, performance, and reliability of the Spacecraft Thermal Control System. This document aims to ensure that all aspects of the system are thoroughly tested to meet the specified requirements and to identify any defects or issues that need to be addressed prior to deployment.

## Scope

This document covers all test cases related to the Spacecraft Thermal Control System, including but not limited to functional tests, performance tests, integration tests, and user interface tests. The scope includes test cases for the Temperature Simulation Library, Thermal Control Function, User Visualization Interface and the system integration. It encompasses both automated and manual testing procedures.

## Audience

The intended audience of this document is the team of Summer Camp interns responsible for the validation of the Spacecraft Thermal Control System modules.

## Document Structure

The Test Case Specification document is structured to provide a comprehensive framework for testing the Spacecraft Thermal Control System and is divided into the following sections:

* Section 1 (Introduction): This section provides an overview of the document, including its objective, scope, intended audience, and structure. It also lists applicable and reference documents that are relevant to the testing process.
* Section 2 (Test Plan): This section outlines the test plan for the Spacecraft Thermal Control System. It includes detailed descriptions of the features to be tested, features not to be tested, the testing approach, specific testing tasks, and the environmental needs required for testing.
* Section 3 (Test Cases): This section contains detailed test cases for various usage scenarios, starting with the Thermal Simulator Library. Each test case includes the necessary steps to validate the functionality and performance of different components of the system.

## Applicable Documents

There are no applicable documents for this document.

## Reference Documents

There are no reference documents for this document.

# Test Plan

## Features to be Tested

Identify all software features and combinations of software features to be tested.

## Features not to be Tested

Identify all features and significant combinations of features that will not be tested and the reasons.

## Approach

Describe the overall approach to testing. For each major group of features or combination of features, specify the approach that will ensure that these feature groups are adequately tested. Specify the major activities, techniques, and tools that are used to test the designated groups of features.

The approach should be described in sufficient detail to permit identification of the major testing tasks and estimation of the time required to do each one.

Specify the minimum degree of comprehensiveness desired. Identify the techniques that will be used to judge the comprehensiveness of the testing effort (e.g. determining which statements have been executed at least once). Specify any additional completion criteria (e.g. error frequency). The techniques to be used to trace requirements should be specified.

Identify significant constraints on testing such as test item availability, testing resource availability, and deadlines.

## Testing Tasks

Identify the set of tasks necessary to prepare for and perform testing. Identify all intertask dependencies and any special skills required.

## Environment Needs

Specify both the necessary and desired properties of the test environment. This specification should contain the physical characteristics of the facilities including the hardware, the communication and system software, the mode of usage (e.g. stand-alone), and any other software or supplies needed to support the test. Also specify the level of security that must be provided for the test facilities, system software, and proprietary components such as software, data, and hardware.

## Test Case Identification

Each test case within this specification is uniquely identified using a structured ID format to ensure clarity and consistency. The test case IDs follow the format **TC-<MODULE ACRONYM>-<XXXX>**, where:

* TC: Stands for "Test Case" indicating the nature of the document.
* <MODULE ACRONYM>: Represents a short, descriptive acronym of the module or component being tested
  + TSL – Thermal Simulator Library
  + TCF – Thermal Control Function
  + VUI – Visualization User Interface
* <XXXX>: A unique numerical identifier for each test case within the module, starting from 0010 and increasing in increments of 10 (e.g., 0010, 0020, 0030). This numbering scheme allows for easy identification and referencing of test cases, as well as the insertion of additional test cases in future updates without requiring a renumbering of existing ones.

# Test Cases

## Usage SceNario 1 – Thermal Simulation Library

### TC-STL-0010 – Environment Initialization

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | ***TC-STL-0010*** | | | |
| **Verification Method** | Inspection | | | |
| **Objective** | Ensure that the temperature of every thermistor, the status of every heater and the environmental status are initialized correctly. | | | |
| **Procedure Description** | *Ensure the TSL\_init function is called at the start of the execution and initializes the simulation period to NORMAL. Ensure that the thermistors and heaters are initialized correctly.* | | | |
| **Requirement IDs** | ***STCS-SRS-TSL-FUNC-0010*** The TSL shall initialize the simulation environment with the following parameters:   * Temperatures of each thermistor (in degrees Celsius): <random value between -5 and 7> * Status of each heater: OFF * Simulation period: NORMAL | | | |
| **Pass/Fail Criteria** | *The test fails if the temperatures are initialized with values outside the range –5:7 or any of the heaters are not OFF or the current period is not set to NORMAL and passes otherwise.* | | | |
| **Test Step** | **Test Step Description** | **Expected Result** | **Test Result** | **Pass/Fail** |
| **001** | Ensure TSL\_init function initializes the variable corresponding to the simulation period to NORMAL | Variable is initialized to NORMAL | <Image of code section> |  |
| **002** | Ensure the TSL\_init function is called at the start of the execution | Function is called at the start of the execution | <Image of code section> |  |
| **003** | Ensure that the thermistors are initialized correctly | Temperature between –5 and 7 | <Image of code section> |  |
| **004** | Ensure that the heaters are initialized correctly | <Image of code section> |  |  |
| **Test Status** |  | | | |
| **Date** |  | | | |
| **Test Conductor** |  | | | |

