System Design Document

For

Ingenion Telemetry Web Server

Team members:
Ryan Flinchum
Morgan Smith
Thomas Swenson
Ricky Nelson
Jack Capuano
Conrad Prisby
Hamilton Henneberg

Version/Author	Date	Changes
v1.0 / Hamilton Henneberg	9/22/23	Creation of Document Initialization of 1 - 2.3
v1.1 / Jack Capuano	9/29/23	Review & Revision
v1.2 / Thomas Swenson	9/29/23	Final Review

TABLE OF CONTENT

SYSTEM DESIGN DOCUMENT

1 INTRODUCTION

1.1 Purpose and Scope

Purpose

The System Design Document serves as a comprehensive blueprint for the development and implementation of the Ingenion Telemetry Web Server project. It aims to provide a detailed understanding of the system's architecture, functionality, and interfaces. This document will guide project stakeholders, including developers, engineers, and project managers, in the design, development, and integration of the project.

Scope

The scope of the System Design Document encompasses various aspects of the Ingenion Telemetry Web Server project. These include:

Operating Environment: It will describe the environment in which the system will operate, including hardware, software, and infrastructure.

System and Subsystem Architecture: The document will provide an architectural overview of the system, outlining its high-level structure and the relationships between subsystems.

Input Formats: The expected formats of data inputs to the system will be specified, ensuring compatibility with external data sources.

Output Layouts: This section will describe the structure and format of system-generated outputs, including telemetry data displays and user interfaces.

Human-Machine Interfaces: The document will discuss the interfaces and interactions between users and the system, highlighting user experience considerations.

Detailed Design: Detailed technical specifications, algorithms, and system components will be presented to guide development.

Processing Logic: The logic and algorithms governing system behavior and data processing will be thoroughly explained.

External Interfaces: The document will define the interactions and protocols used for communication with external systems.

1.2 Project Executive Summary

1.2.1 System Overview

This section of the System Design Document provides a high-level description of the project from a management perspective and outlines the framework within which the conceptual system design was developed.

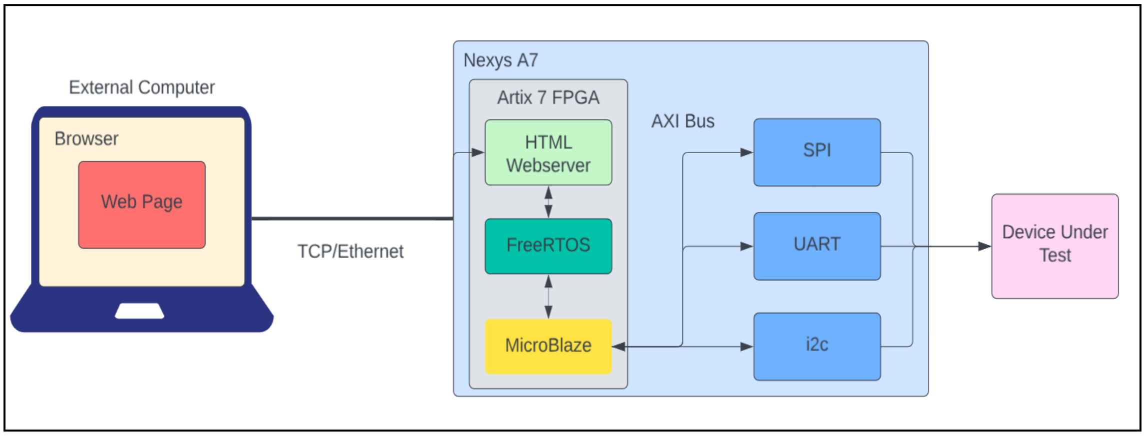


Figure 1.2.1: High-level System Architecture Diagram

The design of the Ingenion Telemetry Webserver project is framed within the context of enhancing the existing Total Verification System (TVS) used by NASA's Goddard Space Flight Center for testing and verifying satellite hardware components. The project aims to augment the TVS with a range of features and capabilities, primarily centered around the integration of a Xilinx MicroBlaze soft-core CPU on the Artix 7 FPGA and the implementation of FreeRTOS as an open-source real-time operating system. These enhancements will enable the system to host an interactive web server, establish TCP over Ethernet connections with external computers, and manage telemetry data from external, user-friendly connections.

