Detailed Design Documentation

SCADA System Development for Oversight of all MARS Facilities

Document Acronyms

Table 01-2 Document Acronyms

Acronym	Description
DHF	Design History File
IEEE	Institute of Electrical and Electronics Engineers
CR	Customer Requirement
SR	System Requirement
VPN	Virtual Private Network
DAS	Data Acquisition Systems
IDS	Intrusion Detection Systems
VSA	Virginia Spaceport Authority
SCADA	Supervisory Control and Data Acquisition
PLC	Programmable Logic Controller
HMI	Human-Machine Interface
GUI	Graphical User Interface
MARS	Mid-Atlantic Regional Spaceport

Problem Statement

Design and implement a SCADA system to integrate all MARS launch pad facilities, providing a unified, secured, reliable, redundant and scalable interface that efficiently collects sensor and relay device data to the central server’s database and GUI.

Key Stakeholders

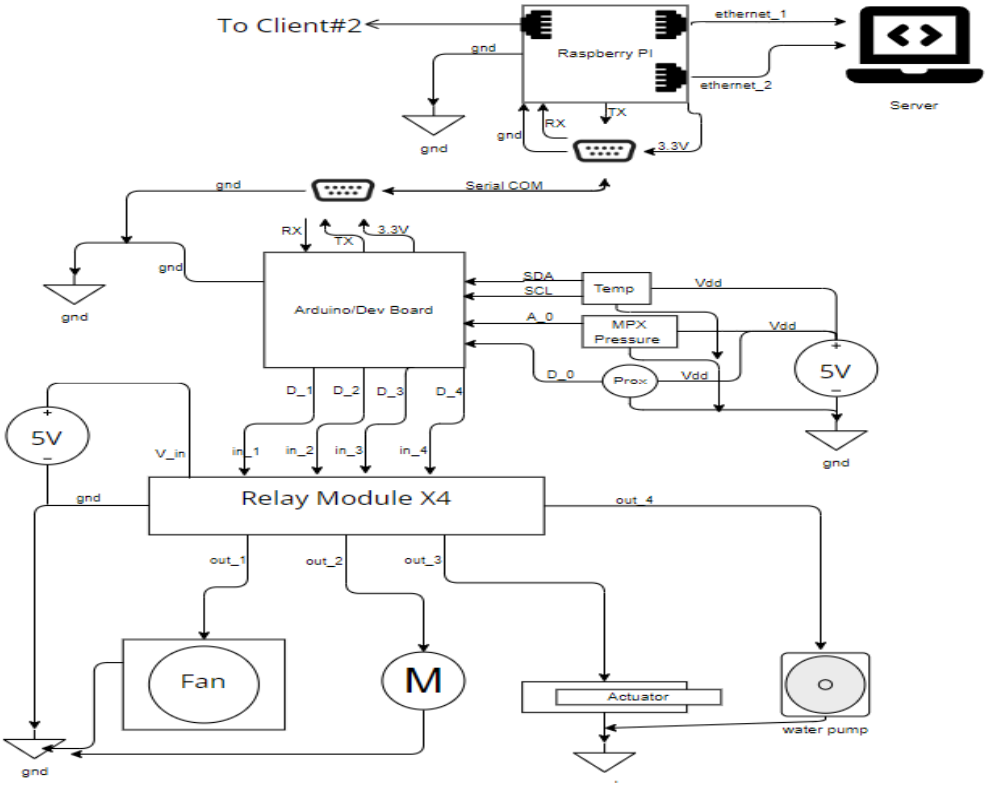
1. Virginia Spaceport Authority (VSA):
    - As the governing body for spaceport activities in the state, the VSA will be directly impacted by the efficiency, reliability, and security of the SCADA system. Their operational success depends on the comprehensive visibility and control offered by this system.
2. MARS Facility Management Teams:
    - They manage the day-to-day operations of the individual facilities. The SCADA system will be their primary tool to oversee, coordinate, and troubleshoot processes across the MARS sites.
3. IT and Network Department:
    - This team will be responsible for the actual implementation, maintenance, and troubleshooting of the SCADA system. They would ensure the system's integration with existing IT infrastructure, maintain cybersecurity standards, and ensure the system's uptime.
4. Local and State Regulatory Bodies:
    - These entities have set guidelines and standards that the SCADA system and its operations must comply with. The compliance, safety, and environmental impact of the SCADA system directly concern them.
5. Launch Service Providers and Customers:
    - Companies or organizations that utilize the MARS facilities for their launch activities are directly impacted by the efficiency and effectiveness of the SCADA system. They rely on the facilities to ensure timely, safe, and coordinated launch processes. The SCADA system's data visibility, logging, and manipulation capabilities will have direct implications on their missions' success, scheduling, and cost-efficiency.