### TC-STL-0020 – Thermistor-Heater Relationships

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | ***TC-STL-0020*** | | | |
| **Verification Method** | Test | | | |
| **Objective** | Ensure that when enabling/disabling a heater only one thermistor temperature is going to be affected. | | | |
| **Procedure Description** | *Turn on each heater one by one and ensure only the correspondent thermistor changes the temperature.* | | | |
| **Requirement IDs** | ***STCS-SRS-TSL-FUNC-0020*** *The relationships between thermistors and heaters shall be ensured:* *E.g:. THERM-01 – HTR-01* | | | |
| **Pass/Fail Criteria** | *The test fails if any heater changes the temperature of any thermistor besides its own.* | | | |
| **Test Step** | **Test Step Description** | **Expected Result** | **Test Result** | **Pass/Fail** |
| **001** | Ensure THERM-01 and HTR-01 relationship | When HTR-01 is turned on, only THERM-01 changes temperature |  |  |
| **002** | Ensure THERM-02 and HTR-02 relationship | When HTR-02 is turned on, only THERM-02 changes temperature |  |  |
| **003** | Ensure THERM-03 and HTR-03 relationship | When HTR-03 is turned on, only THERM-03 changes temperature |  |  |
| **004** | Ensure THERM-04 and HTR-04 relationship | When HTR-04 is turned on, only THERM-04 changes temperature |  |  |
| **Test Status** |  | | | |
| **Date** |  | | | |
| **Test Conductor** |  | | | |

### TC-STL-0030 – System Monotonic Clock & Simulation Cycle Interval

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | | | | ***TC-STL-0030*** | | | |
| **Verification Method** | | | | Test | | | |
| **Objective** | | | | Ensure the program maintains a 16-bit clock variable and it is incremented 1 unit every simulation cycle with a 5Hz interval; Ensure the temperature of each thermistor is changes every clock cycle. | | | |
| **Procedure Description** | | | | Check the monotonic clock variable size. Run the program for 2 seconds to check that the clock is incremented by 1 unit at a rate of 5Hz and the temperature of each thermistor changes every clock cycle. | | | |
| **Requirement IDs** | | | | ***STCS-SRS-TSL-FUNC-0030*** The TSL shall maintain a monotonic clock on a 16-bit variable, incremented by 1 unit every simulation cycle.  ***STCS-SRS-TSL-FUNC-0040*** The TSL simulation cycle interval shall be 5Hz.  ***STCS-SRS-TSL-FUNC-0050*** The TSL shall update the temperature values for the 4 available thermistors every simulation cycle, based on environmental conditions such as solar exposure and eclipse periods. | | | |
| **Pass/Fail Criteria** | | | | *The test fails if the clock doesn’t have a size of 16 bits, doesn’t increment exactly 1 unit every simulation cycle, it has a rate different than 5Hz or one or more transistors maintain the same temperature through two or more consecutive clock cycles.* | | | |
| **Test Step** | | | | **Test Step Description** | **Expected Result** | **Test Result** | **Pass/Fail** |
| **001** | | | | Check the monotonic clock variable size | 16-bit variable |  |  |
| **002** | Run the program for 2 seconds and open the log. | Clock must increment 5 times 1 unit each second |  |  | | | |
| **003** | | | | Check if the temperature values change for the 4 thermistors every simulation cycle | The temperature change for the 4 thermistors every simulation cycle |  |  |
| **Test Status** | | | |  | | | |
| **Date** | | | |  | | | |
| **Test Conductor** | | | |  | | | |

