System Test Plan

For

Ingenion Telemetry Web Server

Version/Author	Date
1.0/Ryan Flinchum	10/18/2023
1.1/Ryan Flinchum & Jack Capuano	12/7/2023
2.0/Ryan Flinchum	02/25/2024
3.0/Jack Capuano	4/7/2024

Table of Contents

1. Introduction 1
    1.1 Purpose 2
    1.2 Objectives 2
2. Functional Scope 2
3. Overall Strategy and Approach 2
    3.1 Testing Strategy 2
    3.2 System Testing Entrance Criteria 3
    3.3 Testing Types 4
    3.4 Suspension Criteria and Resumption Requirements 5
4. Execution Plan 6
    4.1 Execution Plan 6
5. Traceability Matrix & Defect Tracking 9
    5.1 Traceability Matrix 9
    5.2 Defect Severity Definitions 12
6. Environment 12
    6.1 Environment 12
7. Assumptions 13
8. Risks and Contingencies 14

1. Introduction

1.1 Purpose

This document is a test plan for Ingenion Telemetry Web Server Testing, produced by the Telemetry Testing team. It outlines the testing strategy and approach the team will adopt to ensure that the telemetry web server aligns with the set business requirements before deployment.

1.2 Objectives

- Adherence to requirements, specifications, and telemetry protocols.
- Facilitates the intended telemetry functions and maintains requisite standards.
- Satisfies the Entrance Criteria for System Deployment.
- Upholds internal security benchmarks.
- Ensures compliance with all relevant data regulations.

2. Functional Scope

The Modules in the scope of testing for the Ingenion Telemetry Webserver are detailed in the following documents:

1. Telemetry Requirements Specification document:
   System Requirements Specification.docx

3. Overall Strategy and Approach

3.1 Testing Strategy

Ingenion Telemetry Web Server System Testing will include testing of all functionalities that are in scope (Refer Functional Scope Section) identified. System testing activities will include the testing of new functionalities, modified functionalities, screen level validations, work flows, functionality access, testing of internal & external interfaces.

3.1.1 Usability Testing

Test Objective: Ensure the web server provides an intuitive user experience with clarity, logical navigation, and readability.

Technique: Users will navigate the interface, ensuring attributes like access keys, tab order, and font readability align with usability standards.

Completion Criteria: All interface elements should be assessed for usability, and any deviations from the standard usability practices should be addressed.

Special Consideration: Access to the Ingenion Telemetry Webserver and the corresponding System Requirements Specification document.

3.1.2 Functional Testing

Test Objective: Verify that the system's data entry, retrieval, processing, and command transmission are congruent with the specified requirements in the SRS.

Technique: Implement use cases from the use case diagram, where valid data yields the expected outcome, and invalid data results in appropriate warning messages.

Completion Criteria: All functional use cases have been executed, and all defects have been rectified.

Special Consideration: Emphasis on validating comprehensive logs for communication events, errors, and warnings to facilitate system analysis and debugging.

3.1.3 Hardware and Software Integration Testing

Test Objective: Validate the harmonious integration between the web server software component, FreeRTOS, and the hardware interfaces.

Technique: Simulate data processing, transmission, and command actions, checking the system's adherence to the IEEE 802.3 standards and real-time data communication via WebSockets.

Completion Criteria: Seamless interaction between the software and hardware components without data loss or communication issues.

3.2 System Testing Entrance Criteria

To commence system testing, the following criteria must be satisfied:

1. Readiness of Test Environment: Ensure that the test environment mirrors the production environment, complete with the necessary hardware and software interfaces.
2. Availability of Test Data: Suitable data sets for telemetry should be available or generated to simulate real-world scenarios.
3. Test Case Preparation: Comprehensive test cases that cover all the functionalities mentioned in the SRS should be ready for execution.
4. Stable Build: A stable build/version of the Ingenion Telemetry Web Server System without critical defects should be available for testing.
5. Documentation: Updated SRS and other relevant documentation must be readily

available to the testing team.

3.3 Testing Types

3.3.1 Usability Testing

The focus here is on the end-user experience, ensuring clarity, readability, and ease of navigation throughout the interactive web server.

System Requirements Specification, 3.4.1: "The web server's interface will be checked for intuitive design, emphasizing clarity, readability, and logical navigation."

3.3.2 Hardware and Software Integration Testing

Objective: Validate the harmonious integration between the web server software component, FreeRTOS, and the hardware interfaces.

Technique: Simulate data processing, transmission, and command actions, checking the system's adherence to the IEEE 802.3 standards and real-time data communication via WebSockets.