1.2.2 Design Constraints

This section outlines the constraints and considerations that have influenced the system design. It references any trade-off analyses conducted during the design process, such as resource utilization versus productivity, and addresses any conflicts or limitations encountered in the project's development.

<u>Resource Allocation</u>: The project must operate within predefined resource constraints, including FPGA resources, memory, and processing power. Trade-offs have been considered to optimize resource utilization without compromising functionality.

<u>Compatibility</u>: Compatibility with existing systems, including the TVS and external computer systems, imposes constraints on the design to ensure seamless integration.

Time, Budget, Labor: The project operates within specific time, budget, and labor constraints, in order to complete this project within the given time period (1 semester), budget ($0), and labor (7 team members).

1.2.3 Future Contingencies

This section anticipates potential contingencies that may arise during the system design and development process. Contingencies refer to unforeseen circumstances or challenges that could alter the project's direction. It also addresses potential workarounds or alternative plans to mitigate the impact of these contingencies.

Interface Agreements: If interface agreements are difficult to establish, contingency plans will involve identifying alternative communication protocols or temporary solutions.

Architectural Stability: In the event that architectural stability issues arise with the chosen FPGA or CPU platform, the project team will explore alternative hardware options or adapt the design to mitigate instability.

Project Scale: If, during the course of the project, it is determined that the scope is either too large or too small to align with project objectives, contingency plans will be activated to scale the project accordingly. This may involve expanding or reducing features, adjusting resource allocation, or revisiting project goals to ensure alignment with project size and complexity.

2 SYSTEM ARCHITECTURE

2.1 System Hardware Architecture

In this section, we will describe the overall hardware architecture of the Ingenion Telemetry Webserver. This will include an overview of the system's hardware components and their interconnections.

The hardware architecture of the Ingenion Telemetry Webserver comprises various components essential for its operation. These components include:

Digilent Nexys A7 Development Board: Initially, the system is developed and tested on this development board, which provides the necessary hardware interfaces and connectivity options.

Xilinx Artix 7 FPGA: The FPGA on the Nexys A7, serves as the central processing unit for the system. It houses the Xilinx MicroBlaze soft-core CPU and provides programmable logic for custom functions.

MicroBlaze Soft-Core CPU: This CPU core is implemented on the FPGA and serves as the system's primary processor. It runs the FreeRTOS operating system and manages system tasks.

Ethernet Interface: The system uses an Ethernet connection for communication with external computers and data transfer.

AXI Bus: The onboard AXI bus facilitates communication with various peripherals and data sources within the FPGA.

Peripheral Devices: These may include sensors, memory modules, and other hardware elements necessary for system functionality.

2.2 System Software Architecture

In this section, we will describe the overall software architecture of the Ingenion Telemetry Webserver. This includes a breakdown of software modules, computer languages used, and any computer-aided software engineering tools employed in the development process. We will also use structured organization diagrams or object-oriented diagrams to illustrate the software architecture.

The software architecture of the Ingenion Telemetry Webserver consists of various software modules, functions, and components that collectively enable system functionality. These include:

FreeRTOS: This open-source real-time operating system runs on the MicroBlaze CPU, providing task scheduling and management.

Web Server Software: The software responsible for hosting the interactive web server, enabling user interactions and data display.

Communication Protocols: Modules for handling TCP/IP communication over Ethernet for external computer connectivity.

Telemetry Data Processing: Software components responsible for reading, processing, and displaying telemetry data from the AXI bus and peripheral devices.

User Interface: Modules for creating the human-machine interface through which users interact with the system.

2.3 Internal Communications Architecture

In this section, we will provide an overview of the internal communications architecture within the Ingenion Telemetry Webserver system. This includes describing the communication pathways and protocols used for data exchange within the system, as well as any specific architectures implemented.

Ethernet (TCP/IP): The system leverages Ethernet connectivity for high-speed data exchange between various system components. It employs the TCP/IP protocol suite to establish reliable communication channels.