### TC-STL-0040 – Simulation Current Period Update

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | ***TC-STL-0040*** | | | |
| **Verification Method** | Test | | | |
| **Objective** | Confirm that when the values of the last 8 bits of the clock change, it changes the simulation cycle accordingly. | | | |
| **Procedure Description** | Run the program for 51 seconds and check the log to see that it changes the simulation current period at the correct timestamps. | | | |
| **Requirement IDs** | ***STCS-SRS-TSL-FUNC-0060*** Whenever the system’s monotonic clock lower 8 bits are 0x00 to 0x1F or 0x60 to 0xFF, the TSL shall set the current period to NORMAL.  ***STCS-SRS-TSL-FUNC-0080*** Whenever the system’s monotonic clock lower 8 bits are 0x20 to 0x3F, the TSL shall set the current period to ECLIPSE.  ***STCS-SRS-TSL-FUNC-0100*** Whenever the system’s monotonic clock lower 8 bits are 0x40 to 0x5F, the TSL shall set the current period to SUN\_EXPOSURE. | | | |
| **Pass/Fail Criteria** | *The test fails if at any moment the current period doesn’t correspond to the expected one.* | | | |
| **Test Step** | **Test Step Description** | **Expected Result** | **Test Result** | **Pass/Fail** |
| **001** | Check the log for when the last 8-bits of the clock are between 0x00 and 0x1, or 0x60 and 0xFF | The simulation period should be NORMAL |  |  |
| **002** | Check the log for when the last 8-bits of the clock are between 0x20 and 0x3F | The simulation period should be ECLIPSE |  |  |
| **003** | Check the log for when the last 8-bits of the clock are between 0x40 and 0x5F | The simulation period should be SUN\_EXPOSURE |  |  |
| **Test Status** |  | | | |
| **Date** |  | | | |
| **Test Conductor** |  | | | |

### TC-STL-0050 – Period Temperatures

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case ID** | ***TC-STL-0050*** | | | | |
| **Verification Method** | Test | | | | |
| **Objective** | Confirm that the temperature values of each thermistor are correctly changed according to the simulation period. | | | | |
| **Procedure Description** | Set the current period to NORMAL, ECLIPSE or SUN\_EXPOSURE and for ach turn the heater ON or OFF and check the temperature changes in the corresponding thermistors. | | | | |
| **Requirement IDs** | ***STCS-SRS-TSL-FUNC-0070*** Whenever the current period is set to NORMAL, if the Heater is ON the temperature of the thermistor should increase by 1 unit and if the Heater is OFF the temperature of the thermistor should decrease by 1 unit  ***STCS-SRS-TSL-FUNC-0090*** Whenever the current period is set to ECLIPSE, if the Heater is ON the temperature of the thermistor should increase by 4 units and if the Heater is OFF the temperature of the thermistor should decrease by 7 units  ***STCS-SRS-TSL-FUNC-0110*** Whenever the current period is set to SUN\_EXPOSURE, if the Heater is ON the temperature of the thermistor should increase by 7 units and if the Heater is OFF the temperature of the thermistor should decrease by 1 unit | | | | |
| **Pass/Fail Criteria** | *The test fails if the thermistor temperature doesn’t vary accordingly for each simulation period and heater status combination.* | | | | |
| **Test Step** | **Test Step Description** | | **Expected Result** | **Test Result** | **Pass/Fail** |
| **001** | Set the current period to NORMAL and turn the heater ON | | The thermistor should increase the temperature by 1 unit |  |  |
| **002** | Set the current period to NORMAL and turn the heater OFF | The thermistor should decrease the temperature by 1 unit |  | |  |
| **003** | Set the current period to ECLIPSE and turn the heater ON | The thermistor should increase the temperature by 4 units |  | |  |
| **004** | Set the current period to ECLIPSE and turn the heater OFF | | The thermistor should decrease the temperature by 7 units |  |  |
| **005** | Set the current period to SUN\_EXPOSURE and turn the heater ON | | The thermistor should increase the temperature by 7 units |  |  |
| **006** | Set the current period to SUN\_EXPOSURE and turn the heater OFF | | The thermistor should decrease the temperature by 1 unit |  |  |
| **Test Status** |  | | | | |
| **Date** |  | | | | |
| **Test Conductor** |  | | | | |

### TC-STL-0060 – API Test

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | ***TC-STL-0060*** | | | |
| **Verification Method** | Test | | | |
| **Objective** | Ensure that the TSL provide APIs to retrieve the current temperature of each thermistor, the current power status of each heater and to set the status of a given heater. | | | |
| **Procedure Description** | Check each API: for the temperature of the thermistor and the heater status check the data received; for the heater status setting check that the status is updated after a request. | | | |
| **Requirement IDs** | ***STCS-SRS-TSL-FUNC-0120*** The TSL shall provide an API to retrieve the current temperature of each thermistor.  ***STCS-SRS-TSL-FUNC-0130*** The TSL shall provide an API to retrieve the current power status of each heater.  ***STCS-SRS-TSL-FUNC-0140*** The TSL shall provide an API to set the status of a given heater to ON.  ***STCS-SRS-TSL-FUNC-0150*** The TSL shall provide an API to set the status of a given heater to OFF. | | | |
| **Pass/Fail Criteria** | *The test fails if the APIs don't retrieve the current temperature of each thermistor, the current power status of each heater or set the status of a given heater.* | | | |
| **Test Step** | **Test Step Description** | **Expected Result** | **Test Result** | **Pass/Fail** |
| **001** | Check the API that retrieves the current temperature of the thermistor | Should receive a number |  |  |
| **002** | Check the API that retrieves the status of the heater | Should receive an ON or OFF indication |  |  |
| **003** | Attempt to change the heater status through the provided API | Heater status should change accordingly |  |  |
| **Test Status** |  | | | |
| **Date** |  | | | |
| **Test Conductor** |  | | | |