Architectural Design Concept

- The system needs to be designed to ensure redundancy in almost any condition. To achieve this, multiple connections will be made to each client and from client to server, effectively creating a full mesh network architecture.
- Security is also extremely important, most critically, where data has traveled, and what each user is interacting with.
    - Ethernet connections provide physical communication lines between clients and from client to server.
    - The project will adhere to ITAR and NIST regulations.
- Hardware implementation will consist of the following components:
    - Laptop acts as the main server.
        - Multiple connections to the Clients or "facilities" via ethernet.
        - Localized data storage.
        - Primary and secondary virtual machines for redundancy.

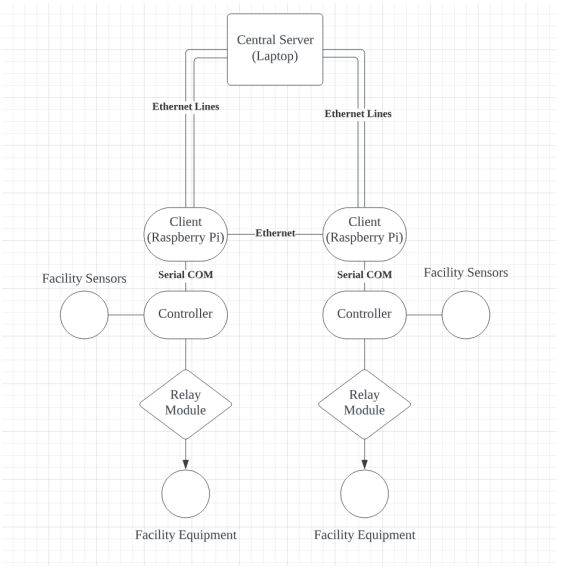
- Raspberry PI boards act as the Clients.
  - Redundant ethernet connections to the main server.
  - Ethernet connection between clients.
  - Serial connection to PLC board running different serial protocols (RS-232, RS-485, ModBus, etc..)
- Arduino acting as the PLC controllers:
  - Analog and digital sensors are connected directly to this board.
  - PLC will control a 4-channel relay device.
- Relay board:
  - 4-channel relay module to control end devices.

- Sensors and Devices:
  - Relay Devices:
    - LEDs
    - Fans
    - DC Linear Actuators
    - Motors
    - Electric heater
    - Water pump
  - Sensors:
    - AHT10 analog temperature and humidity sensor
    - TSL2561 digital light sensor
    - MPX series analog pressure sensor
    - ADXL335 analog accelerometers
    - Infrared sensor