AXI Bus Communication: Internally, the AXI (Advanced eXtensible Interface) bus serves as a primary communication pathway for data transfer between the MicroBlaze CPU, FPGA logic, and peripheral devices.

3 HUMAN-MACHINE INTERFACE

3.1 Inputs

This section provides a detailed design of the system and subsystem inputs, focusing on the user/operator interaction. Inputs encompass user-initiated actions, commands, and data that are processed by the system. Below are the components to be addressed in this section:

User Commands: Users interact with the system by providing commands via the web interface. These commands may include actions like starting telemetry data capture, requesting specific data, or configuring system settings.

Data Entry: Users may be required to input parameters or settings through the user interface. For example, configuring data sampling rates or specifying telemetry data sources.

3.2 Outputs

In this section, we will focus on the primary system output of the enhanced Total Verification System (TVS) project, which is the real-time telemetry display on the web server. This display serves the purpose of providing users with live telemetry data from satellite hardware components.

Contents: The real-time telemetry display presents live telemetry data from the onboard AXI bus and other peripherals in a user-friendly web interface.

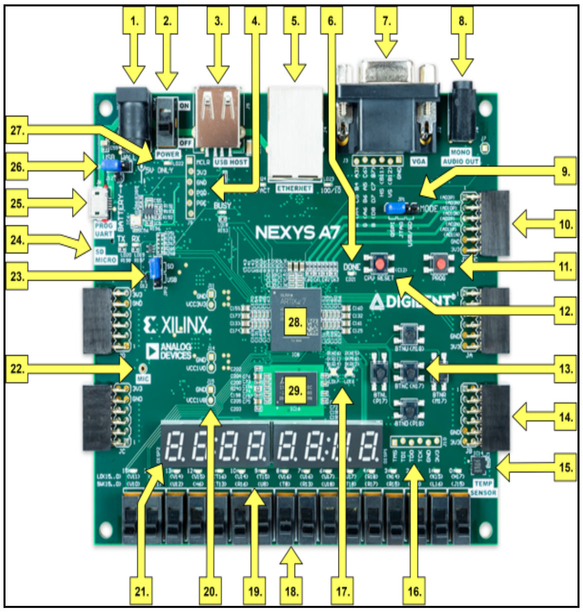
Purpose: This output is designed for operators who require continuous monitoring of telemetry data during satellite testing, enabling them to detect and respond to anomalies promptly.

4 DETAILED DESIGN

4.1 Hardware Detailed Design

The Digilent Nexys A7 board serves as the development platform for implementing the first stage of the Telemetry Webserver. Here, we provide detailed information on the hardware requirements and specifications for this critical component:

Graphical Representation:



Callout	Component Description	Callout	Component Description
1	Power jack	16	JTAG port for (optional) external cable
2	Power switch	17	Tri-color (RGB) LEDs
3	USB host connector	18	Slide switches (16)
4	PIC24 programming port (factory use)	19	LEDs (16)
5	Ethernet connector	20	Power supply test point(s)
6	FPGA programming done LED	21	Eight digit 7-seg display
7	VGA connector	22	Microphone
8	Audio connector	23	External configuration jumper (SD / USB)
9	Programming mode jumper	24	MicroSD card slot
10	Analog signal Pmod port (XADC)	25	Shared UART/ JTAG USB port
11	FPGA configuration reset button	26	Power select jumper and battery header
12	CPU reset button (for soft cores)	27	Power-good LED
13	Five pushbuttons	28	Xilinx Artix-7 FPGA
14	Pmod port(s)	29	DDR2 memory
15	Temperature sensor		

Power input requirements:
Powered from USB or any 4.5V-5.5V external power source

Connector specifications:
- -USB-JTAG programming circuitry
- -USB-UART bridge
- -USB HID Host for mice, keyboards and memory sticks
- -10/100 Ethernet PHY
- -Pmod connector for XADC signals
- -Four Pmod connectors providing 32 total FPGA I/O
- -12-bit VGA output
- -PWM audio output
- -PDM microphone