### TC-STL-0070 – Log Simulation Data

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | ***TC-STL-0070*** | | | |
| **Verification Method** | Inspection | | | |
| **Objective** | Ensure that the TSL maintains a detailed log of all temperature data generated during the simulation. | | | |
| **Procedure Description** | Inspect if the code output is a log of all temperature data generated during the simulation, including timestamps (monotonic clock), environmental conditions changes, heater status changes, thermistor temperatures and errors. | | | |
| **Requirement IDs** | ***STCS-SRS-TSL-FUNC-0160*** The TSL must maintain a detailed log of all temperature data generated during the simulation, including timestamps (monotonic clock), environmental conditions changes, heater status changes, thermistor temperatures and errors. | | | |
| **Pass/Fail Criteria** | *The test fails if* the TSL doesn’t maintain a detailed log of all temperature data generated during the simulation. | | | |
| **Test Step** | **Test Step Description** | **Expected Result** | **Test Result** | **Pass/Fail** |
| **001** | Run the program and check the log | Should receive a detailed log with timestamps (monotonic clock), environmental conditions changes, heater status changes, thermistor temperatures and errors. |  |  |
| **Test Status** |  | | | |
| **Date** |  | | | |
| **Test Conductor** |  | | | |

### TC-STL-0080 – Error Handling

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | ***TC-STL-0080*** | | | |
| **Verification Method** | Test | | | |
| **Objective** | Ensure that the TSL handles errors gracefully. | | | |
| **Procedure Description** | Create errors and check that the error message is displayed, the error is logged, and the user is notified. | | | |
| **Requirement IDs** | ***STCS-SRS-TSL-FUNC-0170*** The TSL shall handle errors gracefully by providing appropriate error messages for invalid API calls, corrupted data, or any other unexpected conditions. The system should log these errors and notify the user. | | | |
| **Pass/Fail Criteria** | *The test fails if the API fails to display an error message, log the error or notify the user when an error occurs.* | | | |
| **Test Step** | **Test Step Description** | **Expected Result** | **Test Result** | **Pass/Fail** |
| **001** | Create an invalid API call | The TSL should display the appropriate error message, logging it and notifying the user |  |  |
| **002** |  |  |  |  |
| **003** |  |  |  |  |
| **Test Status** |  | | | |
| **Date** |  | | | |
| **Test Conductor** |  | | | |

## Usage SceNario 2 – THERMAL CONTROL FUNCTION

### TC-TCF-0010 – PID Implementation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | ***TC-TCF-0010*** | | | |
| **Verification Method** | Inspection | | | |
| **Objective** | Ensure the implementation of a software PID controller for each thermistor. | | | |
| **Procedure Description** | Check if for each thermistor the control management function defined is a software PID controller. | | | |
| **Requirement IDs** | ***STCS-SRS-TCF-FUNC-0180*** The TCF shall implement a Thermal Control Management Function to perform the thermal control based on a software Proportional-Integral-Derivative (PID) controller for each thermistor. | | | |
| **Pass/Fail Criteria** | *The test fails if for any of the thermistors, the control management function is not a PID controller* | | | |
| **Test Step** | **Test Step Description** | **Expected Result** | **Test Result** | **Pass/Fail** |
| **001** | For each thermistor, check the thermal control management function | It should be based on a software PID controller |  |  |
| **Test Status** |  | | | |
| **Date** |  | | | |
| **Test Conductor** |  | | | |

### TC-TCF-0020 – PID Parameters and Setpoints

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | ***TC-TCF-0020*** | | | |
| **Verification Method** | Test | | | |
| **Objective** | Test the user command line to change PID controller parameters and the setpoint temperatures. | | | |
| **Procedure Description** | Open the command line and try to change the PID controller parameters and the setpoint temperatures. | | | |
| **Requirement IDs** | ***STCS-SRS-TCF-FUNC-0190*** The TCF shall allow the user to set the PID controller parameters (proportional, integral, and derivative gains) and the setpoint temperatures for the thermistors before the start of execution. This configuration should include Proportional gain (Kp), Integral gain (Ki), Derivative gain (Kd).  ***STCS-SRS-TCF-FUNC-0210*** The TCF shall allow a user to change the setpoint temperature for a specific thermistor or all thermistors at runtime between the values of –20 and 20. | | | |
| **Pass/Fail Criteria** | *The test fails if the user can't either change the PID controller parameters nor the setpoint temperatures, or if it is allowed to set a setpoint temperature outside the –20 and 20 interval* | | | |
| **Test Step** | **Test Step Description** | **Expected Result** | **Test Result** | **Pass/Fail** |
| **001** | Try changing each of the PID controller parameters. | The user can change the parameters. |  |  |
| **002** | Try changing the setpoint temperature between –20 and 20, for 1 or all thermistors. | The user can change the setpoints. |  |  |
| **003** | Try changing the setpoint temperature outside of the range –20:20. | The user can’t change the setpoints. |  |  |
| **Test Status** |  | | | |
| **Date** |  | | | |
| **Test Conductor** |  | | | |

