**System Requirements Specification**

**(SRS) - DriveSync**

**Team: Four\_A**

**Ahmed Sayed**

**Abdelrahman Sanad**

**Alaa Youssif**

**Aliaa El Khateb**

**Supervised by: NGEN\_Innovations**

**Document Change History:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Release** | **Changed by** | **Change Description** |
| **2023- 9- 28** | **A23\_9\_1** | **Four\_A** | * **Initial Release** |
| **2023- 10- 4** | **A23\_10\_2** | **Four\_A** | * **No content changes** * **Changed SRS\_ID** |

# System Requirements Specification

# (SRS) - DriveSync

Table of Contents

1. **Introduction**

1.1 Purpose  
1.2 Scope  
1.3 Definitions, Acronyms, and Abbreviations  
1.4 References  
1.5 Overview of the Document

1.6 Requirement Naming Convention

1. **System Overview**

2.1 System Components  
2.2 Operational Modes  
2.3 System Architecture

1. **System Overview**

3.1 **Software and Hardware Requirements**  
3.2 **Integration Requirments**

1. **Security Requirements**

4.1 Authentication  
4.2 Data Integrity  
4.3 Access Control

1. **Safety and Performance Requirements**

5.1 Prioritization  
5.2 Initialization  
5.3 Voltage Monitoring  
5.4 Response Times  
5.5 Resource Utilization  
5.6 Fault Handling

1. **Communication and Data Handling**

6.1 Communication  
6.2 Sensor Data  
6.3 Frame Structures  
6.4 Firmware Updates

1. **Error Handling and Logging**

7.1 Error Messages  
7.2 Fault Logging  
7.3 Firmware Updates Log

1. **Size and Format Constraints**

8.1 Firmware Size

1. **Operation Modes**

9.1 Operation Mode  
9.2 Protected Engineering Mode

1. **Requirements Validation and Testing**

10.1 Validation Strategies  
10.2 Integration and Validation

1. **Appendices**

11.1 Glossary of Terms  
11.2 Acronyms and Abbreviations  
11.3 Document Revision History

### 1. Introduction

#### 1.1 Purpose

The purpose of this SRS is to provide a detailed description of the requirements and specifications for the DriveSync system, which facilitates communication and control between automotive Electronic Control Units (ECUs) in a vehicle.

#### 1.2 Scope

This document covers the functional and non-functional requirements, operational modes, security features, safety requirements, communication protocols, error handling, and firmware management for the DriveSync system.

#### 1.3 Definitions, Acronyms, and Abbreviations

* ECU: Electronic Control Unit
* CPU: Central Processing Unit
* HMI: Human-Machine Interface
* UART: Universal Asynchronous Receiver-Transmitter
* ADC: Analog-to-Digital Converter
* NVM: Non-Volatile Memory
* CRC: Cyclic Redundancy Check

#### 1.4 References

* STM controller documentation
* C programming language documentation

#### 1.5 Overview of the Document

This section provides a brief overview of the document structure and the content within each section, aiding readers in navigating the SRS document

**1.6 Requirement Naming Convention**

The naming convention for requirements follows a structured format to provide meaningful information about each requirement. The convention is as follows:

* **AUTOSAR Layer**: The name after SRS represents the AUTOSAR layer to which the requirement pertains. Each AUTOSAR layer has a unique numeric identifier.
* **Edition**: The first digit from the left indicates the edition of the SRS document. It represents the version or release of the document.
* **Section**: The second digit represents the name of the section in which the requirement is categorized. Each section has a unique numeric identifier.
* **Sub-Section**: The third digit represents the sub-section number within the specified section. Sub-sections are used to group related requirements.
* **Requirement Number**: The two digits following the sub-section number represent the unique identifier for each requirement within the sub-section.

For example, consider the requirement **SRS\_BSW\_13101**:

* **AUTOSAR Layer**: BSW (This indicates that it’s in BSW layer.)
* **Edition**: 1 (This indicates the first edition or version of the SRS document.)
* **Section**: 3 (This corresponds to a specific section within the SRS document.)
* **Sub-Section**: 1 (This further categorizes the requirement within the section.)
* **Requirement Number**: 01 (This is a unique identifier for the specific requirement within the sub-section.)

### 2. System Overview

#### 2.1 System Components

* **Central Processing Unit (CPU):** Responsible for managing communication protocols, processing sensor inputs, and controlling actuators. It interacts with multiple ECUs through a UART interface, ensuring seamless communication.
* **Human-Machine Interface (HMI) Unit:** Provides a user-friendly interface for displaying real-time brake and accelerator values. It includes input interfaces for voltage measurement, accelerator sensor input, and brake sensor input. Additionally, it incorporates LED indicators for left and right brake signals, enabling visual feedback.
* **Sensors:** Utilizes sensors, such as potentiometers, to capture brake and accelerator inputs.
* **Actuators:** Controls the braking and acceleration mechanisms of the vehicle.

#### 2.2 Operational Modes

* **Operation Mode:** In this mode, the system functions under normal operational conditions. It includes the following functionalities:
  + Reading sensor values from the ECUs.
  + Displaying sensor data on the HMI.
  + Controlling actuators to manage braking and acceleration.
* **Protected Engineering Mode:** This mode provides access to advanced system functions, typically intended for maintenance, calibration, and diagnostics. Protected Engineering Mode includes the following capabilities:
  + Configuring parameters, such as brake indicator brightness.
  + Performing NVM calibration data changes.
  + Utilizing a seed key-based diagnostic service for authorized access.

Access to Protected Engineering Mode is strictly controlled and requires proper authentication to prevent unauthorized access to critical system functions.

#### 2.3 System Architecture

* **Central Processing Unit (CPU):** At the core of the architecture, the CPU orchestrates communication between various system components. It manages communication protocols, processes sensor inputs, and executes control commands to the actuators.
* **Electronic Control Units (ECUs):** The system operates with a distributed architecture, which typically includes multiple ECUs. While the exact number of ECUs may vary depending on the vehicle and its requirements, a typical configuration includes three main ECUs:
  + **Brake ECU:** Responsible for managing brake-related functionalities.
  + **Accelerator ECU:** Responsible for managing accelerator-related functionalities.
  + **Central ECU:** Acts as a central coordinator, facilitating communication and processing between the different ECUs, ensuring seamless integration and synchronization.

**3. Input and Control Requirements**

**3.1 Software and Hardware Requirements**

# 3.1.1. Functional SRS:

|  |  |
| --- | --- |
| ID | SRS\_BSW\_13101 |
| Description | The system shall be developed using the C programming language. |
| Rationale | Using C ensures consistency and compatibility in software development. |
| Use Case | All software components and modules shall be written in the C programming language. |
| Priority | High |
| Dependencies | Availability of C compiler and development tools. |

# 3.1.2 Non-Functional SRS:

|  |  |
| --- | --- |
| ID | SRS\_PER\_13102 |
| Description | The system shall be optimized to utilize the capabilities of the ARM-based controller efficiently, ensuring optimal performance. |
| Rationale | Optimizing controller utilization maximizes system efficiency and responsiveness. |
| Use Case | The system is designed to make efficient use of the capabilities of the ARM-based controller, avoiding unnecessary resource wastage. |
| Priority | Medium |
| Dependencies | Performance optimization techniques for ARM-based controllers. |

|  |  |
| --- | --- |
| ID | SRS\_BSW\_13103 |
| Description | The system's software development process shall follow industry-standard coding practices. |
| Rationale | This requirement ensures adherence to recognized coding practices, enhancing code quality and maintainability. |
| Use Case | Developers follow industry-standard coding practices while writing software components for the system. |
| Priority | High |
| Dependencies | Familiarity with and implementation of industry-standard coding practices. |

|  |  |
| --- | --- |
| ID | SRS\_BSW\_13104 |
| Description | The software development process shall include version control and documentation practices. |
| Rationale | This requirement emphasizes the importance of version control and documentation for software development. |
| Use Case | Version control and documentation practices are integral parts of the software development process. |
| Priority | High |
| Dependencies | Implementation of version control and documentation tools and processes. |

# ****3.2 Integration Requirements****

# 3.2.1. Functional SRS

|  |  |
| --- | --- |
| ID | SRS\_SWC\_13201 |
| Description | The software components (SWCs) shall integrate with **sensors** for **brake** inputs, including potentiometers or equivalent sensor types. |
| Rationale | Sensor integration ensures that the system can accurately monitor brake inputs. |
| Use Case | SWCs responsible for monitoring brake inputs are configured to interface with the respective sensors. |
| Priority | High |
| Dependencies | Sensor interface and data processing within SWCs. |

|  |  |
| --- | --- |
| ID | SRS\_SWC\_13204 |
| Description | The software components (SWCs) shall integrate with **sensors** for **accelerator** inputs, including potentiometers or equivalent sensor types. |
| Rationale | Sensor integration ensures that the system can accurately monitor accelerator inputs. |
| Use Case | SWCs responsible for monitoring accelerator inputs are configured to interface with the respective sensors. |
| Priority | High |
| Dependencies | Sensor interface and data processing within SWCs. |

|  |  |
| --- | --- |
| ID | SRS\_SWC\_13205 |
| Description | The software components (SWCs) shall integrate with **actuators** for **brake** controls. |
| Rationale | Actuator integration ensures that the system can control brake mechanisms effectively. |
| Use Case | SWCs responsible for controlling brake actions are configured to interface with the respective actuators. |
| Priority | High |
| Dependencies | Actuator interface and control algorithms within SWCs. |

|  |  |
| --- | --- |
| ID | SRS\_SWC\_13206 |
| Description | The software components (SWCs) shall integrate **actuators** for **accelerator** controls. |
| Rationale | Actuator integration ensures that the system can control accelerator mechanisms effectively. |
| Use Case | SWCs responsible for controlling brake and accelerator actions are configured to interface with the respective actuators. |
| Priority | High |
| Dependencies | Actuator interface and control algorithms within SWCs |

# 3.2.2 Non-Functional SRS

|  |  |
| --- | --- |
| ID | SRS\_PER\_13207 |
| Description | The system shall ensure that the latency in processing sensor data for **brake** input does not exceed 5 milliseconds. |
| Rationale | Low sensor data latency is critical for real-time responsiveness. |
| **Performance Criteria** | The system shall process sensor data with minimal delay to maintain system responsiveness. |
| Priority | Medium |
| Dependencies | Sensor data processing efficiency. |

|  |  |
| --- | --- |
| ID | SRS\_PER\_13208 |
| Description | The system shall ensure that the latency in processing sensor data for **accelerator** input does not exceed 5 milliseconds. |
| Rationale | Low sensor data latency is critical for real-time responsiveness. |
| **Performance Criteria** | The system shall process sensor data with minimal delay to maintain system responsiveness. |
| Priority | Medium |
| Dependencies | Sensor data processing efficiency. |