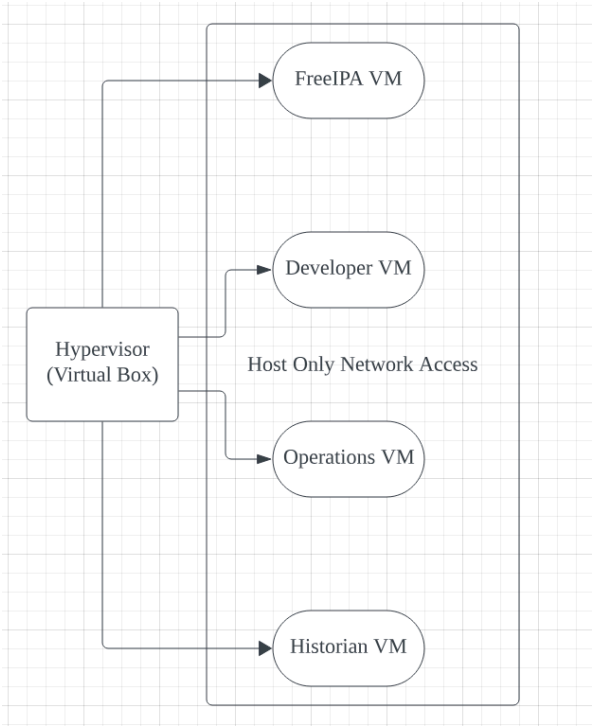


Detailed Design

- Network Diagram
    - As mentioned above, a laptop will act as our main server, from the laptop, four ethernet lines will connect to two clients. These clients represent the facilities at VSA. The clients also share a connection via ethernet in case communication from a client to the server is malfunctioning. The network diagram is visible in the image below.



- Redundancy using Virtual Machines
    - FreeIPA will be used as the management system and supports the following components:
        - Audit logging
        - Role-based access control is used to monitor individual users and their interactions with the system.
        - Sudo Rules
        - Syslog integration
    - Operations Virtual Machine
        - Client of the FreeIPA server with configuration for user authentication and centralized logs.
        - Integrates with all other VMs to run a UI program for system monitoring and control.
        - Setup with an identical backup VM for redundancy which can take over within 100ms (Customer Requirement).
    - Historian Virtual Machine
        - Client of the FreeIPA server with configuration for user authentication and centralized logs.
        - Implements SQL database and Zeek (formerly Bro) security monitor software.
        - Tracks device IDs for all connected devices and logs for any changes.
        - Writes data to SQL database and stored on multiple external devices.
        - Setup with an identical backup VM for redundancy which can take over within 100ms (Customer Requirement).
    - Developer Virtual Machine
        - Client of the FreeIPA server with configuration for user authentication and centralized logs.
        - Configured with sudo access and rules.
        - Will come with development tools installed.
    - Virtual Box Hypervisor:
        - VMs will be air-gapped to external networks using Host-Only Network Access.



- Security Integrations
    - This system must comply with ITAR, specifically category 11. Category 11 covers electronic systems, and equipment and/or software for the purpose of gathering data. Overall, the concern lies with tracking data and user-system interactions and not so much with actual data encryption.
    - This system must also comply with NIST SP 800-171 and NIST SP 800-172 regulations
        - NIST SP 800-171 provides requirements for protecting confidential unclassified information.
        - NIST SP 800-172 is a supplementary document to 800-171 which is designed to safeguard sensitive information through the best practices and security controls.
    - Network Time Protocol or NTP will be used as a central server clock. NTP will allow for synchronization across facilities and distributed between applications. NTP also allows for the use of firewall policies.
    - Zeek Network Security Monitor
        - Used for monitoring connected devices.
        - Custom scripts are used to generate and write logs to SQL database in the specified format.
        - Monitor device health with support for alerts.
- SQL Database
    - The data will be stored locally in a SQL database using SQLite.
    - SQLite is lightweight and serverless, easy to set up, and has the ability to store the entire database as a single file.

    - Data storage breakdown:
        - First column: Timestamp
        - Second column: Source (where the data is coming from) using device ID
        - Third column: Device ID
        - Fourth column: Data type
        - Fifth column: Data value

Timestamp	Facility/Source	Device ID	Data Type	Value

- User Interface
    - A user interface will be developed with a version for the main server and the individual clients. Both implementations will be developed using C++ and QT5 for the graphical components.
    - Server Implementation:
        - Contains quick views of the connected clients. Views will include errors, alerts, and data logging.
        - There will be a login window which the user will be required to login with before any data is available to that user.
        - Depending on the user's credentials, certain parameters and items may not be available to them.
    - Client Implementation:
        - Very similar to the server implementation but with more of a focus on end-device monitoring rather than client/facility monitoring.
        - A login window will be standard at startup, different users will have differing levels of access.
        - The home page will have relevant end-device status information and will contain relevant errors and alerts.