### TC-TCF-0030 – Setpoint Temperatures Init

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | ***TC-TCF-0030*** | | | |
| **Verification Method** | Inspection | | | |
| **Objective** | Test that the setpoint temperature for all thermistors is initialized, default value is 0. | | | |
| **Procedure Description** | Look at the code and find the setpoint temperatures for all thermistors, check that the default value for all is 0. | | | |
| **Requirement IDs** | ***STCS-SRS-TCF-FUNC-0200*** The TCF shall define a setpoint temperature for each thermistor. The default setpoint shall be 0 for all thermistors. | | | |
| **Pass/Fail Criteria** | *The test fails if there are no setpoint temperatures or the default value is not 0.* | | | |
| **Test Step** | **Test Step Description** | **Expected Result** | **Test Result** | **Pass/Fail** |
| **001** | Check that the setpoint temperatures is defined for each thermistor. | It is defined |  |  |
| **002** | Check that the default value for the setpoint temperatures is 0 for all thermistors. | The default value is 0. |  |  |
| **Test Status** |  | | | |
| **Date** |  | | | |
| **Test Conductor** |  | | | |

### TC-TCF-0040 – Enable/Disable Functionality

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | ***TC-TCF-0040*** | | | |
| **Verification Method** | Inspect | | | |
| **Objective** | Ensure that the interface to enable and disable the TCF functionality is correctly implemented. | | | |
| **Procedure Description** | Inspect the code to check that TCF calls the necessary methods to start the TCF functionality when it receives an enable message and calls the necessary methods to stop the TCF functionality when it receives a disabled message. | | | |
| **Requirement IDs** | ***STCS-SRS-TCF-FUNC-0220*** The TCF must provide an interface to enable its functionality. Upon enabling, the TCF should start reading temperature values from the TSL, feed them into the PID controller, and adjust the heater power status accordingly.  ***STCS-SRS-TCF-FUNC-0230*** The TCF must provide an interface to disable its functionality. Upon disabling, the TCF should stop reading temperature values and cease controlling the heater power status, ensuring all heaters are switched off. | | | |
| **Pass/Fail Criteria** | *The test fails if the TCF functionality is not correctly enabled or disabled following the interface commands.* | | | |
| **Test Step** | **Test Step Description** | **Expected Result** | **Test Result** | **Pass/Fail** |
| **001** | Check that there is an interface to enable and disable the TCF functionality | There is a working interface |  |  |
| **002** | Check that when TCF is enabled, the necessary methods are called | The TCF calls methods to read the temperatures of the thermistors from the TSL, feeds them into the PID controller and adjusts the heater power status |  |  |
| **003** | Check that when TCF is disabled, the necessary methods are called | The TCF stops reading the temperature values and stops controlling the heaters, ensuring all of them are switched off |  |  |
| **Test Status** |  | | | |
| **Date** |  | | | |
| **Test Conductor** |  | | | |

### TC-TCF-0050 – Data Acquisition

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | ***TC-TCF-0050*** | | | |
| **Verification Method** | Inspection | | | |
| **Objective** | Ensure the TCF, at the start of execution, acquire the power status of the heaters and, every 5 Hz, acquire the values of the thermistors. | | | |
| **Procedure Description** | Check if the TCF, at the start of execution, acquire the power status of the following heaters: HTR-01, HTR-02, HTR-03, HTR-04; the TCF, every 5 Hz, acquire the values of the following thermistors: THERM-01, THERM-02, THERM-03, THERM-04. | | | |
| **Requirement IDs** | ***STCS-SRS-TCF-FUNC-0240*** The TCF shall, at the start of execution, acquire the power status of the following heaters: HTR-01, HTR-02, HTR-03, HTR-04.  ***STCS-SRS-TCF-FUNC-0250*** The TCF shall, every 5 Hz, acquire the values of the following thermistors: THERM-01, THERM-02, THERM-03, THERM-04. | | | |
| **Pass/Fail Criteria** | *The test fails if t*he TCF, at the start of execution, doesn’t acquire the power status of an heaters or, every 5 Hz, doesn’t acquire the values of a thermistors. | | | |
| **Test Step** | **Test Step Description** | **Expected Result** | **Test Result** | **Pass/Fail** |
| **001** | For each heater, check if the TCF acquire the power status, at the start of execution | The TCF should acquire the power status for each heater, at the start of the execution |  |  |
| **002** | For each thermistor, check if the TCF acquire the values, every simulation cycle | The TCF should acquire the values for each thermistor, every 5 Hz |  |  |
| **Test Status** |  | | | |
| **Date** |  | | | |
| **Test Conductor** |  | | | |