## 4. Security Requirements

### 4.1 Authentication

# 4.1.1. Functional SRS

|  |  |
| --- | --- |
| ID | SRS\_SEC\_14101 |
| Description | The system shall include a seed key-based diagnostic service. |
| Rationale | This requirement ensures the presence of a diagnostic service for system maintenance and calibration data changes. |
| Use Case | The diagnostic service is available for users to initiate calibration data changes in Protected Engineering Mode using a valid seed key. |
| Priority | High |
| Dependencies | The availability of Protected Engineering Mode and proper seed key management. |

### 4.2 Data Integrity

# 4.2.1. Functional SRS

|  |  |
| --- | --- |
| ID | SRS\_SEC\_14201 |
| Description | The seed key-based diagnostic service shall be capable of changing NVM calibration data. |
| Rationale | This requirement specifies the capability of the diagnostic service to modify NVM calibration data. |
| Use Case | Authorized users can utilize the diagnostic service to modify the calibration data securely. |
| Priority | High |
| Dependencies | Proper implementation of the NVM calibration data modification mechanism. |

# 4.2.1. Non-Functional SRS

|  |  |
| --- | --- |
| ID | SRS\_SEC\_14202 |
| Description | The seed key-based diagnostic service shall operate in Protected Engineering Mode. |
| Rationale | Operating in Protected Engineering Mode ensures that only authorized personnel can access and use the diagnostic service. |
| Use Case | The service is accessible only when the system is in Protected Engineering Mode, typically requiring authentication and authorization. |
| Priority | High |
| Dependencies | Availability and proper functionality of Protected Engineering Mode. |

|  |  |
| --- | --- |
| ID | SRS\_SEC\_14203 |
| Description | The **input** to the seed key-based diagnostic service shall consist of 4 bytes. |
| Rationale | This requirement defines the expected size of the input data, ensuring compatibility with the service's design. |
| Use Case | When invoking the seed key-based diagnostic service, the user must provide a 4-byte input key as part of the request. |
| Priority | High |
| Dependencies | None |

|  |  |
| --- | --- |
| ID | SRS\_SEC\_14204 |
| Description | The **output** from the seed key-based diagnostic service shall consist of 4 bytes. |
| Rationale | This requirement specifies the size of the output data, ensuring consistency and compatibility with the service's design. |
| Use Case | After successfully executing the seed key-based diagnostic service, the system shall provide a 4-byte output as a result. |
| Priority | High |
| Dependencies | None |

|  |  |
| --- | --- |
| ID | SRS\_SEC\_14205 |
| Description | The seed key-based diagnostic service shall execute within specified time limits to ensure efficient modification of NVM calibration data. |
| Rationale | This requirement ensures that the service operates with acceptable response times, minimizing system downtime during calibration data updates. |
| Use Case | The system receives a request to modify NVM calibration data using the seed key-based diagnostic service. The service executes within the specified time limits, allowing the system to quickly apply the requested changes and return control to the user or application. |
| Priority | High |
| Dependencies | This requirement depends on the efficient implementation of the seed key validation and NVM modification algorithms, as well as the performance characteristics of the underlying hardware and software components. |

### 4.3 Access Control

# 4.3.1. Non-Functional SRS

|  |  |
| --- | --- |
| ID | SRS\_SEC\_14301 |
| Description | The seed key-based diagnostic service shall authenticate users before allowing NVM calibration data changes. |
| Rationale | This requirement ensures that only authorized users can access and use the diagnostic service for calibration data changes. |
| Use Case | When initiating the diagnostic service, the system verifies the authenticity of the provided seed key and grants access only if the key is valid. |
| Priority | High |
| Dependencies | Implementation of secure authentication mechanisms. |

|  |  |
| --- | --- |
| ID | SRS\_SEC\_14302 |
| Description | The seed key-based diagnostic service shall use strong encryption methods for data protection. |
| Rationale | This requirement specifies the use of robust encryption techniques to protect data during transmission and storage. |
| Use Case | The diagnostic service ensures that sensitive data is securely handled using encryption. |
| Priority | High |
| Dependencies | Implementation of encryption algorithms. |

|  |  |
| --- | --- |
| ID | SRS\_SEC\_14303 |
| Description | The seed key-based diagnostic service shall restrict access to authorized personnel only. |
| Rationale | This requirement ensures that access to the diagnostic service is limited to authorized personnel. |
| Use Case | Only designated individuals or roles can access and use the diagnostic service. |
| Priority | High |
| Dependencies | Implementation of access control mechanisms. |

|  |  |
| --- | --- |
| ID | SRS\_SEC\_14304 |
| Description | The seed key-based diagnostic service shall prevent unauthorized access to the NVM calibration data. |
| Rationale | This requirement guarantees the integrity and security of the NVM calibration data by preventing unauthorized access. |
| Use Case | The diagnostic service enforces access controls to prevent unauthorized users from modifying critical calibration data. |
| Priority | High |
| Dependencies | Implementation of access control mechanisms for NVM data. |

## 5. Safety and Performance Requirements

### 5.1 Prioritization

# 5.1.1. Functional SRS

|  |  |
| --- | --- |
| ID | SRS\_BSW\_15101 |
| Description | When both brake and accelerator pedals are simultaneously pressed, the system shall prioritize brake signal processing. |
| Rationale | Prioritizing brake signals ensures safety in cases of concurrent pedal presses. |
| Use Case | When the system detects both pedals are pressed, it prioritizes brake signal processing. |
| Priority | High |
| Dependencies | Real-time signal processing algorithms. |

# 5.1.2. Non-Functional SRS

|  |  |
| --- | --- |
| ID | SRS\_BSW\_15102 |
| Description | The system's brake signal prioritization mechanism shall respond within 10 milliseconds of detecting simultaneous brake and accelerator pedal presses |
| Rationale | Ensuring a rapid response to prioritize brake signals is critical for safety. |
| **Performance Criteria** | The prioritization mechanism must activate the brake signal within 10 milliseconds |
| Priority | High |
| Dependencies | Real-time signal processing and prioritization algorithms. |

|  |  |
| --- | --- |
| ID | SRS\_BSW\_15103 |
| Description | The system's brake signal prioritization mechanism shall prevent conflicts or race conditions between brake and accelerator signals. |
| Rationale | Preventing conflicts ensures proper prioritization and safe operation. |
| **Performance Criteria** | The mechanism must be designed to avoid conflicts or race conditions between brake and accelerator signals. |
| Priority | High |
| Dependencies | Careful design and testing of prioritization logic. |

### 5.2 Initialization

# 5.2.1. Functional SRS

|  |  |
| --- | --- |
| ID | SRS\_BSW\_15201 |
| Description | The system shall initialize within 50 milliseconds after power-up. |
| Rationale | Ensuring a fast initialization time after power-up is critical for reducing system downtime and enabling quick usability. |
| **Performance Criteria** | Upon power-up, the system must complete its initialization processes and be fully operational within 50 milliseconds. |
| Priority | High |
| Dependencies | Proper design and implementation of initialization routines and hardware components. |

### 5.3 Voltage Monitoring

To prevent potential issues, the system continuously monitors input voltage. It generates an error if the voltage exceeds 12V or drops below 4V, promoting safe operation and preventing electrical damage.

# 5.3.1. Functional SRS

|  |  |
| --- | --- |
| ID | SRS\_BSW\_15301 |
| Description | The system shall continuously and at a rate of [specify rate, e.g., 100 ms] monitor the input voltage. |
| Rationale | Continuous monitoring at a specified rate ensures real-time awareness of voltage changes. |
| Use Case | The system samples and checks the input voltage at the specified rate. |
| Priority | High |
| Dependencies | Power Control Module. |

|  |  |
| --- | --- |
| ID | SRS\_BSW\_15302 |
| Description | The system the system shall transmit the monitored input voltage value onto the communication bus in accordance with [specify communication protocol, e.g., CAN]. have a predefined initialization sequence. |
| Rationale | Precise communication is essential for other system components to access voltage data accurately. |
| Use Case | The system periodically sends the voltage value using the specified communication module and protocol. |
| Priority | High |
| Dependencies | Communication module driver software. |

|  |  |
| --- | --- |
| ID | SRS\_BSW\_15303 |
| Description | The system shall generate an error if the monitored voltage exceeds 12V. |
| Rationale | Detecting overvoltage conditions is critical for system protection. |
| Use Case | If the voltage exceeds 12V, the system triggers an error state. |
| Priority | High |
| Dependencies | Power Control Module. |

|  |  |
| --- | --- |
| ID | SRS\_BSW\_15304 |
| Description | The system shall generate an error if the monitored voltage is less than 4V. |
| Rationale | Detecting undervoltage conditions is critical for system protection. |
| Use Case | If the voltage less than 4v, the system triggers an error state. |
| Priority | High |
| Dependencies | Power Control Module. |

# 5.3.2. Non-Functional SRS

|  |  |
| --- | --- |
| ID | SRS\_BSW\_15305 |
| Description | The system shall monitor the input voltage with an accuracy of [specify accuracy, e.g., 0.1V] using [specify ADC settings]. |
| Rationale | High monitoring accuracy ensures precise voltage data. |
| Use Case | The system's voltage monitoring meets the specified accuracy. |
| Priority | High |
| Dependencies | ADC peripheral configuration and driver software. |

|  |  |
| --- | --- |
| ID | SRS\_BSW\_15306 |
| Description | The system shall handle voltage threshold errors gracefully, ensuring system stability and recovery when errors occur. |
| Rationale | Reliable error handling prevents system failures due to voltage issues. |
| Use Case | When an error is generated, the system takes appropriate action, such as logging the error and attempting to recover. |
| Priority | Medium |
| Dependencies | Error handling code and recovery mechanisms. |

|  |  |
| --- | --- |
| ID | SRS\_BSW\_15307 |
| Description | The system shall validate the input voltage within the specified safe range of 4V to 12V. |
| Rationale | Ensuring that the input voltage is within the safe operating range is vital for system safety. |
| Use Case | The system checks the voltage before transmitting it or taking any action. |
| Priority | High |
| Dependencies | Range validation code and safety-critical software modules. |