Target Specifications

Target specifications are prioritized 1 to 5, 5 being the highest priority.

Req. #	Metric	Priority	Units	Marginal Value	Ideal or Target Value	Measurement Device(s)
SR-1	System operating temperature	5	°C	10 - 35°C	20-30°C	Thermocouples
SR-2	System Response Time	4	Seconds	<=3	<=1	Google Public NTP
SR-3	Network Bandwidth	5	Mbps	>=500	>=1000	Network Cable Tester

CR-1	System Operating Voltage	5	V	10-500V	110-240V	Oscilloscope
SR-4	System Data Logging Frequency	5	Seconds per log	Every 5	Every 1	Frequency Counter
SR-5	VM Boot-up Time	2	seconds	<=60	<=30	Central Time Source (CTS)
SR-6	System Monitoring Frequency (Health/Alerts)	5	seconds	Every 10	Every 2	Google Public NTP
CR-2	External Vibration Tolerance	3	g-force	Up to 1g without data loss	Up to 2g	Accelerometer
CR-3	Virtual Machine Configuration	4	NIST Standard	At least 1 VM per operation	2 VMs per operation	Hypervisor

Verification Test Plan

Each target specification below has been attributed to either a system requirement (SR) or a customer requirement (CR). To validate our prototype we have outlined detailed test procedures below:

SR-1: System Operating Temperature

- Detailed Testing Procedure:
  - The primary aim of this test is to gather accurate data which allows for the verification of the system's operating temperature against the predefined target specification.
  - This test ensures that our system consistently operates within the ideal temperature range during its real-world application.
  - Failure to operate within these temperature ranges may cause component/system failure and in more extreme cases could lead to injuries.
- Type of Testing Method & Significance:
  - This test will be conducted by establishing a temperature-stable starting state and varying operating parameters such as connected devices and network traffic.
- Step-by-Step Procedure Instruction:
  - Gather the necessary materials and initiate the system at a stable temperature.
  - Position temperature sensors at strategic points throughout the system (client, server, motors, etc.).
  - Begin logging temperature data at set intervals following the central clock.

    - Begin varying system parameters such as network traffic, operating voltages, and overall loads.
    - Analyze captured temperature data for discrepancies between actual and target temperature ranges.

SR-2: System Response Time

- Detailed Testing Procedure:
    - The goal of this test is to determine the response time of our system's communication protocols and system processes.
- Type of Testing Method & Significance:
    - This test will be conducted through console/GUI logs with NTP to get the exact time of each process. The significance of an immediate system processing time is to ensure that there are no delays when operating the relay devices.
- Step-by-Step Procedure Instruction:
    - Test the serial communications by sending information from client to server.
    - Measure the processing time through logs and validate that the time difference between the send and receive is no more than 3 seconds with an ideal time of less than or equal to 1 second.
    - Measure the system processing time of relay devices by checking the logs of the Arduino clients. The time difference should also be a marginal value of <= 3 seconds and an ideal value of <= 1 second.
    - If the processing times of either test do not meet the target/marginal values, then troubleshoot the system's software aspect/resource allocation to improve system responsiveness.
- Data Collection Methods:
    - Monitor system log messages through console/GUI
    - Ensure system alerts are synchronized with NTP
- Safety Procedures
    - When making adjustments to system processing time (i.e. more efficient data management/data transfer communication), make sure the functionalities of the code remain the same to ensure a quality system.

SR-3: Network Bandwidth

- Detailed Testing Procedure:
    - The goal of this test is to determine the network bandwidth of our system. This is crucial for ensuring efficient data transfer and communication within the network.
- Type of Testing Method & Significance:
    - Network Bandwidth Testing: This testing method assesses the speed and reliability of data transfer between the main server and client devices, ensuring that the network can handle the required data load without performance degradation.
    - Significance: Network bandwidth is crucial for maintaining responsive communication between devices, avoiding bottlenecks, and supporting the overall functionality of the system.

