System Test Plan

For

Ingenion Telemetry Web Server

Team member:

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

Table of Contents

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, 3.5.1: “The system shall rely on FreeRTOS as the operating system running on the MicroBlaze CPU.”

System Requirements Specification, 3.5.2: “The system shall use a web server software component to host the interactive web server.”

System Requirements Specification, 3.5.3: “The system’s web server software component shall interact with FreeRTOS to retrieve real-time telemetry data.”

System Requirements Specification, 3.5.4: “The system shall adhere to IEEE 802.3 standards for Ethernet communication.”

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 be connected to users via an interactive web server that will have reading and display capabilities of telemetry from

peripherals."

System Requirements Specification, 3.1.2: "The system shall be able to send commands to the CPU over Ethernet, sent by the user."

System Requirements Specification, 3.1.3: "The system shall have Ethernet capabilities in order to connect with the user's computer through an Ethernet interface at 10/100 Mbps."

System Requirements Specification, 3.1.4: "The software product shall interface with the DDR3 memory in the Artix 7 FPGA, supporting read and write operations for data storage."

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
3.1.1: The system shall be connected to users via an interactive web server that will have display capabilities of telemetry from peripherals.	1	User accesses the interactive web server interface.	The web server displays telemetry data from peripherals accurately and in real-time.	FAIL
3.1.2: The system shall be able to receive commands to the CPU over ethernet, sent by the user.	2	Execute a memory check on the DDR3 memory in the FPGA.	The system successfully interfaces with the DDR3 memory, showing correct status and memory capacity.	FAIL
3.1.3	3	Perform sequential read and write operations to the DDR3 memory.	The system completes read and write operations successfully without errors, and data integrity is maintained.	FAIL
3.1.4	4	Configure and monitor GPIO signals through the software interface.	GPIO soft cores are successfully configured and monitored, with accurate reflection of signal states.	FAIL
3.1.5	5	Run FreeRTOS on the	FreeRTOS runs efficiently on	FAIL

		MicroBlaze CPU and measure performance metrics.	the MicroBlaze CPU, meeting or exceeding predefined performance benchmarks.	
3.1.6	6	Initialize web server software component on FreeRTOS.	The web server software starts successfully and operates compatibly with FreeRTOS.	FAIL
3.7.1: The system shall be connected to users via an interactive web server that will have reading and display capabilities of telemetry from peripherals.	7	User accesses the web server through a browser and requests telemetry data.	The web server displays the telemetry data from peripherals without errors.	FAIL
3.7.2: The system shall be able to receive commands to the CPU over Ethernet, sent by the user.	8	User inputs a command via the web interface to be sent to the CPU.	The command is transmitted to the CPU over Ethernet without delay or error, and the CPU acknowledges receipt or action.	FAIL
3.7.3: The system shall have Ethernet capabilities in order to connect with	9	User connects to the system over an Ethernet connection with a speed	The connection speed is determined to be within the 10/100 Mbps range.	FAIL

the user's computer through an Ethernet interface at 10/100 Mbps.		test.		
3.7.4: The software product shall interface with the DDR3 memory in the Artix 7 FPGA, supporting read and write operations for data storage.	10	Data write and read commands are executed for the DDR3 memory in the Artix 7 FPGA.	Data is successfully written to and then read from the DDR3 memory.	FAIL
REQ-5.2.1: The system shall continuously monitor for hardware malfunctions	11	Introduce a simulated hardware malfunction.	The system detects the hardware malfunction promptly and records it for further action.	FAIL
REQ-5.2.2: The system's malfunction monitoring shall alert the user of identified issues	12	Simulate a critical hardware malfunction.	The system generates an immediate alert to the user, providing information about the identified hardware issue.	FAIL
REQ-5.2.3: The system shall implement self-diagnostic routines for	13	Trigger a self-diagnostic routine at a specified frequency.	The system successfully performs self-diagnostics , detects faults, and logs	FAIL

fault detection, specifying the frequency and scope of these diagnostics.			diagnostic results within the defined scope and frequency.	
REQ-5.2.4: The system shall eport detected abnormalities to the user, specifying reporting formats and timelines.	14	Simulate abnormal behavior in the system.	The system reports detected abnormalities to the user in the specified reporting formats and within the defined timelines.	FAIL
REQ-5.2.5: The system shall monitor for other abnormal behavior, specifying the log format, retention period, and access protocols.	15	Introduce various abnormal behaviors as specified.	The system monitors and logs other abnormal behaviors according to the specified log format, retention period, and access protocols.	FAIL
REQ-5.3.1: The system shall incorporate measures for data integrity during telemetry data acquisition, specifying allowable error rates.	16	Inject simulated telemetry data with errors beyond allowable rates.	The system identifies and rejects telemetry data with errors exceeding specified rates, ensuring data integrity.	FAIL