### 5.4 Response Times

# 5.4.1. Functional SRS

|  |  |
| --- | --- |
| ID | SRS\_PER\_15400 |
| Description | The system shall ensure that safety-critical requirements achieve a response time of no more than 2% of their specified values. |
| Rationale | Ensuring a rapid response for safety-critical functions is essential for system safety. |
| Use Case | The system continuously monitors and enforces the response time for safety-critical requirements. |
| Priority | High |
| Dependencies | Safety-critical requirement specifications |

|  |  |
| --- | --- |
| ID | SRS\_PER\_15401 |
| Description | The system shall ensure that non-safety requirements achieve a response time of no more than 5% of their specified values. |
| Rationale | Maintaining a reasonable response time for non-safety functions enhances system performance. |
| Use Case | The system continuously monitors and enforces the response time for non-safety requirements. |
| Priority | High |
| Dependencies | Non-safety requirement specifications. |

|  |  |
| --- | --- |
| ID | SRS\_MCAL\_15402 |
| Description | The system shall enable a filter for the brake signal. |
| Rationale | Enabling a filter on the brake signal helps remove noise and ensures signal stability. |
| Use Case | The system applies a specified filter to the brake signal. |
| Priority | High |
| Dependencies | Brake signal filter configuration. |

|  |  |
| --- | --- |
| ID | SRS\_MCAL\_15403 |
| Description | The system shall ensure that the brake signal responds within a maximum time of 20ms. |
| Rationale | Achieving a rapid response time for the brake signal is essential for safety and performance. |
| Use Case | The system continuously monitors the brake signal's response time. |
| Priority | High |
| Dependencies | Brake signal response time monitoring mechanisms. |

# 5.4.2. Non-Functional SRS

|  |  |
| --- | --- |
| ID | SRS\_PER\_15404 |
| Description | The system shall implement performance monitoring mechanisms to measure and enforce the response time of safety requirements. |
| Rationale | Performance monitoring ensures that safety-critical functions meet their response time criteria. |
| Use Case | The system collects and analyzes performance data for safety requirements. |
| Priority | Medium |
| Dependencies | Performance monitoring software, safety-critical requirement specifications. |

|  |  |
| --- | --- |
| ID | SRS\_PER\_15405 |
| Description | The system shall implement performance monitoring mechanisms to measure and enforce the response time of non-safety requirements. |
| Rationale | Performance monitoring ensures that non-safety functions meet their response time criteria. |
| Use Case | The system collects and analyzes performance data for non-safety requirements. |
| Priority | Medium |
| Dependencies | Performance monitoring software, non-safety requirement specifications. |

|  |  |
| --- | --- |
| ID | SRS\_PER\_15406 |
| Description | The system shall handle response time violations gracefully, logging errors and taking corrective actions. |
| Rationale | Handling response time violations ensures system stability and reliability. |
| Use Case | When a violation occurs, the system logs the error, attempts to recover, and may trigger appropriate notifications. |
| Priority | High |
| Dependencies | Error handling code, performance monitoring mechanisms. |

|  |  |
| --- | --- |
| ID | SRS\_CAL\_15407 |
| Description | The system shall implement configurable brake signal filtering parameters to adapt to different operational conditions. |
| Rationale | Configurable filtering parameters allow the system to adapt to varying signal characteristics. |
| Use Case | Users can configure the filter settings for the brake signal. |
| Priority | Medium |
| Dependencies | Brake signal filter configuration software. |

### 5.5 Resource Utilization

# 5.5.1. Functional SRS

|  |  |
| --- | --- |
| ID | SRS\_OS\_15501 |
| Description | The system shall implement CPU load management to ensure that CPU utilization remains below 80%. |
| Rationale | Managing CPU load is critical for maintaining system responsiveness and stability. |
| Use Case | The system continuously monitors CPU utilization and takes actions to control it within the specified limit. |
| Priority | High |
| Dependencies | CPU load monitoring and management software. |

|  |  |
| --- | --- |
| ID | SRS\_DIAG\_15502 |
| Description | The system shall implement ROM usage monitoring to ensure that ROM utilization remains below 70%. |
| Rationale | Monitoring ROM usage is essential to prevent memory-related issues and ensure sufficient space for program code. |
| Use Case | The system periodically checks the ROM usage and triggers actions if it approaches or exceeds the specified limit. |
| Priority | High |
| Dependencies | ROM usage monitoring software. |

|  |  |
| --- | --- |
| ID | SRS\_DIAG\_15503 |
| Description | The system shall implement NVM (Non-Volatile Memory) usage monitoring to ensure that NVM utilization remains below 70%. |
| Rationale | Monitoring NVM usage is critical to prevent storage-related problems and ensure adequate space for data storage. |
| Use Case | The system periodically checks the NVM usage and initiates actions if it approaches or exceeds the specified limit. |
| Priority | High |
| Dependencies | NVM usage monitoring software. |

### Fault Handling

**5.6.1. Functional\_SRS**

|  |  |
| --- | --- |
| ID | SRS\_DIAG\_15601 |
| Description | The system shall implement diagnostic fault detection mechanisms to identify faults within the embedded system. |
| Rationale | Detecting faults is essential for system reliability. |
| Use Case | The system continuously monitors various aspects, including sensors and internal states, to identify faults. |
| Priority | High |
| Dependencies | Diagnostic fault detection algorithms and sensors. |

|  |  |
| --- | --- |
| ID | SRS\_DIAG\_15602 |
| Description | Upon detecting a fault, the system shall isolate the faulted component and generate a detailed fault report for analysis. |
| Rationale | Effective fault isolation and reporting are essential for system maintenance and safety. |
| Use Case | When a fault is detected, the system isolates the affected component and generates a report including fault details. |
| Priority | High |
| Dependencies | Fault isolation and reporting procedures |

|  |  |
| --- | --- |
| ID | SRS\_ECU\_15603 |
| Description | Each Electronic Control Unit (ECU) shall include mechanisms for isolating faults within the ECU and recovering to normal operation. |
| Rationale | Swift ECU fault isolation and recovery are essential for maintaining the ECU's functionality. |
| Use Case | Upon detecting a fault within the ECU, the ECU initiates procedures to isolate the fault and recover to a stable state. |
| Priority | High |
| Dependencies | ECU-specific fault isolation and recovery mechanisms. |

|  |  |
| --- | --- |
| ID | SRS\_COMM\_15604 |
| Description | The communication subsystem shall implement fault handling mechanisms to address communication faults and maintain critical communication functions. |
| Rationale | Fault handling in the communication subsystem ensures reliable communication even in fault-prone conditions. |
| Use Case | In the presence of communication faults, the subsystem takes actions to maintain essential communication functions. |
| Priority | High |
| Dependencies | Fault handling strategies within the communication subsystem. |

|  |  |
| --- | --- |
| ID | SRS\_DIAG\_15605 |
| Description | The system shall implement fault detection mechanisms to identify faulty ADC values for brake and accelerator inputs. |
| Rationale | Detecting faulty ADC values is crucial for ensuring safe and reliable operation. |
| Use Case | The system continuously monitors brake and accelerator inputs, flagging values exceeding 95% or falling below 5% as faulty. |
| Priority | High |
| Dependencies | Fault detection algorithms for ADC inputs. |

|  |  |
| --- | --- |
| ID | SRS\_DIAG\_15606 |
| Description | Upon detecting a fault, the system shall isolate the faulted component and generate a detailed fault report for analysis. |
| Rationale | Effective fault isolation and reporting are essential for system maintenance and safety. |
| Use Case | When a fault is detected, the system isolates the affected component and generates a report including fault details. |
| Priority | High |
| Dependencies | Fault isolation and reporting procedures |

|  |  |
| --- | --- |
| ID | SRS\_DIAG\_15607 |
| Description | Upon detecting faulty ADC values for brake and accelerator inputs, the system shall trigger appropriate actions to mitigate potential risks. |
| Rationale | Handling ADC value faults is critical to maintain safety and prevent adverse consequences. |
| Use Case | When a faulty ADC value is detected, the system initiates predefined actions, which may include alerting the driver, entering a safe mode, or performing corrective actions. |
| Priority | High |
| Dependencies | Fault handling procedures for ADC value faults. |

|  |  |
| --- | --- |
| ID | SRS\_DIAG\_15608 |
| Description | The system shall generate clear and descriptive error messages or indicators to promptly notify users or technicians about the occurrence of faults. |
| Rationale | Clear fault notifications are essential for enabling quick diagnosis and resolution of issues, enhancing system safety and reliability. |
| Use Case | When a fault is detected and requires attention, the system shall display error messages on the user interface and activate fault indicators as appropriate. |
| Priority | High |
| Dependencies | User interface components, fault indicator mechanisms. |

|  |  |
| --- | --- |
| ID | SRS\_DIAG\_15609 |
| Description | The system shall implement fault recovery mechanisms designed to facilitate the recovery from faults and the resumption of normal operation whenever possible. |
| Rationale | Fault recovery is crucial for minimizing system downtime and ensuring uninterrupted functionality. |
| Use Case | When a fault is detected, the system initiates predefined fault recovery procedures to restore normal operation, if feasible. |
| Priority | High |
| Dependencies | Fault recovery algorithms, procedures, and system state management. |

|  |  |
| --- | --- |
| ID | SRS\_DIAG\_15610 |
| Description | In Engineering Mode, the system shall grant authorized technicians access to fault logs for diagnostic purposes, ensuring they can identify and address system issues. |
| Rationale | Access control in Engineering Mode is critical to protect sensitive diagnostic data while enabling troubleshooting. |
| Use Case | Authorized technicians can access fault logs specifically when the system is in Engineering Mode. |
| Priority | High |
| Dependencies | Access control mechanisms for Engineering Mode and fault log accessibility. |

|  |  |
| --- | --- |
| ID | SRS\_SEC\_15611 |
| Description | The system shall provide a designated user interface feature that allows authorized users to activate Engineering Mode with proper authentication and authorization. |
| Rationale | Activating Engineering Mode should require appropriate user authentication and authorization. |
| Use Case | Authorized users can activate Engineering Mode using the designated user interface feature. |
| Priority | Medium |
| Dependencies | User authentication, authorization, and Engineering Mode activation controls. |

|  |  |
| --- | --- |
| ID | SRS\_DIAG\_15612 |
| Description | The Diagnostics module shall implement NVM fault detection mechanisms using CRC checks to identify and handle potential data corruption issues. |
| Rationale | Detecting NVM issues through CRC checks is crucial for maintaining data integrity and system reliability. |
| Use Case | The system periodically performs CRC checks on NVM data, and upon detection of potential data corruption, appropriate actions are initiated for data recovery or error handling. |
| Priority | High |
| Dependencies | Integration with NVM management and CRC check mechanisms. |