Completion Criteria: Seamless interaction between the software and hardware components without data loss or communication issues.

System Requirements Specification, 3.3.1: “The system must run FreeRTOS on the MicroBlaze CPU, with performance metrics meeting customer defined benchmarks.”

System Requirements Specification, 3.3.2: “The system must utilize FreeRTOS for task scheduling, with task delays of less than 1ms.”

System Requirements Specification, 3.3.3: “TThe system shall have a web server software component.”

System Requirements Specification, 3.3.4: “The system’s web server software component shall be compatible with FreeRTOS.”

System Requirements Specification, 3.3.5: “The web server software must interface with FreeRTOS to retrieve telemetry data, demonstrating less than 1ms response times.”

System Requirements Specification, 3.3.6: “he system must handle TCP/IP communication over Ethernet, with Cat 5 cable standards of 10Mbps down and 100Mbps up.”

System Requirements Specification, REQ-3.3.7: The system must gather the telemetry data from the AXI bus, sampling at the standard 400MHz

System Requirements Specification, REQ-3.3.8: The system must process the telemetry data from the AXI bus, processing at a rate of 40ns per transaction (8 bytes) of the bus

System Requirements Specification, REQ-3.3.9: The system must gather telemetry data from peripherals, updating within 50ms of change

System Requirements Specification, REQ-3.3.9: The system must process telemetry data from peripherals, processing within 50ms of data being gathered

System Requirements Specification, REQ-3.3.10: The system must display telemetry data in real-time, with a defined maximum latency of 100ms (50 for gathering, 50 for processing)

3.3.3 Functional Testing

The objective of this test is to ensure that each element of the Ingenion telemetry web server component meets the functional requirements as specified in:

- External Interface Requirements
- Business / Functional Requirements
- Any additional functional documents generated during the project, such as issue resolutions, change requests, or feedback.

System Requirements Specification, 3.1.1: “The system shall host a web server via the fgpa’s ethernet connection.”

System Requirements Specification, 3.1.2: “The system’s web server shall allow the user to read telemetry from the fgpa’s onboard components”

System Requirements Specification, 3.1.3: “The system’s web server shall allow the user to interact with the fpga’s onboard components.”

System Requirements Specification, 3.1.4: “The system shall be able to receive commands to the CPU over ethernet, sent by the user.”

3.4 Suspension Criteria and Resumption Requirements

3.4.1 Suspension Criteria

Testing may be suspended if:

- Critical defects are discovered that halt major functionalities.
- Hardware or network failures that prevent effective system testing.

- Significant discrepancies between the test environment and the production environment that could skew test results.
- Any changes made to the hardware, software, or database during the testing phase.

3.4.2 Resumption Requirements

Testing can be resumed when:

- Critical defects have been addressed and resolved.
- Hardware or network issues are rectified.
- Test environments are realigned to match the production environment.
- Changes made during the testing halt are validated, and the system is deemed stable for testing.

4. Execution Plan

4.1 Execution Plan

The execution plan will detail the test cases to be executed. The Execution plan will be put together to ensure that all the requirements are covered. The execution plan will be designed to accommodate some changes if necessary, if testing is incomplete on any day. All the test cases of the projects under test in this release are arranged in a logical order depending upon their inter dependency.

Requirement (From SRS)	Test Case #	Input	Expected Behavior	Pass/Fail
REQ-3.1.1 **REQ-3.3.6** **REQ-3.4.2**	1	User accesses the interactive web server interface	The web server displays the Home Page with buttons that take you to the other pages	PASS
REQ-3.1.1	2	User accesses the Demo Control page	The web server displays the Demo Control Page with a control panel	PASS
REQ-3.1.1	3	User accesses one of the 5 telemetry pages from the Demo Control page	The web server displays the selected telemetry pages	PASS
REQ-3.1.2	4	User accesses the CoreMark telemetry page and requests a coremark benchmark be performed	The system interfaces with the DDR3 memory, showing performance of the CPU in MHz	FAIL
REQ-3.1.3 **REQ-3.2.5** **REQ-3.3.9** **REQ-3.3.10**	5	User accesses the FPU telemetry page and requests a calculation by the FPGA	The system takes the imputed calculation request and returns the output of the FPU	PASS
REQ-3.1.3 **REQ-3.2.6**	6	User accesses the LED operations telemetry page	The system's LEDS respond to the user's request for	PASS