### TC-TCF-0060 – Update Data Reading Frequency

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | ***TC-TCF-0060*** | | | |
| **Verification Method** | Test | | | |
| **Objective** | Test the user command line to change the data reading frequency | | | |
| **Procedure Description** | Open the command line and try to change the data reading frequency | | | |
| **Requirement IDs** | ***STCS-SRS-TCF-FUNC-0260*** The TCF shall allow the user to update, at runtime, the frequency at which the thermistors data is read from the TSL-provided interface between the values of 1Hz to 5Hz. | | | |
| **Pass/Fail Criteria** | *The test fails if the user can't change the data reading frequency or if it is allowed to set outside the range of 1Hz and 5Hz* | | | |
| **Test Step** | **Test Step Description** | **Expected Result** | **Test Result** | **Pass/Fail** |
| **001** | Try changing the data reading frequency. | The user can change it. |  |  |
| **002** | Try changing the data reading frequency outside of the range 1 to 5Hz. | The user can’t change the freqeuncy. |  |  |
| **Test Status** |  | | | |
| **Date** |  | | | |
| **Test Conductor** |  | | | |

### TC-TCF-0070 – Heater Operation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | ***TC-TCF-0070*** | | | |
| **Verification Method** | Inspection | | | |
| **Objective** | Check if the TCF is operating the heaters via the TSF-provided interface based on the PID controller data. | | | |
| **Procedure Description** | Inspect the code to check that the TCF is taking the output values from the PID controller and then using the TSF-provided interface to change the power heater status accordingly. | | | |
| **Requirement IDs** | ***STCS-SRS-TCF-FUNC-0270*** The TCF shall operate the heaters via the TSF-provided interface based one the output of the PID controller. | | | |
| **Pass/Fail Criteria** | *The test fails if the TCF does not operate the heaters via the provided interface by the TSF or doesn’t do it based on the PID controller.* | | | |
| **Test Step** | **Test Step Description** | **Expected Result** | **Test Result** | **Pass/Fail** |
| **001** | Inspect the code to check that the TCF is reading the output values from the PID controller | The TCF is keeping track of the PID controller output |  |  |
| **002** | Check if the TCF changes the power heater status through the provided TSL-interface | The TCF takes the data from the controller and updated the heater status accordingly through the provided TSL-interface |  |  |
| **Test Status** |  | | | |
| **Date** |  | | | |
| **Test Conductor** |  | | | |

### TC-TCF-0080 – Temperature Maintenance

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | ***TC-TCF-0080*** | | | |
| **Verification Method** | Test | | | |
| **Objective** | Check if the TCF is adjusting the power status of the heaters to match the setpoint temperature of each thermistor. | | | |
| **Procedure Description** | Make API calls to get the current temperatures of all the thermistors. Once one API call is made, check if the actual temperature follows the setpoint one. | | | |
| **Requirement IDs** | ***STCS-SRS-TCF-FUNC-0280*** The TCF must adjust the heater power status to maintain the temperature of each thermistor at the defined setpoint value. | | | |
| **Pass/Fail Criteria** | *The test fails if the TCF does not adjust the power status to maintain the temperature of each thermistor.* | | | |
| **Test Step** | **Test Step Description** | **Expected Result** | **Test Result** | **Pass/Fail** |
| **001** | Make API calls to get the temperature of the thermistors over a certain period. | The data should arrive. |  |  |
| **002** | Check the setpoint temperature and the actual temperature of each thermistor | The actual temperature should try to follow the setpoint one. |  |  |
| **Test Status** |  | | | |
| **Date** |  | | | |
| **Test Conductor** |  | | | |

## USAGE SCENARIO 3 – Visualization User Interface

### TC-VUI-0010 - Acquire Thermistors Temperatures and Heater Status

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | ***TC-VU-0010*** | | | |
| **Verification Method** | Inspect | | | |
| **Objective** | Ensure that the user interface acquires and displays input data from CSV file. | | | |
| **Procedure Description** | Inspect the code to check that VUI extracts the thermistors temperature values and the heater power status from the CSV files sent by the TSL. | | | |
| **Requirement IDs** | ***STCS-SRS-VUI-FUNC-0280***  The VUI shall acquire the following data from the CSV file: Thermistors Temperature Values and Heater Power Status. | | | |
| **Pass/Fail Criteria** | *The test fails if the VUI program is unable to parse the data from the CSV data file.* | | | |
| **Test Step** | **Test Step Description** | **Expected Result** | **Test Result** | **Pass/Fail** |
| **001** | Check that there is a valid CSV file feeding the program. | There's a CSV file with the data corresponding to system’s information. |  |  |
| **002** | Check that when VUI is running it can parse the thermistors temperature values and the heater power status from the CSV file. | The VUI program can parse and store the information in some variable. |  |  |
| **Test Status** |  | | | |
| **Date** |  | | | |
| **Test Conductor** |  | | | |