|  |  |
| --- | --- |
| ID | SRS\_BSW\_15613 |
| Description | The system shall provide a designated user interface feature that allows authorized users to activate Engineering Mode with proper authentication and authorization. |
| Rationale | Activating Engineering Mode should require appropriate user authentication and authorization. |
| Use Case | Upon detecting NVM issues through CRC checks, the BSW initiates data recovery processes to restore corrupted data to a consistent state. |
| Priority | High |
| Dependencies | Integration with NVM management and fault handling mechanisms. |

|  |  |
| --- | --- |
| ID | SRS\_DIAG\_15614 |
| Description | The system shall log NVM fault occurrences, CRC check results, timestamps, and relevant diagnostic information for post-fault analysis and troubleshooting. |
| Rationale | Logging NVM faults and CRC check results aids in understanding the nature of data corruption and supports system improvement. |
| Use Case | When NVM faults are detected through CRC checks, the system logs relevant information for analysis and potential recovery. |
| Priority | Medium |
| Dependencies | Logging mechanisms for NVM faults and CRC check results. |

|  |  |
| --- | --- |
| ID | SRS\_BSW\_15615 |
| Description | The Basic Software (BSW) shall implement CRC protection mechanisms for NVM data to detect and prevent corruption. |
| Rationale | CRC protection is essential to ensure the integrity of NVM data and prevent corruption. |
| Use Case | The system uses CRC checks to validate NVM data integrity during read and write operations, preventing corrupt data from being used. |
| Priority | High |
| Dependencies | Integration with NVM management and CRC check mechanisms. |

|  |  |
| --- | --- |
| ID | SRS\_BSW\_15616 |
| Description | The Diagnostics module shall include functionality for CRC error detection on NVM data, triggering appropriate actions upon CRC check failures. |
| Rationale | Detecting CRC errors on NVM data is critical for identifying and responding to data corruption issues. |
| Use Case | When CRC checks on NVM data fail, the Diagnostics module initiates fault handling procedures, such as alerting the system or logging the error. |
| Priority | High |
| Dependencies | Integration with NVM management and CRC error detection mechanisms. |

|  |  |
| --- | --- |
| ID | SRS\_DIAG\_15617 |
| Description | The system shall log CRC errors detected on NVM data, including details, timestamps, and relevant diagnostic information, for post-fault analysis and troubleshooting. |
| Rationale | Logging CRC errors on NVM data supports the analysis of data corruption incidents and system improvement |
| Use Case | The system performs CRC checks while maintaining optimal performance for NVM data operations. |
| Priority | Medium |
| Dependencies | Optimization of CRC check algorithms and execution. |

|  |  |
| --- | --- |
| ID | SRS\_BSW\_15618 |
| Description | The Basic Software (BSW) shall include mechanisms to activate a fault mode in the event of system faults, ensuring safe operation and preventing incorrect behavior. |
| Rationale | Activating a fault mode is critical for maintaining safety and preventing further system issues during fault conditions. |
| Use Case | When a fault is detected, the BSW activates a predefined fault mode to ensure safe system operation. |
| Priority | High |
| Dependencies | Integration with fault detection mechanisms and fault mode definition. |

|  |  |
| --- | --- |
| ID | SRS\_OS\_15619 |
| Description | The Operating System (OS) shall provide support for handling and managing fault modes, allowing for a controlled transition into fault mode when necessary. |
| Rationale | The OS plays a crucial role in orchestrating the transition to fault mode and ensuring safe operation. |
| Use Case | When a fault is detected, the OS manages the transition into fault mode while maintaining system integrity. |
| Priority | High |
| Dependencies | Integration with OS fault handling capabilities. |

|  |  |
| --- | --- |
| ID | SRS\_DIAG\_15620 |
| Description | The Diagnostics module shall include functionality for reporting fault mode activations, providing diagnostic information for post-fault analysis and troubleshooting. |
| Rationale | Reporting fault mode activations helps in understanding the nature of faults and supports system improvement. |
| Use Case | When the system enters a fault mode, the Diagnostics module generates a report with relevant information for analysis. |
| Priority | Medium |
| Dependencies | Integration with fault mode reporting mechanisms. |

|  |  |
| --- | --- |
| ID | SRS\_SEC\_15621 |
| Description | The system shall ensure that the activation of the fault mode is performed securely, preventing unauthorized or malicious activations. |
| Rationale | Securing fault mode activation is crucial to prevent unauthorized or malicious triggering of fault modes. |
| Use Case | Only authorized entities can activate fault mode, and mechanisms are in place to prevent unauthorized activations. |
| Priority | Medium |
| Dependencies | Security mechanisms for fault mode activation. |

**5.6.2. Non-Functional\_SRS**

|  |  |
| --- | --- |
| ID | SRS\_DIAG\_15622 |
| Description | Error messages generated by the system shall be user-friendly, concise, and in plain language to ensure users and technicians can easily understand and interpret them. |
| Rationale | User-friendly error messages enhance the user experience and facilitate effective fault diagnosis. |
| Use Case | Error messages shall avoid technical jargon and provide clear instructions or information about the fault's nature and potential actions to be taken. |
| Priority | Medium |
| Dependencies | User interface design guidelines for error messages. |

|  |  |
| --- | --- |
| ID | SRS\_SEC\_15623 |
| Description | The system's fault recovery mechanisms shall ensure a specified maximum recovery time, minimizing disruption to normal operation. |
| Rationale | Swift recovery is essential for maintaining system reliability and minimizing downtime. |
| Use Case | The system aims to recover from faults within the specified maximum time frame. |
| Priority | Medium |
| Dependencies | Fault recovery performance measurements and optimization. |

|  |  |
| --- | --- |
| ID | SRS\_SEC\_15624 |
| Description | The system shall maintain a specified minimum recovery success rate, ensuring the effectiveness of fault recovery mechanisms. |
| Rationale | A high recovery success rate is critical for system reliability. |
| Use Case | The system aims to successfully recover from faults in accordance with the specified rate. |
| Priority | Medium |
| Dependencies | Fault recovery success criteria and performance monitoring. |

|  |  |
| --- | --- |
| ID | SRS\_SEC\_15625 |
| Description | The fault logs shall be accessible to authorized users or technicians through secure and controlled means. |
| Rationale | Controlled access ensures that only authorized personnel can view and analyze fault logs. |
| Use Case | Access to fault logs is restricted to authorized users, and mechanisms for secure access control are in place. |
| Priority | Medium |
| Dependencies | Access control mechanisms, user authentication. |

|  |  |
| --- | --- |
| ID | SRS\_CAL\_15626 |
| Description | The system shall define a log retention period, specifying how long fault logs shall be retained before automatic deletion. |
| Rationale | Defining a retention period helps manage storage resources and comply with data retention policies. |
| Use Case | Fault logs are retained for the defined period, after which they are automatically deleted or archived as required. |
| Priority | Medium |
| Dependencies | Logging retention policy and mechanisms. |

## 6. Communication and Data Handling

### 6.1 Communication

## 6.1.1. Functional\_SRS

|  |  |
| --- | --- |
| ID | SRS\_COMM\_16101 |
| Description | The system shall implement the UART (Universal Asynchronous Receiver-Transmitter) communication protocol to enable data exchange between automotive ECUs. |
| Rationale | Implementing the UART protocol is essential for enabling serial communication between ECUs, which is a common standard in automotive systems. |
| Use Case | In an automotive control system, ECUs use UART to exchange sensor data and control commands. |
| Priority | High |
| Dependencies | Depends on the design and implementation of the UART communication protocol. |

|  |  |
| --- | --- |
| ID | SRS\_COMM \_16102 |
| Description | The system shall support mechanisms for identifying individual ECUs within the communication network. |
| Rationale | ECU identification ensures that data is sent to and received from the intended ECU, preventing data mismatches and errors. |
| Use Case | In an automotive network, each ECU must be uniquely identified to send commands or receive data from specific ECUs. |
| Priority | High |
| Dependencies | Depends on the implementation of ECU identification mechanisms. |

|  |  |
| --- | --- |
| ID | SRS\_COMM\_16103 |
| Description | The system shall enable the transmission of data packets between automotive ECUs using UART. |
| Rationale | Data transmission is the core functionality, allowing ECUs to exchange critical information for vehicle operation. |
| Use Case | In a modern car, ECUs use UART to transmit engine sensor data to the central engine control unit (ECU) for processing. |
| Priority | High |
| Dependencies | Depends on the implementation of data transmission mechanisms and the UART protocol. |

|  |  |
| --- | --- |
| ID | SRS\_COMM\_16104 |
| Description | The ECUs shall have the capability to connect through GSM or Wifi for remote monitoring. |
| Rationale | Enabling GSM connectivity is essential for remote monitoring, ensuring a wider range of coverage. |
| Use Case | The system establishes a GSM connection to enable remote monitoring in areas without WiFi coverage. |
| Priority | High |
| Dependencies | Depends on the implementation of data transmission mechanisms and the UART protocol. |

## 6.1.2. Non-Functional\_SRS

|  |  |
| --- | --- |
| ID | SRS\_SEC\_16105 |
| Description | The system shall ensure data integrity during UART communication, minimizing data corruption or loss. |
| Rationale | Data integrity is crucial to prevent communication errors that could impact vehicle performance and safety. |
| Use Case | In an automotive safety system, data integrity ensures that sensor data for features like collision avoidance is reliable. |
| Priority | High |
| Dependencies | Depends on the implementation of error-checking and error-correction mechanisms in the UART communication. |

|  |  |
| --- | --- |
| ID | SRS\_OS\_16106 |
| Description | The system shall support real-time communication between ECUs using UART to meet timing requirements. |
| Rationale | Real-time communication ensures that critical commands and data are processed within specified time frames. |
| Use Case | In an advanced driver-assistance system, real-time communication is necessary to enable timely responses to sensor data. |
| Priority | High |
| Dependencies | Depends on the implementation of real-time communication mechanisms in the UART protocol and system. |