Memory:
- -128MiB DDR2
- -Serial Flash
- -microSD card slot

Processor:
Artix-7 FPGA:

-15,850 Programmable logic slices, each with four 6-input LUTs and 8 flip-flops (*8,150 slices)
-4,860 Kbits of fast block RAM (*2,700 Kbits)
-Six clock management tiles, each with phase-locked loop (PLL)
-240 DSP slices (*120 DSPs)

-Internal clock speeds exceeding 450 MHz
-Dual-channel, 1 MSPS internal analog-digital converter (XADC)

User interfaces:
-16 Switches
-16 LEDs
-Two RGB LEDs
-Two 4-digit 7-segment displays

Additional Information:
For more detailed specifications, pin assignments, and usage guidelines for the Digilent Nexys A7 board, please refer to the official board documentation and user manual provided by Digilent.

4.2 Software Detailed Design

This section provides detailed module designs for the software components of the Ingenion Telemetry Webserver project. Each module's design includes a narrative description, its function, conditions of use, processing logic, interfaces, security requirements, algorithms used, data structures, graphical representations of processing and control flow, and data entry/output graphics where applicable. Industry-standard module specification practices are followed.

Module 1 Telemetry Data Processing Module:

Narrative Description:

Function: The Telemetry Data Processing Module manages the collection and processing of telemetry data from various sources.
Conditions of Use: Scheduled for execution at specified intervals or upon request.
Processing Logic: Collects data from the AXI bus and peripheral devices, processes it for display, and stores it in appropriate formats.
Interfaces: Interfaces with the AXI bus, peripheral devices, and the web server for data presentation.
Security Requirements: Implements access controls to ensure only authorized users can initiate data processing.
Algorithms: Utilizes data sampling and signal processing algorithms for telemetry data processing.
Data Elements and Structures:

Input: Telemetry Data Streams
Output: Processed Telemetry Data, Graphical Representations
Graphical Representation:

Module 2: Command Management Module

Narrative Description:

Function: The Command Management Module is responsible for receiving and processing commands from authorized users and forwarding them to the appropriate subsystems or components for execution.
Conditions of Use: Accessed when authorized users need to send commands to control or configure the system.
Processing Logic: Validates user authorization, interprets received commands, and routes them to the relevant subsystems or modules for execution.
Interfaces: Interfaces with user input from the web interface, validates command syntax, and communicates with subsystems via appropriate interfaces.
Security Requirements: Implements strict access controls to ensure that only authorized users can send commands. Additionally, enforces command validation to prevent unauthorized or malicious commands.
Algorithms: Utilizes algorithms for command parsing, syntax checking, and routing.
Data Elements and Structures:

Input: User-Generated Commands
Output: Command Execution Status, Responses to User

These detailed module designs provide a comprehensive view of the system's functionality, logic, data flow, and user interfaces, ensuring a clear understanding of each module's role within the Ingenion Telemetry Webserver project.

4.3 Internal Communications Detailed Design

In this section, we will outline the detailed design of internal communications within the enhanced Total Verification System (TVS) project. Internal communications are essential for the exchange of information, commands, and data among system components.

The internal communication network for the TVS project will consist of the following components:

Web Server: The Digilent Nexys A7 board, with FreeRTOS running on a MicroBlaze CPU programmed onto its Artix 7 FPGA, acting as a web server, will serve telemetry data to connected clients.

Client Device: A single external computer or device will act as the client, connecting to the web server, via an ethernet connection, to access telemetry data.

For the web server's internal communication, no specific bus timing requirements or bus control mechanisms are necessary. Communication will occur over Ethernet using TCP/IP, which automatically handles timing and packet control.

Data exchange between the DUT and the Nexys A7 will primarily use the standard data formats for the telemetry data depending on the connection, such as UART, i2c, SPI, etc., and then will be transmitted to the web server in a structured format such as JSON or XML for ease of parsing and display.

5 EXTERNAL INTERFACES

5.1 Interface Architecture

In this section, we will provide a comprehensive description of the interface architecture for the Ingenion Telemetry Webserver project. This includes detailing the interfaces between the system being developed and other systems, as well as the architecture and protocols employed. A diagram will be provided to illustrate the communication paths between this system and each of the other systems, aligning with the context diagrams in Section 1.2.1. If necessary, subsections will be used to address specific interfaces being implemented.