- Step-by-Step Procedure Instructions:
    - Establish a baseline measurement of the initial network bandwidth without any additional load.
    - Gradually introduce increasing loads to the network by simulating data transfer activities between the main server and client devices.
    - Apply a peak load to the network to assess its maximum capacity. This could involve simultaneous data transfers, file downloads, or other activities that mimic peak usage scenarios.
    - Measure the latency of data transfer between the main server and client devices.
- Data Collection Methods:
    - Bandwidth Measurement Logs: Maintain detailed logs of bandwidth measurements at various load levels, including baseline, gradual load, and peak load scenarios. Use network testing tools to generate controlled traffic. Record the bandwidth at each load level to identify the point at which the network performance starts to degrade or becomes inconsistent.
    - Latency Logs: Record latency measurements to identify any patterns or trends affecting the responsiveness of the network.
    - Network Traffic Analysis: Utilize network traffic analysis tools to capture and analyze the actual data packets being transmitted, helping to identify any anomalies or inefficiencies.
- Safety Procedures:
    - Follow general safety guidelines when working with networking equipment and ensure that any testing activities do not disrupt critical network services.
    - Use caution when introducing additional loads to the network, and ensure that the testing activities do not impact other users or systems sharing the same network infrastructure.

SR-4: System Data Logging Frequency

- Detailed Testing Procedure:
    - The goal of this test is to ensure the data logging frequency across all client devices in the network is synchronized with the main server. The centralized clock will be used for data logging and to keep clients and connected devices synchronized.
- Type of Testing Method & Significance:
    - Network Time Protocol (NTP) Synchronization Testing: This testing ensures that all devices connected to the server record events based on the same time reference. This is critical for data consistency and event correlation.
    - Significance: Accurate data logging is essential for time-sensitive operations, error logging, troubleshooting, and overall system integrity.
- Step-by-Step Procedure Instructions:
    - Configure the server as an NTP server. This can be accomplished by integrating NTP server software and ensuring it syncs with a reliable external time source.
    - Synchronize each client device with the central NTP server. The client and other relevant devices should sync with the central NTP server at regular intervals.

    - Validate synchronization by periodically checking each client's system clock against the central NTP server to verify accurate synchronization.
    - Record data logging times by logging data containing timestamps from the client devices. Compare data logging frequency to the target frequency of one log per second.
- Data Collection Methods:
    - Timestamp Verification: Collect a sample of log entries and their corresponding timestamps to analyze the precision of the logging frequencies.
    - NTP Logs: Utilize the NTP server logs on the server to monitor synchronization requests and responses from client devices.
    - Centralized Logging: Consolidate all client logs to the central server for a unified analysis. Ensure that client logs have consistent timestamps that match the centralized NTP server.
- Safety Procedures:
    - Follow general safety guidelines and use caution when handling powered devices (self-grounding may be required).

SR-5: VM Boot-up Time

- Detailed Testing Procedure:
    - The goal of this test is to ensure the VM environment configuration works and whether the baseline software setup of the machine is standardized to testing conditions.
- Type of Testing Method & Significance:
    - Efficiency and Speed of VM boot-up process: This test ensures that the VM is working properly and can determine the quality of the machine as well.
    - Impact of different configurations: This test shows the differences in the configurations of the VM and how they change the environment.
- Step-by-Step Procedure Instructions:
    - Initialize testing environment with accurate measurements
    - Create baseline measurements for the VM
    - Vary testing parameters for each environment, or change one variable to see their effect
    - Data collection for this test
    - Analysis of the data to conclude the efficiency and speed of the VM Boot-up time
- Data Collection Methods
    - Automated Timing Tools to precisely measure the time from boot initiation to VM readiness
    - System logs and performance monitoring software
- Safety Procedures
    - Ensure there are emergency protocols set in play for shutting down or restarting the VM
    - Compliance with safety standards for the environment and security of the system
    - Data backup in case any data is lost throughout the test

SR-6: System Monitoring Frequency (Health/Alerts)