### 6.2 Sensor Data

## 6.2.1. Functional\_SRS

|  |  |
| --- | --- |
| ID | SRS\_ SWC\_16201 |
| Description | The system shall read accelerator sensor values from the connected ECUs. |
| Rationale | This requirement ensures that the system can acquire essential data from the vehicle's sensors for further processing and display. |
| Use Case | In an automotive dashboard system, the system must read sensor data to provide real-time feedback to the driver. |
| Priority | High |
| Dependencies | Depends on the availability and compatibility of the sensor interfaces. |

|  |  |
| --- | --- |
| ID | SRS\_ SWC\_16202 |
| Description | The system shall read brake sensor values from the connected ECUs. |
| Rationale | This requirement ensures that the system can acquire essential data from the vehicle's sensors for further processing and display. |
| Use Case | In an automotive dashboard system, the system must read sensor data to provide real-time feedback to the driver. |
| Priority | High |
| Dependencies | Depends on the availability and compatibility of the sensor interfaces. |

|  |  |
| --- | --- |
| ID | SRS\_ SWC\_16203 |
| Description | The system shall process the incoming brake sensor data to derive relevant information. |
| Rationale | Data processing is essential to interpret raw sensor data and extract meaningful information for display. |
| Use Case | In a vehicle control system, processing sensor data allows the system to calculate braking force or accelerator pedal position. |
| Priority | High |
| Dependencies | Depends on the availability of processing capabilities and algorithms. |

|  |  |
| --- | --- |
| ID | SRS\_ SWC\_16204 |
| Description | The system shall process the incoming accelerator sensor data to derive relevant information. |
| Rationale | Data processing is essential to interpret raw sensor data and extract meaningful information for display. |
| Use Case | In a vehicle control system, processing sensor data allows the system to calculate accelerator pedal position. |
| Priority | High |
| Dependencies | Depends on the availability of processing capabilities and algorithms. |

|  |  |
| --- | --- |
| ID | SRS\_SWC\_16205 |
| Description | The system shall display processed brake sensor data on the Human-Machine Interface (HMI). |
| Rationale | Displaying sensor data on the HMI provides real-time feedback to users, such as drivers or operators. |
| Use Case | In an automotive infotainment system, displaying brake data allows the driver to monitor vehicle performance. |
| Priority | High |
| Dependencies | Depends on the availability and functionality of the HMI. |

|  |  |
| --- | --- |
| ID | SRS\_SWC\_16206 |
| Description | The system shall display processed accelerator sensor data on the Human-Machine Interface (HMI). |
| Rationale | Displaying sensor data on the HMI provides real-time feedback to users, such as drivers or operators. |
| Use Case | In an automotive infotainment system, displaying accelerator data allows the driver to monitor vehicle performance. |
| Priority | High |
| Dependencies | Depends on the availability and functionality of the HMI. |

## 6.2.2. Non-Functional\_SRS

|  |  |
| --- | --- |
| ID | SRS\_PER\_16207 |
| Description | The system shall process sensor data in real-time to ensure timely and responsive display on the HMI. |
| Rationale | Real-time data processing is essential for providing immediate feedback to users, particularly in safety-critical applications. |
| Use Case | In an autonomous vehicle system, real-time data processing ensures rapid response to sensor inputs for collision avoidance. |
| Priority | High |
| Dependencies | Depends on the system's processing capabilities and performance. |

|  |  |
| --- | --- |
| ID | SRS\_CAL\_16208 |
| Description | The system shall maintain high data accuracy when processing and displaying accelerator sensor values. |
| Rationale | Data accuracy is critical to ensure that the displayed information reflects the true state of the vehicle's acceleration. |
| Use Case | In a vehicle control system, data accuracy is crucial for driver confidence and safety. |
| Priority | High |
| Dependencies | Depends on the accuracy of the sensor inputs and the processing algorithms. |

|  |  |
| --- | --- |
| ID | SRS\_CAL\_16209 |
| Description | The system shall maintain high data accuracy when processing and displaying brake sensor values. |
| Rationale | Data accuracy is critical to ensure that the displayed information reflects the true state of the vehicle's braking. |
| Use Case | In a vehicle control system, data accuracy is crucial for driver confidence and safety. |
| Priority | High |
| Dependencies | Depends on the accuracy of the sensor inputs and the processing algorithms. |

### 6.3 Frame Structures

## 6.3.1. Functional\_SRS

|  |  |
| --- | --- |
| ID | SRS\_SWC\_16301 |
| Description | The HMI shall include an input interface for measuring voltage. |
| Rationale | Voltage measurement is a fundamental parameter in many applications, including monitoring the power supply or battery voltage. |
| Use Case | In an energy management system, the HMI uses ADC1 to display the current battery voltage. |
| Priority | High |
| Dependencies | Depends on the availability of voltage measurement hardware and integration. |

|  |  |
| --- | --- |
| ID | SRS\_SWC\_16302 |
| Description | The HMI shall include an input interface for monitoring accelerator pedal position. |
| Rationale | Accelerator input provides critical information for vehicle control and performance monitoring. |
| Use Case | In an automotive dashboard, ADC2 is used to display the accelerator pedal position. |
| Priority | High |
| Dependencies | Depends on the availability of accelerator sensor hardware and integration. |

|  |  |
| --- | --- |
| ID | SRS\_SWC\_16303 |
| Description | The HMI shall include an input interface for monitoring brake pedal position. |
| Rationale | Brake input is essential for safety-critical applications, such as determining braking force. |
| Use Case | In an automotive safety system, ADC3 is used to display brake pedal position and trigger appropriate warnings. |
| Priority | High |
| Dependencies | Depends on the availability of brake sensor hardware and integration. |

|  |  |
| --- | --- |
| ID | SRS\_USER\_INTERFACE\_16304 |
| Description | The HMI shall include a separate LED indicator for displaying the status of the left brake. |
| Rationale | Separate LED indicators provide clear visual feedback for specific functions or conditions. |
| Use Case | In an automotive dashboard, the left brake LED indicator signals when the left brake is engaged. |
| Priority | High |
| Dependencies | Depends on the availability of LED hardware and integration. |

|  |  |
| --- | --- |
| ID | SRS\_USER\_INTERFACE \_16305 |
| Description | The HMI shall include a separate LED indicator for displaying the status of the right brake. |
| Rationale | Separate LED indicators ensure distinct representation of different brake conditions. |
| Use Case | In an automotive dashboard, the right brake LED indicator signals when the right brake is engaged. |
| Priority | High |
| Dependencies | Depends on the availability of LED hardware and integration. |

|  |  |
| --- | --- |
| ID | SRS\_COMM\_16306 |
| Description | The system shall define a frame structure for UART communication that includes a format for transmitting brake values. |
| Rationale | Defining a specific frame structure ensures that brake data can be reliably transmitted and interpreted by receiving ECUs. |
| Use Case | In an automotive control system, the frame structure for brake data allows ECUs to exchange real-time braking information. |
| Priority | High |
| Dependencies | Depends on the design and documentation of the frame structure. |

|  |  |
| --- | --- |
| ID | SRS\_COMM\_16316 |
| Description | The system shall define a frame structure for UART communication that includes a format for transmitting accelerator values. |
| Rationale | Defining a specific frame structure ensures that accelerator data can be reliably transmitted and interpreted by receiving ECUs. |
| Use Case | In an electric vehicle system, the frame structure for accelerator data enables the sharing of throttle pedal position information. |
| Priority | High |
| Dependencies | Depends on the design and documentation of the frame structure. |

## 6.3.2. Non-Functional\_SRS

|  |  |
| --- | --- |
| ID | SRS\_PER\_16301 |
| Description | The HMI shall establish a response time threshold of less than 20 milliseconds (ms) for displaying brake values. |
| Rationale | Defining a response time threshold provides a clear target for the system's performance, ensuring timely data presentation to the user. |
| Use Case | In an automotive instrument cluster, this threshold helps maintain a consistent and responsive user experience. |
| Priority | High |
| Dependencies | None |

|  |  |
| --- | --- |
| ID | SRS\_COMM\_16302 |
| Description | The system shall receive brake sensor values promptly upon sensor data transmission. |
| Rationale | Timely data reception is critical for achieving the specified response time threshold on the HMI. |
| Use Case | In an automotive dashboard, the system must quickly receive sensor data to meet the response time requirement. |
| Priority | High |
| Dependencies | Depends on the speed of data transmission from the sensors and the communication infrastructure. |

|  |  |
| --- | --- |
| ID | SRS\_COMM\_16303 |
| Description | The system shall receive accelerator sensor values promptly upon sensor data transmission. |
| Rationale | Timely data reception is critical for achieving the specified response time threshold on the HMI. |
| Use Case | In an automotive dashboard, the system must quickly receive sensor data to meet the response time requirement. |
| Priority | High |
| Dependencies | Depends on the speed of data transmission from the sensors and the communication infrastructure. |

|  |  |
| --- | --- |
| ID | SRS\_OS\_16304 |
| Description | The system shall process received sensor data and prepare it for display within a specified time frame. |
| Rationale | Efficient data processing is necessary to meet the response time threshold for displaying brake values. |
| Use Case | In a vehicle control system, rapid data processing ensures that the HMI can display real-time sensor information. |
| Priority | High |
| Dependencies | Depends on the system's processing capabilities and algorithms. |

|  |  |
| --- | --- |
| ID | SRS\_PER\_16305 |
| Description | The HMI shall render the processed sensor data and update the display within the defined response time threshold. |
| Rationale | The time taken for rendering and updating the display contributes to meeting the response time requirement. |
| Use Case | In an aircraft cockpit, the HMI must render and update critical sensor data, such as altitude, within the specified time frame. |
| Priority | High |
| Dependencies | Depends on the speed and efficiency of the display hardware and software. |

|  |  |
| --- | --- |
| ID | SRS\_OS\_16306 |
| Description | The system shall continuously monitor and optimize its components to ensure that the response time threshold is consistently met. |
| Rationale | Ongoing monitoring and optimization are necessary to maintain the system's performance over time. |
| Use Case | In a medical monitoring device, continuous optimization ensures that vital signs are displayed promptly to healthcare professionals. |
| Priority | High |
| Dependencies | Depends on the implementation of monitoring and optimization mechanisms. |

|  |  |
| --- | --- |
| ID | SRS\_SRS\_USER\_INTERFACE\_16307 |
| Description | The HMI's user interface shall provide responsive input and visual feedback to ensure a seamless user experience. |
| Rationale | Responsiveness is critical for user satisfaction and system usability. |
| Use Case | In a vehicle dashboard, responsive input and LED indicators enhance driver interaction and safety. |
| Priority | High |
| Dependencies | Depends on the design and implementation of the user interface components. |

|  |  |
| --- | --- |
| ID | SRS\_SEC\_16308 |
| Description | The defined frame structures shall incorporate mechanisms for ensuring data integrity and error handling during UART communication. |
| Rationale | Data integrity and error handling mechanisms are crucial for maintaining the reliability of transmitted data. |
| Use Case | In a safety-critical application like aerospace, data integrity and error handling are essential to prevent data corruption during transmission. |
| Priority | High |
| Dependencies | Depends on the inclusion of error-checking and error-correction mechanisms in the frame structure. |

|  |  |
| --- | --- |
| ID | SRS\_COMM\_16309 |
| Description | The defined frame structures shall be designed to ensure compatibility and interoperability with various ECUs and UART communication protocols. |
| Rationale | Compatibility and interoperability ensure that the frame structures can be used across different ECU configurations and communication standards. |
| Use Case | In an automotive network, compatibility and interoperability enable the frame structures to work with a wide range of ECUs and systems. |
| Priority | High |
| Dependencies | Depends on the consideration of industry-standard UART communication practices and ECU requirements. |