### TC-VUI-0020 - Data Acquisition Frequency

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | ***TC-VU-0020*** | | | |
| **Verification Method** | Inspection | | | |
| **Objective** | Ensure that the user interface has a default frequency for updating data and allows the user adjustments | | | |
| **Procedure Description** | Inspect the code to check that VUI extracts the thermistors temperature values and the heater power status from the CSV files sent by the TSL. | | | |
| **Requirement IDs** | ***STCS-SRS-VUI-FUNC-0290***  The data acquisition performed by the VUI shall be done at a frequency of 0.2 seconds by default.  ***STCS-SRS-VUI-FUNC-0300***  The VUI shall allow the user to adjust the data acquisition frequency between 0.2 seconds and 1 second. | | | |
| **Pass/Fail Criteria** | *The test fails if the VUI program is not fetching data from the CSV files at a rate of 0.2 seconds by default and/or the user interface doesn’t allow the user to change the data acquisition rate between 0.2 and 1 seconds.* | | | |
| **Test Step** | **Test Step Description** | **Expected Result** | **Test Result** | **Pass/Fail** |
| **001** | Search the code to check that the VUI is performing the data acquisition each 0.2 seconds by default. | The data acquisition is being performed every 0.2 seconds. |  |  |
| **002** | Check if the user interface allows the user to change the data acquisition rate between 0.2 and 1 seconds. | The interface allows the user to alter the data acquisition frequency from 0.2 to 1 second. |  |  |
| **Test Status** |  | | | |
| **Date** |  | | | |
| **Test Conductor** |  | | | |

### TC-VUI-0030 – Display Thermistor Temperatures and Heater Status

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | ***TC-VU-0030*** | | | |
| **Verification Method** | Inspection | | | |
| **Objective** | Ensure that the user interface display the latest temperature values for the thermistors and the current power status of the heaters. | | | |
| **Procedure Description** | Inspect that the VUI display the latest temperature values for all 4 thermistors and the current power status of all 4 heaters. | | | |
| **Requirement IDs** | ***STCS-SRS-VUI-FUNC-0310*** The VUI shall display the latest temperature value for all 4 thermistors.  ***STCS-SRS-VUI-FUNC-0320*** The VUI shall display the current power status of all 4 heaters. | | | |
| **Pass/Fail Criteria** | *The test fails if the VUI doesn’t* display the latest temperature values for the thermistors or the current power status of the heaters. | | | |
| **Test Step** | **Test Step Description** | **Expected Result** | **Test Result** | **Pass/Fail** |
| **001** | Check if the user interface displays the latest temperature value for all 4 thermistors. | There’s a section in the user interface displaying the latest temperature value for all 4 thermistors. |  |  |
| **002** | Check if the user interface displays the current power status of all 4 heaters. | There’s a section in the user interface displaying the current power status of all 4 heaters. |  |  |
| **Test Status** |  | | | |
| **Date** |  | | | |
| **Test Conductor** |  | | | |

### TC-VUI-0040 - Thermistor Temperatures and Heater Status Plots

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | ***TC-VU-0040*** | | | |
| **Verification Method** | Inspection | | | |
| **Objective** | Ensure that the VUI displays plots with the acquired data for the thermistor temperature and heater status from the past 5 seconds. | | | |
| **Procedure Description** | Check the VUI for the plots and ensure that the default time range is 5 seconds for each one. | | | |
| **Requirement IDs** | ***STCS-SRS-VUI-FUNC-0330*** The VUI shall display a plot with the acquired data for each thermistor temperature value from the past 5 seconds.  ***STCS-SRS-VUI-FUNC-0340***  The VUI shall display a plot with the historical power status of each heater from the past 5 seconds. | | | |
| **Pass/Fail Criteria** | *The test fails if one of the plots is not displayed or the time range is incorrect.* | | | |
| **Test Step** | **Test Step Description** | **Expected Result** | **Test Result** | **Pass/Fail** |
| **001** | Check for the thermistor temperature plot | It should be displayed with a time range of 5 seconds |  |  |
| **002** | Check for the heater status plot | It should be displayed with a time range of 5 seconds |  |  |
| **Test Status** |  | | | |
| **Date** |  | | | |
| **Test Conductor** |  | | | |