- Detailed Testing Procedure:
    - The goal of this test is to monitor our system's health by periodically checking the status of our system every 10 (marginal) or 2 (ideal) seconds. It is imperative that the user is notified of the system's status so that manual action can be taken to resolve the issue immediately. The three status markers should be DANGER (red), WARNING (yellow), and SAFE (green).
- Type of Testing Method & Significance:
    - Console and GUI testing of timestamp logs using Google Public Network Time Protocol (NTP) for standard timekeeping.
- Step-by-Step Procedure Instructions:
    - 1. Implement a logging feature in all client and server UI
    - 2. Periodically print out the client system’s status through the console UI
    - 3. Send status and timestamps to server UI (diverges into 4.a or 4.b)
    - 4.a If the status is DANGER or WARNING, the client should display this message so that an operator can resolve the issue at the site (i.e. reset power supply to fans/actuator/pump). Once resolved, the status should be updated to the local and server UI
    - 4.b If status is SAFE then facility operations continue.
- Data Collection Methods
    - Monitor system log messages through console/GUI
    - Ensure system alerts are synchronized with NTP
- Safety Procedures
    - When resolving issues, the operator must comply with the necessary safety precautions (i.e if handling fluids, wear PPE)

CR-1: System Operating Voltage

- Detailed Testing Procedure:
    - Measure the system’s operating voltage range using an oscilloscope to ensure compliance with the target specification.
    - Measure connected devices to ensure they operate within their respective voltage ranges.
- Type of Testing Method & Significance:
    - Electrical testing using an oscilloscope or software-implemented voltage monitors. This type of testing is significant for verifying the system’s ability to operate within the specified voltage range without malfunctions.
- Step-by-Step Procedure Instructions:
    - Connect an oscilloscope or ensure the software voltage is logging properly.
    - Enable the system and ensure all components are operating properly.
    - Gradually vary the system's operating voltage within its respective operating range.
    - Monitor voltage readings for each component, ensuring voltage levels remain within their respective ranges.
- Data Collection Methods:

    - Capture oscilloscope or internal voltage readings at specified intervals.
    - Create an output log containing all voltage readings for future revision.
- Safety Procedures:
    - Ensure all electrical connections are secure and properly connected.
    - Use appropriate PPE such as safety glasses.
    - Have an extinguisher on hand.
    - Use caution and slowly adjust voltage levels to avoid sudden spikes in voltage/current.

CR-2: External Vibration Tolerance

- Detailed Testing Procedure:
    - The goal of the test is to ensure that the device is not affected by external vibration factors which could be caused by something like vibration caused by motors or a launch etc.
- Type of Testing Method & Significance:
    - Testing external vibration tolerance by initializing communication between a Raspberry Pi and an Arduino to act as a server and a client and introducing vibration (in the form of shaking enclosure) which is measured using an accelerometer.
- Step-by-Step Procedure Instructions:
    - 1. Connect the Arduino to the Raspberry pi and ensure network connectivity.
    - 2. Initialize the data transfer/communication from Raspberry pi to Arduino.
    - 3. During the data transfer, introduce a light shake to the enclosure, while checking the readings from the accelerometer.
    - 4. Ensure the integrity of the transferred data by comparing it to the original using a separate device if required.
    - 5. Repeat step 2 but increase the vibration (shaking) to the enclosure slowly.
    - 6. Store accelerometer results.
- Data Collection Methods
    - Use a Network Cable Tester to ensure the integrity of the ethernet cable.
    - Use a Raspberry Pi as a server and Arduino as a client.
    - Monitor system logs for accelerometer readings.
- Safety Procedures
    - Wear an ESD Strap during testing to prevent any electrostatic discharge from damaging the Arduino.
    - Work on an ESD-safe surface to prevent electrostatic discharge from damaging the Arduino.
    - Be careful with implementing the procedure as this is not a drop test but an external vibration tolerance test.