The interface architecture of the Ingenion Telemetry Webserver project encompasses various communication pathways and protocols that enable interaction with external systems and components. Key elements of the interface architecture include:

<u>Ethernet Communication</u>: The primary means of communication between the Ingenion Telemetry Webserver and external systems is through Ethernet connections. This enables data transfer and command exchange between the web server system and external computers.

5.2 Interface Detailed Design

In this section, we will provide detailed design specifications for each interface that facilitates information exchange between the Ingenion Telemetry Webserver and external systems. This includes data format requirements, hand-shaking protocols, error reporting formats, graphical representations of data flow, and query and response descriptions. Each interface will be addressed separately, ensuring clarity and precision in interface design.

<u>Data Format Requirements:</u>
The data exchanged over Ethernet will adhere to standard data formats such as JSON (JavaScript Object Notation) for ease of parsing and compatibility with web technologies. Data may need to be converted into JSON format before transmission or processed upon reception.

<u>Hand-shaking Protocols:</u>
The hand-shaking between the web server and external systems will involve a request-response model. Handshake messages will include TCP requests and responses. The format will follow TCP protocol standards, including headers, status codes, and payload data. Timing for message exchange will be within specified response time limits to ensure timely communication.

Error Reporting:
Error reports will be communicated using standard TCP error status codes (e.g., 4xx or 5xx) in response messages. Error details will be included in the response body, allowing for proper error handling and logging. Critical errors may trigger operator notifications through alarm flags.

Graphical Representation:
A diagram (not shown here) will illustrate the connectivity between the web server and external systems, showing the direction of data flow. Arrows will depict data transmission pathways and the systems involved.

Query and Response:
Queries sent to the web server will be structured as TCP GET requests with specific endpoints and query parameters. Responses will be provided in JSON format, containing requested data or acknowledgments.

Data Format Requirements:
Communication with the Digilent Nexys A7 development board will utilize specific data formats compatible with FPGA programming and control. Data may need to be converted into binary or other formats suitable for FPGA interfacing.

Hand-shaking Protocols:
The hand-shaking with the Digilent Nexys A7 will involve predefined control commands and acknowledgment signals. Timing for handshakes will adhere to FPGA interface specifications.

Error Reporting:
Errors related to FPGA interfacing will be logged and may trigger system notifications. Detailed error reports will be retained in system logs for analysis and debugging.

Graphical Representation:
A separate diagram will illustrate the connectivity between the web server and the Digilent Nexys A7 development board, indicating the flow of commands and responses.

Query and Response:
Queries to the Digilent Nexys A7 will consist of control commands sent over dedicated communication channels. Responses will include acknowledgments or status updates.

For each interface, the detailed design specifications ensure that data exchange is accurately formatted, transmitted, and received and that hand-shaking, error handling, and communication protocols are well-defined and compliant with the respective systems being interfaced. Any formal Interface Control Documents (ICDs) relevant to these interfaces will be referenced as needed to maintain consistency and alignment with established standards.

6 SYSTEM INTEGRITY CONTROLS

Sensitive systems, particularly those handling information that, if compromised, could affect State programs or individual privacy, require robust integrity controls. These controls are essential for maintaining the confidentiality, availability, and accuracy of critical data. Below, we outline the minimum levels of control and specifications that should be implemented to ensure the integrity of the Ingenion Telemetry Webserver system.