REQ-3.3.9 **REQ-5.1.1**		and requests the LEDS to be colored in a certain way, or off	coloration or power off.	
REQ-3.1.3 **REQ-3.2.2** **REQ-3.2.3** **REQ-3.2.7** **REQ-3.2.8** **REQ-3.3.5** **REQ-3.3.7** **REQ-3.3.8**	7	User accesses the AXI interface telemetry page and requests a value from a inputted memory address	The system displays the value from the inputted memory address on the webserver	PASS
REQ-3.2.1 **REQ-3.4.2** **REQ-3.4.3** **REQ-3.4.4**	8	User sends any command or request over ethernet	The ethernet request/response is analyzed by external speed test tools (ex: windows ethernet interface page)	PASS
REQ-5.1.2 **REQ-5.1.4** **REQ-5.1.5** **REQ-5.1.6** **REQ-5.1.7** **REQ-5.4.4**	9	User sends any command or request to the system	The system tracks and outputs the various delays and processing times produced by subsystems	PASS
REQ-5.2.1 **REQ-5.2.2**	10	User simulates an error in the system	System monitors, catches, reports and deals with	PASS

REQ-5.2.3 **REQ-5.2.4** **REQ-5.2.5**			the simulated error (IEEE 829)	
REQ-5.3.1 **REQ-5.3.2** **REQ-5.4.3**	11	User sends any command or request over ethernet that would compromise data integrity	The system responds to the request/command in a way to restore data integrity	PASS
REQ-5.3.3 **REQ-5.3.4**	12	Any data changes within the system	Data is encrypted in backup-protected storage	PASS
REQ-5.4.1 **REQ-5.4.2**	13	The user uses the system in any way (idle, working, etc)	The system achieves an MTBF and MTTR defined by the requirement	PASS
REQ-3.3.1 **REQ-3.3.2** **REQ-3.3.3** **REQ-3.3.4**	14	User programs the bitstream and runs the configuration	The system's web server component runs on FreeRTOS and is accessible	PASS
REQ-3.4.1	15	User interacts with any webserver page	System complies with HTTP protocols when responding	PASS

5. Traceability Matrix & Defect Tracking

5.1 Traceability Matrix

REQ ID	Test Case ID	Criticality
REQ-3.1.1 **REQ-3.3.6** **REQ-3.4.2**	1	**Medium**
REQ-3.1.1	2	**Medium**
REQ-3.1.1	3	**Medium**
REQ-3.1.2	4	**Medium**
REQ-3.1.3 **REQ-3.2.5** **REQ-3.3.9** **REQ-3.3.10**	5	**Medium**
REQ-3.1.3 **REQ-3.2.6** **REQ-3.3.9** **REQ-5.1.1**	6	**Medium**
REQ-3.1.3 **REQ-3.2.2** **REQ-3.2.3** **REQ-3.2.7** **REQ-3.2.8**	7	**Medium**

REQ-3.3.5 **REQ-3.3.7** **REQ-3.3.8**		
REQ-3.2.1 **REQ-3.4.2** **REQ-3.4.3** **REQ-3.4.4**	8	**Low**
REQ-5.1.2 **REQ-5.1.4** **REQ-5.1.5** **REQ-5.1.6** **REQ-5.1.7** **REQ-5.4.4**	9	**Low**
REQ-5.2.1 **REQ-5.2.2** **REQ-5.2.3** **REQ-5.2.4** **REQ-5.2.5**	10	**Low**
REQ-5.3.1 **REQ-5.3.2** **REQ-5.4.3**	11	**Critical**

REQ-5.3.3 **REQ-5.3.4**	12	**Critica**
REQ-5.4.1 **REQ-5.4.2**	13	**Medium**

5.2 Defect Severity Definitions

Critical	The defect causes a catastrophic or severe error that results in major problems and the functionality rendered is unavailable to the user. A manual procedure cannot be either implemented or a high effort is required to remedy the defect. Examples of a critical defect are as follows: ● System abends ● Data cannot flow through a business function/lifecycle ● Data is corrupted or cannot post to the database
Medium	The defect does not seriously impair system function can be categorized as a medium Defect. A manual procedure requiring medium effort can be implemented to remedy the defect. Examples of a medium defect are as follows: ● Form navigation is incorrect ● Field labels are not consistent with global terminology
Low	The defect is cosmetic or has little to no impact on system functionality. A manual procedure requiring low effort can be implemented to remedy the defect. Examples of a low defect are as follows: ● Repositioning of fields on screens ● Text font on reports is incorrect

6. Environment

6.1 Environment

- **Hardware Setup**:
    - A setup consisting of the necessary hardware interfaces as specified in the System Requirements Specification. This includes networking hardware compliant with IEEE 802.3 standards, the Artix 7 FPGA, and other peripherals necessary for telemetry data acquisition and processing.
- **Software Configuration**:

    - The testing environment will include the latest stable build of the Ingenion Telemetry Web Server software component, running on FreeRTOS as the operating system.
    - The web server software will be configured to interface seamlessly with the FreeRTOS operating system and the DDR3 memory in the FPGA for data storage.
- **Network Infrastructure**:
    - A reliable Ethernet connection capable of supporting 10/100 Mbps speed, ensuring consistent and uninterrupted communication between the user's computer and the telemetry web server.
- **Test Data**:
    - A comprehensive set of test data that simulates real-world telemetry scenarios. This includes various types of telemetry data inputs, command sequences, and error conditions.
- **Diagnostic and Monitoring Tools**:
    - Tools and utilities for monitoring system performance, logging activities, and diagnosing issues in real-time. This includes software for tracking network traffic, analyzing system logs, and capturing performance metrics.
- **Backup and Recovery Systems**:
    - Redundant storage and backup systems as per the specifications, ensuring data integrity and minimal data loss in case of system failures.
- **User Interface (UI) Setup**:
    - A user-friendly interface accessible via web browsers, designed for intuitive navigation, displaying telemetry data, and allowing users to send commands to the CPU.
- **Security Measures**:
    - Implemented security protocols and encryption standards to protect telemetry data during transmission and storage.
- **Controlled Environment for Reliability Testing**:
    - An environment that enables the assessment of the system's Mean Time Between Failures (MTBF) and Mean Time To Recover (MTTR), ensuring the system meets reliability and performance benchmarks.
- **Documentation and Access**:
    - Ready access to updated System Requirements Specification and other relevant documentation for reference during testing.

7. Assumptions

- **Stable Software Build**: It is assumed that a stable version of the Ingenion Telemetry Web

Server software, which has already passed basic unit and integration testing, is available for system testing.

- **Test Environment Reliability**: The test environment is presumed to accurately mimic the production environment, including hardware setups, software configurations, and network infrastructure.
- **Availability of Resources**: It is assumed that all necessary resources, including hardware, software, and human resources (testing team), are available and in optimal condition for the duration of the testing process.
- **Documentation Accuracy**: The System Requirements Specification (SRS) and related documentation provided for testing are assumed to be complete, accurate, and up-to-date.
- **Third-Party Components**: Any third-party software or hardware components integrated into the Ingenion Telemetry Web Server are assumed to function as per their specifications without introducing unforeseen issues.
- **Test Data Validity**: The test data used for testing purposes is assumed to be representative of real-world scenarios and sufficient in quantity and variety to cover all testing aspects.
- **Network Stability**: A consistent and stable network connection is assumed for all tests involving network communication and telemetry data transmission.
- **User Cooperation**: In cases of usability testing, it is assumed that users involved will provide objective and constructive feedback.
- **No Major Changes**: No significant changes to the web server's functionalities or design are assumed to be made during the testing phase.

8. Risks and Contingencies

1. **Software Instability**: Risk of encountering unstable or buggy software builds that could impede testing progress.
    - *Contingency*: Regular communication with the development team to ensure immediate resolution of critical issues.
2. **Resource Unavailability**: Unforeseen unavailability of key resources, such as hardware or personnel.
    - *Contingency*: Maintaining a buffer in resource allocation and having a plan for resource substitution.
3. **Test Environment Mismatch**: Differences between the test and production environments leading to inaccurate test results.
    - *Contingency*: Regular environment reviews and adjustments to align closely with the production setup.
4. **Documentation Gaps**: Missing or inaccurate documentation may lead to testing inefficiencies.
    - *Contingency*: Regular updates and reviews of the documentation, with feedback loops to the documentation team.
5. **Third-Party Component Failure**: Failure or malfunction of third-party components.
    - *Contingency*: Have backup components available and establish support agreements with vendors.
6. **Network Issues**: Network instability or failure affecting testing, especially for telemetry data transmission.

- *Contingency*: Implement redundant network pathways and schedule testing during low network usage periods.

7. **User Feedback Delays**: Delays in obtaining user feedback during usability testing.
   - *Contingency*: Schedule and plan user testing sessions well in advance and have backup users available.
8. **Security Breaches**: Potential security vulnerabilities in the system being tested.
   - *Contingency*: Conduct regular security audits and involve cybersecurity experts in the testing process.
9. **Changing Requirements**: Risk of changes in requirements during the testing phase.
   - *Contingency*: Establish a clear change management process and maintain regular communication with stakeholders.