CR-3: Virtual Machine Configuration

- Detailed Testing Procedure:

    - The goal of the test is to ensure the redundancy of the VM configuration and check that a secondary VM takes over the operation if the primary VM goes down.
- Type of Testing Method & Significance:
    - Testing software developed on development VM environment and testing redundancy of historian and operation VMs using Hypervisor to monitor the tests.
- Step-by-Step Procedure Instructions:
    - 1. Initialize a development VM and write software to be tested while using some libraries that work in that environment.
    - 2. Ensuring that the developed software works in the operations VM environment.
    - 3. Initialize operations VM which will monitor the system and initialize historian VM which will store the data. Also, initialize the secondary VMs for the same.
    - 4. Have a system monitoring log on the main OS and compare it to the operations VM system monitoring log.
    - 5. Shut down the primary operations VM. Ensure the secondary operations VM is still running and compare the 2 logs to see if any updates were lost.
    - 5. Similarly, have primary historian VM store data being provided by an Arduino (client). Compare the data from Arduino to the data being stored in the VM.
    - 6. Shut down primary historian VM. Ensure the secondary historian VM is still running and compare the data stored to see if it is compromised.
- Data Collection Methods
    - Use Hypervisor to monitor VM status and performance.
    - Use an Arduino to provide data from a sensor.
    - Monitor system log messages through console/GUI on main OS and VM
- Safety Procedures
    - Robust Software practices to prevent device-breaking malfunctions.
    - Robust software to ensure secure testing and prevent missing bugs.

CR-4: Communication Redundancy

- Detailed Testing Procedure:
    - The goal of the test is to transfer data between an Arduino and a Raspberry pi using EtherNet/IP and a network cable tester to ensure the connection status. We will cut a connection between the devices during a data transfer and want to ensure that the transfer still goes through. This simulates the Customer's Requirement for Communication Redundancy
- Type of Testing Method & Significance:
    - Testing communication and redundancy using EtherNet/IP and WPA2 WiFi between Arduino and Raspberry Pi acting as a client and a server.
- Step-by-Step Procedure Instructions:
    - 1. Connect the Arduino to the Raspberry Pi and ensure network connectivity.
    - 2. Initialize the data transfer from Raspberry Pi to Arduino.
    - 3. During the data transfer, cut the EtherNet/IP connection. The secondary communication line should be connected (Could be EtherNet/IP again or WiFi WPA2) and take over the data transfer.

    - 4. Ensure the integrity of the transferred data by comparing it to the original using a separate device if required.
    - 5. Update the log with information on the cut connection and the status of the data transfer.
    - 6. Repeat step 2 but transfer the data from Arduino to Raspberry Pi.
- Data Collection Methods
    - Use a Network Cable Tester to ensure the integrity of the ethernet cable.
    - Use a Raspberry Pi as a server and Arduino as a client.
    - Monitor system log messages through console/GUI
- Safety Procedures
    - Wear an ESD Strap during testing to prevent any electrostatic discharge from damaging the Arduino.
    - Work on an ESD-safe surface to prevent electrostatic discharge from damaging the Arduino.
    - Robust Software practices to prevent device-breaking malfunctions.

Table 01-6 Standards and Statutory Requirements

Req. #	Requirement	Source Document (e.g., standard, regulatory requirements)	Details
SR-1	Server Configuration Development part 1: Configure redundant virtual machines	NIST CSF 2.0	Section 3.1: Creating and using framework profiles to understand, assess, prioritize, and communicate
SR-2	Scalable Network Topology, Server Configuration Development	IEC 62443	Standards for securing industrial controls systems against cyber threats and vulnerabilities
SR-3	Human-Machine Interfaces	ISO 92-41-410 [6]	Standards related to the ergonomic design of human-system interfaces, particularly displays and control software.
SR-4	Interoperability in Power Utilities	IEC 61850 [7]	Standards focused on data and communication exchange in power utility automation systems
SR-5	Quality Management System Requirements	ISO 9001:2015 [8]	Standards that specify requirements for establishing, implementing, and maintaining a quality management system within an organization. Emphasizes customer satisfaction.

SR-6	Security and Privacy Controls for Information Systems	NIST SP 800-53 [9]	Standards for protecting organizational operations and assets from a diverse set of threats and risks.
SR-7	Manufacture, Export and temporary Import of Defense Articles	ITAR Part 121 [10]	Standards for International Traffic in Arms Regulations, specifically the US Munitions List.