<u>Access Restriction</u>: Implement internal security measures that restrict access to critical data items to only those access types required by authorized users. Role-based access control (RBAC) or similar mechanisms should be utilized to ensure that users can only access data relevant to their roles and responsibilities.
Audit Procedures:

<u>Control and Reporting:</u> Develop audit procedures to control, report, and retain audit information. This includes tracking user activities, system events, and changes to critical data.
Retention Period: Define a retention period for audit logs and reports, ensuring that historical data is retained for compliance, analysis, and investigation purposes.
Application Audit Trails:

<u>Dynamic Auditing</u>: Implement dynamic auditing mechanisms that record retrieval access to designated critical data. Audit trails should capture details such as user identity, timestamp, accessed data, and actions performed.
Standard Tables for Data Validation:

<u>Data Validation</u>: Utilize standard reference tables and validation rules for data fields. These tables should be used to validate data entered or received by the system to ensure accuracy and consistency.
Verification Processes:

<u>Data Modification Controls:</u> Establish robust verification processes for additions, deletions, or updates of critical data. Such processes should include validation checks, approval workflows, and logging of changes.
Audit Information Identification:

<u>User Identification</u>: Ensure that all audit information is associated with the user's identification, allowing traceability of actions to specific individuals.

<u>Network Terminal Identification:</u> Record the network terminal or device used for the actions performed in the audit trail.

<u>Date and Time Stamps:</u> Include date and time stamps in audit logs to accurately track when events occurred.

<u>Data Accessed or Changed:</u> Document the specific data accessed or changed during user actions to provide a complete audit trail.

7 Document Organization

7.1 Project References

This section provides a bibliography of key project references and deliverables that have been produced before this point.

"Arty Freertos Web Server." ARTY FreeRTOS Web Server - Xilinx Wiki - Confluence, xilinx-wiki.atlassian.net/wiki/spaces/A/pages/18841844/ARTY+FreeRTOS+Web+Server. Accessed 29 Sept. 2023.

"Microblaze Soft Processor Core." AMD, www.xilinx.com/products/design-tools/microblaze.html. Accessed 29 Sept. 2023.

"Microblaze Webserver Demo for the Arty Evaluation Kit." AMD, www.xilinx.com/video/fpga/microblaze-webserver-demo.html. Accessed 29 Sept. 2023.

"The LXI Primer." Lxistandard.Org, www.lxistandard.org/Documents/LXI%20Primer/The%20LXI%20Primer%20Final.pdf. Accessed 29 Sept. 2023.

7.2 Glossary

- Ingenion Telemetry Web Server: The software component responsible for facilitating communication and interaction between the Digilent Nexys A7 development board, Xilinx Artix 7 FPGA, MicroBlaze soft-core CPU running FreeRTOS, and external client devices via Ethernet. It is a critical part of the Total Verification System (TVS) project.
- Total Verification System (TVS) Project: A project aimed at enhancing satellite hardware testing by integrating advanced features and improving telemetry data management.
- Digilent Nexys A7: A development board used in the project for interfacing with satellite hardware components.
- Xilinx Artix 7 FPGA: A Field-Programmable Gate Array (FPGA) from Xilinx used to host the MicroBlaze soft-core CPU and execute the software.
- MicroBlaze: A soft-core microprocessor designed by Xilinx for implementation on FPGAs.
- FreeRTOS: An open-source real-time operating system used for running tasks on the MicroBlaze CPU.
- Ethernet: A network communication protocol used for connecting the Nexys A7 board to external client devices.
- TCP/IP: Transmission Control Protocol/Internet Protocol, the suite of communication protocols used for transmitting data over networks, including the internet.
- User Interface (UI): The graphical interface that allows end-users to interact with and control the telemetry system.
- Telemetry Data: Data collected from satellite hardware components, used for monitoring and control.

- External Computer: A device connected to the telemetry system via Ethernet for data exchange and management.
- Real-Time Web Server: A web server that provides immediate and interactive access to telemetry data and system controls.
- Communication Protocols: Standards like Ethernet and TCP/IP used for reliable data exchange within the system.
- Xilinx MicroBlaze CPU: The soft-core CPU integrated into the Artix 7 FPGA for processing tasks.
- HTTP Protocol: Hypertext Transfer Protocol, used for communication between web browsers and the web server.
- WebSockets: A communication protocol that enables real-time, bidirectional communication between the web server and connected clients.
- IEEE 802.3: A standard for Ethernet communication.
- HTTP POST Requests: A method for sending data from a user to a web server.
- System Features: Key functionalities or capabilities of the telemetry web server, such as communication link establishment, user web interface, and telemetry data parsing and display.