### 6.4 Firmware Updates

## 6.4.1. Functional\_SRS

|  |  |
| --- | --- |
| ID | SRS\_BSW\_16401 |
| Description | The Basic Software (BSW) shall implement validation checks as part of the flashing process to verify the integrity and authenticity of the firmware image before installation. |
| Rationale | Validation checks are essential to ensure that only legitimate and unaltered firmware images are installed, maintaining system integrity. |
| Use Case | During the flashing process, the BSW performs validation checks on the firmware image to confirm its integrity and authenticity. |
| Priority | High |
| Dependencies | Integration with firmware flashing and validation mechanisms. |

|  |  |
| --- | --- |
| ID | SRS\_SEC\_16402 |
| Description | The flashing process shall include authentication and integrity verification procedures to ensure that the firmware image has not been tampered with or altered. |
| Rationale | Authentication and integrity verification are critical for preventing the installation of compromised firmware. |
| Use Case | Before installation, the flashing process authenticates the firmware image and checks its integrity. |
| Priority | High |
| Dependencies | Security mechanisms for authentication and integrity verification. |

|  |  |
| --- | --- |
| ID | SRS\_STO\_16403 |
| Description | The system shall provide sufficient storage capacity to accommodate multiple firmware versions, ensuring that they are readily available for installation or rollback. |
| Rationale | Adequate storage capacity is necessary to store multiple firmware versions for recovery and maintenance purposes. |
| Use Case | The system allocates storage space for each firmware version, enabling easy access and management. |
| Priority | High |
| Dependencies | Storage management mechanisms for firmware versions. |

|  |  |
| --- | --- |
| ID | SRS\_BSW\_16404 |
| Description | The system shall provide a mechanism for users to initiate firmware version rollback to a previous version in case of issues with updated firmware., timestamps, and relevant diagnostic information, for post-event analysis and troubleshooting. |
| Rationale | User-initiated firmware version rollback is essential for resolving issues quickly. |
| Use Case | Users can select a previous firmware version and initiate a rollback to address problems with updated firmware. |
| Priority | High |
| Dependencies | Rollback mechanism and user interface features. |

|  |  |
| --- | --- |
| ID | SRS\_BSW\_16405 |
| Description | The Basic Software (BSW) shall include mechanisms to enforce version compatibility checks during firmware installation to ensure that only compatible firmware versions are installed. |
| Rationale | Version compatibility enforcement prevents the installation of firmware versions that may lead to system instability or errors. |
| Use Case | Before installing firmware, the BSW checks the compatibility of the selected version with the current system configuration and prevents incompatible installations. |
| Priority | High |
| Dependencies | Integration with version compatibility check mechanisms. |

|  |  |
| --- | --- |
| ID | SRS\_DIAG\_16406 |
| Description | The Diagnostics module shall include functionality for handling version compatibility errors, providing diagnostic information for troubleshooting and user guidance. |
| Rationale | Handling version compatibility errors helps users understand and resolve issues related to firmware installation. |
| Use Case | When a version compatibility check fails, the Diagnostics module generates error messages and provides guidance for resolution. |
| Priority | Medium |
| Dependencies | Integration with version compatibility error handling mechanisms. |

|  |  |
| --- | --- |
| ID | SRS\_DIAG\_16407 |
| Description | The system shall provide user-friendly notifications when version compatibility checks detect incompatible firmware versions, offering clear guidance on resolution. |
| Rationale | User-friendly notifications enhance the user experience and assist users in addressing compatibility issues. |
| Use Case | When an incompatible firmware version is selected, the system displays a notification with guidance on selecting a compatible version. |
| Priority | Medium |
| Dependencies | Integration with user interface features for notifications. |

|  |  |
| --- | --- |
| ID | SRS\_BSW\_16408 |
| Description | The Basic Software (BSW) shall implement mechanisms for verifying the integrity of the flashed firmware during runtime, using techniques such as checksum or CRC checks. |
| Rationale | Runtime firmware integrity verification ensures that the flashed firmware remains unaltered and reliable during system operation. |
| Use Case | Periodic runtime checks, such as checksum or CRC checks, are performed on the flashed firmware to detect any alterations or corruption. |
| Priority | High |
| Dependencies | Integration with runtime firmware integrity verification mechanisms. |

|  |  |
| --- | --- |
| ID | SRS\_DIAG\_16409 |
| Description | The Diagnostics module shall include functionality for generating alerts and logging events when runtime firmware integrity checks detect anomalies, providing diagnostic information for troubleshooting. |
| Rationale | Generating alerts and logging events for runtime firmware integrity issues assists in timely identification and resolution of potential problems. |
| Use Case | When a runtime integrity check detects anomalies, the Diagnostics module generates alerts and logs relevant information for analysis. |
| Priority | Medium |
| Dependencies | Integration with version compatibility error handling mechanisms. |

|  |  |
| --- | --- |
| ID | SRS\_BSW\_16410 |
| Description | The system's basic software shall include support for a UART bootloader mechanism to enable firmware updates |
| Rationale | A UART bootloader is a fundamental component for updating the firmware in embedded systems. |
| Use Case | Technicians can initiate firmware updates through UART, ensuring the system stays up-to-date. |
| Priority | High |
| Dependencies | The system must have an operating system with real-time capabilities. |

|  |  |
| --- | --- |
| ID | SRS\_OS\_16411 |
| Description | The operating system shall provide real-time support for the bootloader mechanism to ensure timely response during firmware updates. |
| Rationale | A real-time operating system is essential for managing firmware updates effectively. |
| Use Case | During firmware updates, the operating system must prioritize bootloader tasks to avoid delays. |
| Priority | High |
| Dependencies | The system must have an operating system with real-time capabilities. |

|  |  |
| --- | --- |
| ID | SRS\_COMM\_16412 |
| Description | The system shall establish UART communication channels to facilitate firmware updates. |
| Rationale | UART communication is a reliable method for transferring firmware updates. |
| Use Case | Firmware update files can be transmitted via UART from external sources to the system. |
| Priority | Medium |
| Dependencies | The system must have UART communication capability. |

|  |  |
| --- | --- |
| ID | SRS\_DIAG\_16413 |
| Description | The system shall implement diagnostic mechanisms to handle errors and interruptions during firmware updates |
| Rationale | Robust error handling ensures the integrity of firmware updates. |
| Use Case | If an error occurs during a firmware update, the system must handle it gracefully and provide diagnostic information. |
| Priority | High |
| Dependencies | The system must have diagnostic capabilities. |

|  |  |
| --- | --- |
| ID | SRS\_ECU\_16414 |
| Description | Each Electronic Control Unit (ECU) shall be integrated with the UART bootloader mechanism to allow individual ECU firmware updates. |
| Rationale | ECU-specific firmware updates are essential for system maintenance. |
| Use Case | Individual ECUs can receive firmware updates through the UART bootloader. |
| Priority | High |
| Dependencies | Each ECU must be compatible with the UART bootloader mechanism. |

|  |  |
| --- | --- |
| ID | SRS\_COMM\_16415 |
| Description | The flashing process for the system shall adhere to a defined protocol to ensure proper and reliable data transfer. |
| Rationale | A standardized protocol is necessary to guarantee consistency and data integrity during the flashing process. |
| Use Case | During firmware updates, the system follows the defined protocol to receive and verify data integrity during transfer. |
| Priority | High |
| Dependencies | The flashing process depends on the availability and implementation of the defined protocol for data transfer. |

|  |  |
| --- | --- |
| ID | SRS\_SEC\_16416 |
| Description | The system shall implement a secure bootloader mechanism to prevent unauthorized or malicious firmware updates. |
| Rationale | Security is paramount to protect the system from potential threats and ensure the integrity of firmware updates. |
| Use Case | The secure bootloader mechanism verifies the authenticity and integrity of firmware updates before installation, preventing unauthorized updates. |
| Priority | High |
| Dependencies | The secure bootloader mechanism relies on security measures to prevent unauthorized updates. |

## 6.4.2. Non-Functional\_SRS

|  |  |
| --- | --- |
| ID | SRS\_PER\_16420 |
| Description | The flashing process shall perform validation checks with minimal impact on system performance to ensure efficient firmware installation. |
| Rationale | Efficient validation checks maintain system responsiveness during firmware flashing. |
| Use Case | The flashing process executes validation checks while minimizing delays in the firmware installation process. |
| Priority | Medium |
| Dependencies | Optimization of validation check algorithms and execution. |

|  |  |
| --- | --- |
| ID | SRS\_SEC\_16421 |
| Description | The system shall ensure that version compatibility checks are performed securely to prevent tampering or bypassing of the checks. |
| Rationale | Secure version compatibility checks are necessary to maintain system integrity and prevent malicious installations. |
| Use Case | Version compatibility checks are performed securely, and mechanisms are in place to prevent unauthorized interference. |
| Priority | Medium |
| Dependencies | Security mechanisms for version compatibility checks. |

|  |  |
| --- | --- |
| ID | SRS\_DIAG\_16422 |
| Description | The system shall perform runtime firmware integrity checks with minimal impact on system performance to ensure efficient system operation. |
| Rationale | Efficient runtime checks maintain system responsiveness while ensuring firmware integrity. |
| Use Case | The system executes runtime checks, such as checksum or CRC checks, with minimal performance impact. |
| Priority | Medium |
| Dependencies | Optimization of runtime check algorithms and execution. |

|  |  |
| --- | --- |
| ID | SRS\_SEC\_16423 |
| Description | The flashing process shall include security measures to ensure data integrity during transfer, preventing unauthorized or malicious alterations. |
| Rationale | Data integrity and security are critical during the flashing process to protect the system from potential vulnerabilities. |
| Use Case | The system verifies data integrity and authenticity during firmware updates to prevent tampering. |
| Priority | High |
| Dependencies | The flashing process relies on security measures to maintain data integrity. |