REQ-5.3.2: The system shall maintain data integrity during telemetry transmission, with defined security protocols and encryption standards.	17	Simulate telemetry data transmission with potential security breaches.	The system transmits telemetry data securely, following defined security protocols and encryption standards, ensuring data integrity.	FAIL
REQ-5.3.3: The system shall implement redundant storage to prevent data loss, specifying redundancy levels and recovery processes.	18	Simulate a storage failure and loss of data in one storage unit.	The system detects the failure, activates redundant storage, and recovers data based on the specified redundancy levels and recovery processes, preventing data loss.	FAIL
REQ-5.3.4: The system shall establish redundant backup systems, detailing backup frequency, storage locations, and restoration procedures.	19	Trigger a failure in the primary system and data loss.	The backup system is activated according to the defined frequency, and the system restores data from the specified storage locations, adhering to the restoration procedures, ensuring	FAIL

			minimal data loss and system continuity.	
REQ-5.4.1: The system shall achieve a defined MTBF that aligns with customer reliability expectations.	20	System running in a controlled environment for a specified period.	MTBF value calculated based on the time between failures.	FAIL
REQ-5.4.2: The system shall maintain a low MTTR with effective error logging, specifying log details and error resolution protocols.	21	Introduce a simulated error in the system.	System logs error details, and the resolution protocol is executed successfully with a low MTTR.	FAIL
REQ-5.4.3: The system shall recover gracefully from unexpected failures or errors, detailing recovery processes and maximum allowable downtime.	22	Introduce a critical failure in the system.	The system initiates recovery processes and resumes normal operation within the defined maximum allowable downtime.	FAIL
REQ-5.4.4: The system shall ensure optimal performance for processing	23	Feed the system with a set of telemetry data at or above its defined	The system processes the telemetry data within the defined performance	FAIL

telemetry data, defining performance benchmarks.		capacity.	benchmarks without significant degradation in performance.	

5. Traceability Matrix & Defect Tracking

5.1 Traceability Matrix

Requirement ID	Test Case #	Test Input	Expected Behavior
3.1.1	1	User accesses the interactive web server interface.	The web server displays telemetry data from peripherals accurately and in real-time.
3.1.2	2	Execute a memory check on the DDR3 memory in the FPGA.	The system successfully interfaces with the DDR3 memory, showing correct status and memory capacity.
3.1.3	3	Perform sequential read and write operations to the DDR3 memory.	The system completes read and write operations successfully without errors, and data integrity is maintained.
3.1.4	4	Configure and monitor GPIO signals through the software interface.	GPIO soft cores are successfully configured and monitored, with accurate reflection of signal states.
3.1.5	5	Run FreeRTOS on the MicroBlaze CPU and measure performance metrics.	FreeRTOS runs efficiently on the MicroBlaze CPU, meeting or exceeding predefined performance benchmarks.
3.1.6	6	Initialize web server software component on FreeRTOS.	The web server software starts successfully and operates compatibly

			with FreeRTOS.
3.7.1	7	User accesses the web server through a browser and requests telemetry data.	The web server displays the telemetry data from peripherals without errors.
3.7.2	8	User inputs a command via the web interface to be sent to the CPU.	The command is transmitted to the CPU over Ethernet without delay or error, and the CPU acknowledges receipt or action.
3.7.3	9	User connects to the system over an Ethernet connection with a speed test.	The connection speed is determined to be within the 10/100 Mbps range.
3.7.4	10	Data write and read commands are executed for the DDR3 memory in the Artix 7 FPGA.	Data is successfully written to and then read from the DDR3 memory.
REQ-5.2.1	11	Introduce a simulated hardware malfunction.	The system detects the hardware malfunction promptly and records it for further action.
REQ-5.2.2	12	Simulate a critical hardware malfunction.	The system generates an immediate alert to the user, providing information about the identified hardware issue.
REQ-5.2.3	13	Trigger a self-diagnostic routine at a specified frequency.	The system successfully performs self-diagnostics, detects faults, and logs diagnostic results within the defined scope and frequency.
REQ-5.2.4	14	Simulate abnormal behavior in the system.	The system reports detected abnormalities to the user in the specified reporting formats and within the

			defined timelines.
REQ-5.2.5	15	Introduce various abnormal behaviors as specified.	The system monitors and logs other abnormal behaviors according to the specified log format, retention period, and access protocols.
REQ-5.3.1	16	Inject simulated telemetry data with errors beyond allowable rates.	The system identifies and rejects telemetry data with errors exceeding specified rates, ensuring data integrity.
REQ-5.3.2	17	Simulate telemetry data transmission with potential security breaches.	The system transmits telemetry data securely, following defined security protocols and encryption standards, ensuring data integrity.
REQ-5.3.3	18	Simulate a storage failure and loss of data in one storage unit.	The system detects the failure, activates redundant storage, and recovers data based on the specified redundancy levels and recovery processes, preventing data loss.
REQ-5.3.4	19	Trigger a failure in the primary system and data loss.	The backup system is activated according to the defined frequency, and the system restores data from the specified storage locations, adhering to the restoration procedures, ensuring minimal data loss and system continuity.
REQ-5.4.1	20	System running in a controlled environment for a specified period.	MTBF value calculated based on the time between failures.

REQ-5.4.2	21	Introduce a simulated error in the system.	System logs error details, and the resolution protocol is executed successfully with a low MTTR.
REQ-5.4.3	22	Introduce a critical failure in the system.	The system initiates recovery processes and resumes normal operation within the defined maximum allowable downtime.

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.