### TC-VUI-0050 - Time Plot Range Adjustment

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | ***TC-VU-0050*** | | | |
| **Verification Method** | Inspection | | | |
| **Objective** | Ensure that the VUI allows user to define the time range for the available plots and synchronize them accordingly. | | | |
| **Procedure Description** | Check if the if there is a working time range modifier available for the available plots in user interface and if the changes are synchronized between plots. | | | |
| **Requirement IDs** | ***STCS-SRS-VUI-FUNC-0350*** The VUI shall allow the user to individually change the time range of the available plots between the values of 1 second and 75 seconds.  ***STCS-SRS-VUI-FUNC-0360*** The VUI shall allow the user to synchronize the time range for the two available plots (i.e., when changing the time range for one plot the same time range should be applied for the other plot). | | | |
| **Pass/Fail Criteria** | *The test fails if the user is not able to change the time range of both plots independently, between the ranges of 1 second and 75 seconds, and/or can’t synchronize the time range of both plots so that the modification of this property in one plot implies the same change in the other plot.* | | | |
| **Test Step** | **Test Step Description** | **Expected Result** | **Test Result** | **Pass/Fail** |
| **001** | Check if the user interface allows the user to change the time range of each plot (thermistor plot and heater plot) independently between the ranges f 1 second and 75 seconds. | The user can change the time range of each plot independently between the ranges of 1 second and 75 seconds. |  |  |
| **002** | Check if the user interface allows the user to synchronize the time range of each plot (thermistor plot and heater plot) so that when one is changed the other follows. | The user can synchronize the time range of the plots and when he changes one the other follows the alteration. |  |  |
| **Test Status** |  | | | |
| **Date** |  | | | |
| **Test Conductor** |  | | | |

### TC-VUI-0060 – Display Error Messages

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | ***TC-VU-0060*** | | | |
| **Verification Method** | Inspection | | | |
| **Objective** | Ensure that the VUI displays a panel with logged error messages from the TSL. | | | |
| **Procedure Description** | Check the VUI for the panel with error messages and ensure they are being read from the TSL log file. | | | |
| **Requirement IDs** | ***STCS-SRS-VUI-FUNC-0370*** The VUI shall display a panel with logged error messages from the TSL. | | | |
| **Pass/Fail Criteria** | *The test fails if the error message panel is not present or if it is not displaying the messages from the log.* | | | |
| **Test Step** | **Test Step Description** | **Expected Result** | **Test Result** | **Pass/Fail** |
| **001** | Check for the error message panel | It should be displayed |  |  |
| **002** | Check if it is displaying messages from the read TSL log file | The messages should come from the TSL log file |  |  |
| **Test Status** |  | | | |
| **Date** |  | | | |
| **Test Conductor** |  | | | |

### TC-VUI-0070 – Error Handling

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | ***TC-VU-0070*** | | | |
| **Verification Method** | Test | | | |
| **Objective** | Ensure that the VUI handles errors gracefully. | | | |
| **Procedure Description** | Check if the VIU displays appropriate error messages in case of missing data, corrupted CSV files, or issues during data acquisition. | | | |
| **Requirement IDs** | ***STCS-SRS-VUI-FUNC-0380*** The VUI shall handle errors gracefully, displaying appropriate error messages in case of missing data, corrupted CSV files, or issues during data acquisition. | | | |
| **Pass/Fail Criteria** | *The test fails if the VIU doesn’t handle errors gracefully.* | | | |
| **Test Step** | **Test Step Description** | **Expected Result** | **Test Result** | **Pass/Fail** |
| **001** | Forçar erros |  |  |  |
| **Test Status** |  | | | |
| **Date** |  | | | |
| **Test Conductor** |  | | | |

### **ANNEXES**

1. Test Specification table format

This is the template that needs to be followed when creating a new test specification table:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | **TC-<MODULE ACRONYM>-<XXXX>** | | | |
| **Verification Method** | <Test/Inspection> | | | |
| **Objective** | <The feature that this test aims to check (e.g., “Verify that…)> | | | |
| **Procedure Description** | *<Explain how we can execute the verification of the requirement in case.>* | | | |
| **Requirement IDs** | **<ID of Requirement being tested>** <Requirement description> | | | |
| **Pass/Fail Criteria** | *<What is the pass/fail criteria applied for this test? What makes the test considered a success as a whole?>* | | | |
| **Test Step** | **Test Step Description** | **Expected Result** | **Test Result** | **Pass/Fail** |
| **001** | <Singular step to be executed #1> | <What is expected to happen when executing the specified in the test step 001 description column> | <What actually happened when executing the specified in the test step description column> | <Pass/Fail> |
| **002** | <Singular step to be executed #2> | <What is expected to happen when executing the specified in the test step 002 description column> | <What actually happened when executing the specified in the test step 002 description column> | <Pass/Fail> |
| **Test Status** | **<Passed/Failed/Qualified Pass>** | | | |
| **Date** | <YYYY-MM-DD> | | | |
| **Test Conductor** | <Your Name> | | | |

Uma imagem com texto, captura de ecrã, Tipo de letra, encarnado

Descrição gerada automaticamente