## 7. Error Handling and Logging

**7.1 Error Messages**

**7.1.1. Functional\_SRS**

|  |  |
| --- | --- |
| ID | SRS\_DIAG\_17101 |
| Description | The Diagnostics module shall include functionality for generating alerts and logging events when runtime firmware integrity checks detect anomalies, providing diagnostic information for troubleshooting. |
| Rationale | Generating alerts and logging events for runtime firmware integrity issues assists in timely identification and resolution of potential problems. |
| Use Case | When a runtime integrity check detects anomalies, the Diagnostics module generates alerts and logs relevant information for analysis. |
| Priority | Medium |
| Dependencies | Integration with version compatibility error handling mechanisms. |

### 7.2 Fault Logging

**7.2.1. Functional\_SRS**

|  |  |
| --- | --- |
| ID | SRS\_ DIAG \_17201 |
| Description | The system shall record failures and log them during runtime. |
| Rationale | The recording of failures and logging them during runtime is crucial for effective system monitoring and troubleshooting. By capturing and logging failures, the system can provide valuable information for diagnosing issues. |
| Use Case | The system is operational.  Failure detection mechanisms are in place.  The failure information is successfully logged for further analysis and diagnostics and the system continues normal operation. |
| Priority | High |
| Dependencies | The system must have appropriate failure detection mechanisms in place to identify and categorize failures accurately.  A reliable log storage mechanism needs to be available to store the logged failure information. |

|  |  |
| --- | --- |
| ID | SRS\_ DIAG \_17202 |
| Description | The logged failures shall include relevant information such as timestamps, error codes, and error descriptions. |
| Rationale | This requirement enhances the diagnostic value of the logs. This information facilitates efficient troubleshooting and root cause analysis. |
| Use Case | The logged failures include accurate and relevant information for diagnostic purposes. |
| Dependencies | The system must have access to the necessary information sources to capture relevant failure details, such as timestamps, error codes, and error descriptions. |
| Priority | High |

|  |  |
| --- | --- |
| ID | SRS\_SEC\_17203 |
| Description | Clearing the logs shall require appropriate authorization and confirmation to prevent accidental data loss. |
| Rationale | This requirement adds an extra layer of security and ensures that only authorized users can perform the log clearing operation. |
| Use Case | Log clearing is performed with appropriate authorization and confirmation, reducing the risk of accidental data loss. |
| Dependencies | The system must have an appropriate user authentication and authorization mechanism in place to verify the user's identity and access rights.  The log clearing functionality should be integrated with the authorization and confirmation process. |
| Priority | High |

|  |  |
| --- | --- |
| ID | SRS\_SEC\_17204 |
| Description | The system shall ensure the security and integrity of the logged data, preventing unauthorized access or tampering. |
| Rationale | This requirement focuses on maintaining the confidentiality, integrity, and availability of the logged failure data. |
| Use Case | The logged failure data remains secure, intact, and available for authorized users. |
| Dependencies | The system should have robust security mechanisms in place, including authentication, encryption, and access controls.  Data integrity checks should be integrated into the logging and retrieval processes.  Redundant storage or backup mechanisms should be implemented to ensure data availability and resilience. |
| Priority | High |

|  |  |
| --- | --- |
| ID | SRS\_BSW\_17205 |
| Description | The logged failures shall be stored in a persistent and reliable manner to ensure availability even during system restarts. |
| Rationale | This requirement emphasizes the importance of maintaining the availability and continuity of the logged failure data. |
| Use Case | The logged failure data is stored persistently and reliably, ensuring its availability and continuity. |
| Dependencies | The system should have access to a reliable and persistent storage medium, such as non-volatile memory or disk storage.  Redundancy measures, such as backup systems or replication mechanisms, should be in place to prevent data loss. |
| Priority | High |

**7.2.2. Non-Functional\_SRS**

|  |  |
| --- | --- |
| ID | SRS\_DIAG\_17206 |
| Description | The system shall provide a simple diagnostics interface in the operation mode to read the logged failures. |
| Rationale | This requirement enhances the system's usability and facilitates efficient troubleshooting and analysis. |
| Use Case | The system is in operation mode.  Failure logs are available.  The operator successfully retrieves and views the logged failure information. |
| Dependencies | The failure logs must be stored in a reliable and accessible manner.  The system should have a user-friendly interface framework in place to support diagnostics and log retrieval. |
| Priority | Medium |

|  |  |
| --- | --- |
| ID | SRS\_SEC\_17207 |
| Description | Only users in Engineering Mode shall have the ability to clear the logs. |
| Rationale | This requirement focuses on maintaining data integrity and preventing unauthorized access. |
| Use Case | The user has appropriate authorization in Engineering Mode.  The logs are successfully cleared, ensuring a clean log storage for future failures. |
| Dependencies | The system must have an Engineering Mode feature implemented with appropriate access controls and authorization mechanisms. |
| Priority | High |

|  |  |
| --- | --- |
| ID | SRS\_DIAG\_17208 |
| Description | The diagnostics interface shall be user-friendly, allowing operators to easily navigate and retrieve log information. |
| Rationale | This requirement improves the efficiency of troubleshooting and analysis, reducing the time required for issue resolution. |
| Use Case | The operator easily navigates and retrieves the desired failure log information. |
| Dependencies | The system should have a user interface framework capable of supporting intuitive navigation and presentation of log information.  The logged failure information should be stored in a structured format that allows for easy retrieval and display. |
| Priority | Medium |

### 7.3 Firmware Updates Log

**7.3.1. Functional\_SRS**

|  |  |
| --- | --- |
| ID | SRS\_DIAG\_17301 |
| Description | The log entry for each firmware update event shall include the version number of the updated firmware. |
| Rationale | This requirement enables easy identification and reference to a specific firmware version associated with each update event and It helps in tracking and managing firmware versions. |
| Use Case | The system extracts the version number from the firmware update package and includes it in the log entry. |
| Priority | High |
| Dependencies | The record of changes made to the firmware over time. |

|  |  |
| --- | --- |
| ID | SRS\_DIAG\_17302 |
| Description | The log entry for each firmware update event shall include the timestamp indicating when the update occurred. |
| Rationale | This requirement helps in troubleshooting, auditing, and analyzing the sequence of firmware updates. |
| Use Case | When a firmware update is performed, the system records the current timestamp and associates it with the log entry. |
| Priority | High |
| Dependencies | The record of changes made to the firmware over time. |

|  |  |
| --- | --- |
| ID | SRS\_DIAG\_17303 |
| Description | The log entry for each firmware update event shall include any relevant diagnostic information associated with the update. |
| Rationale | This requirement provides additional context and aids in understanding the cause of any potential problems. |
| Use Case | During the firmware update process, the system captures and includes diagnostic information, such as error codes or descriptions, in the log entry. |
| Priority | High |
| Dependencies | The record of changes made to the firmware over time. |

**7.3.2. Non-Functional\_SRS**

|  |  |
| --- | --- |
| ID | SRS\_DIAG\_17304 |
| Description | The system shall maintain a log of firmware update events. |
| Rationale | This requirement allows for tracking, auditing, and historical analysis of firmware updates and provides a record of changes made to the firmware over time. |
| Use Case | When a firmware update is initiated, the system creates a log entry to capture the event. |
| Dependencies | None |
| Priority | High |

## 8. Size and Format Constraints

### 8.1 Firmware Size

**8.1.1. Functional\_SRS**

|  |  |
| --- | --- |
| ID | SRS\_BSW\_18101 |
| Description | The system shall enforce a maximum size constraint on the firmware image. |
| Rationale | This requirement prevents issues such as insufficient storage or excessive memory usage. |
| Use Case | When a firmware image is uploaded, the system checks its size against the specified maximum constraint before proceeding with the update. |
| Priority | High |
| Dependencies | None |

|  |  |
| --- | --- |
| ID | SRS\_BSW\_18102 |
| Description | The system shall support specific formats for the firmware image. |
| Rationale | This requirement mitigates the risk of errors or malfunctions caused by using incompatible formats. |
| Use Case | When a firmware image is uploaded, the system verifies that it is in one of the supported formats before proceeding with the update. |
| Priority | High |
| Dependencies | None |

|  |  |
| --- | --- |
| ID | SRS\_MCAL\_18103 |
| Description | The system shall validate the compatibility of the firmware image with the target hardware. |
| Rationale | This requirement helps avoid issues such as hardware incompatibility or failed firmware updates. |
| Use Case | During the firmware update process, the system checks the compatibility of the firmware image with the target hardware before initiating the update. |
| Priority | High |
| Dependencies | None |

|  |  |
| --- | --- |
| ID | SRS\_BSW\_18104 |
| Description | The system shall perform a memory resource check before executing a firmware update. |
| Rationale | This requirement ensures that sufficient memory is available to accommodate the firmware update and prevents potential failures or system instability due to insufficient memory. |
| Use Case | Prior to starting a firmware update, the system checks the available memory resources to ensure they meet the requirements of the update. |
| Priority | High |
| Dependencies | None |

|  |  |
| --- | --- |
| ID | SRS\_ECU\_18105 |
| Description | The system shall define a maximum allowable firmware image size for the embedded system. |
| Rationale | This requirement prevents potential issues related to NVM storage space. It sets a clear constraint for developers and ensures that the firmware remains within manageable limits. |
| Use Case | This requirement applies when configuring the embedded system for firmware development. It provides a basis for determining the firmware image size during the development process. |
| Priority | High |
| Dependencies | None |

|  |  |
| --- | --- |
| ID | SRS\_BSW\_18106 |
| Description | The system shall continuously monitor the available storage space in the NVM. |
| Rationale | This requirement ensure that the system can detect any deviations from the maximum allowable firmware image size and take corrective actions in real-time. |
| Use Case | This requirement is applicable during system operation. It ensures that the system constantly tracks NVM storage space usage. |
| Priority | High |
| Dependencies | None |

**8.1.2. Non-Functional\_SRS**

|  |  |
| --- | --- |
| ID | SRS\_DIAG\_18107 |
| Description | The system shall provide a mechanism for checking the current firmware image size. |
| Rationale | This requirement providing a mechanism for checking the current firmware image size allows system administrators to assess whether the firmware size is within acceptable limits. |
| Use Case | This requirement is useful when system administrators need to verify the firmware size for maintenance, troubleshooting, or compliance checks. |
| Priority | Medium |
| Dependencies | SRS\_BSW\_10076 (To determine the allowable firmware image size) |

|  |  |
| --- | --- |
| ID | SRS\_DIAG\_18108 |
| Description | If the firmware image size exceeds the available storage space in the NVM, the system shall generate a warning or error message. |
| Rationale | Generating a warning or error message informs system administrators and users of potential issues, allowing them to take appropriate actions to avoid system failures. |
| Use Case | This requirement is applicable when the firmware size exceeds the defined limits, alerting system administrators to take corrective measures. |
| Priority | High |
| Dependencies | SRS\_BSW\_10076 (To define the allowable firmware image size)  SRS\_BSW\_10077 (To monitor available storage space) |

|  |  |
| --- | --- |
| ID | SRS\_BSW\_18109 |
| Description | The system shall provide an automated mechanism for reducing the firmware image size if it exceeds the available storage space in the NVM. |
| Rationale | The automated mechanism ensures that the system can self-correct by reducing the firmware size if it becomes too large for the available NVM storage. |
| Use Case | This requirement comes into play when the firmware size exceeds available NVM storage space, and the system needs to automatically optimize the firmware. |
| Priority | High |
| Dependencies | SRS\_BSW\_10079 (To define the allowable firmware image size) |

|  |  |
| --- | --- |
| ID | SRS\_DIAG\_18110 |
| Description | The system shall log events related to firmware image size and NVM storage space usage. |
| Rationale | Logging events allows for system performance analysis, troubleshooting, and auditing to ensure compliance with this requirement. |
| Use Case | This requirement is relevant during system operation and maintenance, allowing administrators to review logs for performance analysis or troubleshooting. |
| Priority | Medium |
| Dependencies | SRS\_BSW\_10077 (To obtain storage space usage data) |

|  |  |
| --- | --- |
| ID | SRS\_DIAG\_18111 |
| Description | The system shall provide documentation specifying the maximum allowable firmware image size and instructions for monitoring and managing firmware size. |
| Rationale | Documentation ensures that system administrators have the necessary information to configure and manage firmware image size effectively. |
| Use Case | This requirement is relevant when configuring the system and during maintenance when administrators need guidance on firmware size management. |
| Priority | Medium |
| Dependencies | None |

## 9. Operation Modes

**9.0.1. Functional\_SR**

|  |  |
| --- | --- |
| ID | SRS\_BSW\_19001 |
| Description | The system shall implement an Operation Mode for standard system functionality. |
| Rationale | Operation Mode provides regular system functionality, ensuring the system performs its intended tasks. |
| Use Case | * In an industrial automation system, Operation Mode allows continuous production and monitoring. |
| Priority | High |
| Dependencies | Depends on the design and implementation of Operation Mode. |

|  |  |
| --- | --- |
| ID | SRS\_BSW\_19002 |
| Description | The system shall implement a Protected Engineering Mode for restricted access to configuration and diagnostic functions. |
| Rationale | Protected Engineering Mode ensures that critical system settings and diagnostics are available for authorized users only, enhancing security and system maintenance. |
| Use Case | * In a medical device, engineers use Protected Engineering Mode to perform diagnostics and calibration securely. |
| Priority | High |
| Dependencies | Depends on the design and implementation of Protected Engineering Mode. |

|  |  |
| --- | --- |
| ID | SRS\_BSW\_19003 |
| Description | The system shall provide a mechanism for users to switch between Operation Mode and Protected Engineering Mode. |
| Rationale | Mode switching allows authorized users to transition between standard operation and configuration or diagnostic tasks as needed. |
| Use Case | * In an automotive system, an authorized technician can switch to Protected Engineering Mode to calibrate sensors and then return to Operation Mode. |
| Priority | Medium |
| Dependencies | Depends on the design and implementation of the mode-switching mechanism. |

**9.0.2. Non-Functional\_SR**

|  |  |
| --- | --- |
| ID | SRS\_BSW\_19004 |
| Description | Access to Protected Engineering Mode shall be controlled through authentication mechanisms. |
| Rationale | Access control ensures that only authorized personnel can enter Protected Engineering Mode, maintaining security. |
| Use Case | * In a manufacturing facility, access control ensures that only trained engineers can access the engineering mode. |
| Priority | High |
| Dependencies | Depends on the integration of authentication and access control mechanisms. |

|  |  |
| --- | --- |
| ID | SRS\_BSW\_19005 |
| Description | Both Operation Mode and Protected Engineering Mode shall be protected by appropriate security measures to prevent unauthorized access. |
| Rationale | Security measures safeguard both modes from unauthorized access, protecting the system's integrity. |
| Use Case | * In a financial application, security measures ensure that financial data remains confidential in both modes. |
| Priority | High |
| Dependencies | Depends on the implementation of security measures for both modes. |

### 9.1 Operation Mode

**9.1.1. Functional\_SR**

|  |  |
| --- | --- |
| ID | SRS\_BSW\_19101 |
| Description | The system shall be able to read sensor values from connected sensors. |
| Rationale | Reading sensor data is essential for the system to gather information about its environment and make informed decisions or provide feedback. |
| Use Case | In a home automation system, when the temperature sensor detects a temperature drop below a set threshold, the system needs to read this data to trigger the heating system. |
| Priority | High |
| Dependencies | This requirement depends on the existence and proper functioning of the sensors connected to the system. |

|  |  |
| --- | --- |
| ID | SRS\_AP\_19102 |
| Description | The system shall display sensor values on the Human-Machine Interface (HMI) in a user-friendly format. |
| Rationale | Providing users with real-time sensor data on the HMI is crucial for monitoring and making informed decisions. |
| Use Case | In a weather monitoring system, the HMI should display current temperature, humidity, and wind speed to help users plan their activities. |
| Priority | High |
| Dependencies | This requirement depends on the availability and proper functioning of the Human-Machine Interface (HMI) components, including display screens and user interface software. |

|  |  |
| --- | --- |
| ID | SRS\_AP\_19103 |
| Description | The system shall control actuators based on sensor data and user inputs. |
| Rationale | Controlling actuators is necessary to respond to sensor data and user commands, enabling the system to take appropriate actions. |
| Use Case | In an industrial automation system, when a pressure sensor detects a sudden increase, the system should immediately shut down the corresponding machine to prevent damage. |
| Priority | High |
| Dependencies | This requirement depends on the existence and proper functioning of the actuators connected to the system. |

|  |  |
| --- | --- |
| ID | SRS\_AP\_19104 |
| Description | The system shall have an "Operation Mode" which encompasses normal functionality. |
| Rationale | Designating an "Operation Mode" ensures that the system is ready to perform its core functions as intended. |
| Use Case | In an autonomous vehicle, the "Operation Mode" indicates that the vehicle is in normal driving mode, where it is actively sensing the environment and following its programmed route. |
| Priority | Medium |
| Dependencies | This requirement depends on the overall system architecture and software design, as it defines the system's operating state. |

|  |  |
| --- | --- |
| ID | SRS\_AP\_19105 |
| Description | The HMI shall allow users to interact with the system, including configuring sensor thresholds and controlling actuators. |
| Rationale | User interaction is necessary for customization and control over the system's behavior. |
| Use Case | In a smart home system, users should be able to set temperature thresholds and manually control lighting through the HMI. |
| Priority | Medium |
| Dependencies | This requirement depends on the design and implementation of the HMI components and the user interface software. |

**9.1.2. Functional\_SR**

|  |  |
| --- | --- |
| ID | SRS\_BSW\_19106 |
| Description | The system shall refresh sensor data at a rate of at least X readings per second. |
| Rationale | Defining a minimum data refresh rate ensures that the system provides timely and accurate information to users. |
| Use Case | A medical monitoring system, sensor data related to a patient's vital signs must be refreshed at a high rate to enable real-time monitoring by medical staff. |
| Dependencies | This requirement depends on the capabilities of the sensor hardware and the system's processing capacity. |
| Priority | Medium |

|  |  |
| --- | --- |
| ID | SRS\_BSW\_19107 |
| Description | The system shall ensure that actuator responses to control commands occur within Y milliseconds. |
| Rationale | Timely actuator responses are critical to maintaining system performance and safety. |
| Use Case | In a robotics assembly line, the robot arm must respond quickly to control commands to precisely manipulate objects. |
| Dependencies | This requirement depends on the performance capabilities of the actuators, the control algorithms, and the communication infrastructure. |
| Priority | Medium |

|  |  |
| --- | --- |
| ID | SRS\_BSW\_19108 |
| Description | The system shall incorporate safety overrides to halt actuator control in case of emergency or abnormal sensor readings. |
| Rationale | Safety overrides are critical for preventing accidents and damage to equipment. |
| Use Case | In an industrial automation system, if a smoke detector senses a fire, the system must immediately override any ongoing processes and initiate fire safety protocols. |
| Dependencies | This requirement depends on the presence of safety mechanisms, sensors, and the system's ability to monitor sensor data. |
| Priority | High |

### 9.2 Protected Engineering Mode

**9.2.1. Functional\_SR**

|  |  |
| --- | --- |
| ID | SRS\_BSW\_19201 |
| Description | The system shall implement a Protected Engineering Mode to restrict access to configuration parameters. |
| Rationale | This requirement ensures that unauthorized users or processes cannot modify critical configuration settings, enhancing system stability and safety. |
| Use Case | In an automotive system, engineers need to adjust brake light brightness for testing and calibration, and this requirement ensures they can do so securely. |
| Priority | High |
| Dependencies | Depends on the design and implementation of the Protected Engineering Mode. |

|  |  |
| --- | --- |
| ID | SRS\_BSW\_19202 |
| Description | The system shall include access control mechanisms to grant access to Protected Engineering Mode. |
| Rationale | Access control ensures that only authorized personnel can enter Engineering Mode, preventing unauthorized changes. |
| Use Case | In a medical device, trained technicians should have access to Engineering Mode to adjust safety-critical parameters. |
| Priority | High |
| Dependencies | Depends on the implementation of access control mechanisms. |

|  |  |
| --- | --- |
| ID | SRS\_BSW\_19203 |
| Description | To enter Protected Engineering Mode, the system shall require user authentication to verify the identity of users. |
| Rationale | User authentication ensures that only authorized individuals can make changes in Engineering Mode, enhancing system security. |
| Use Case | In an industrial control system, engineers must authenticate themselves before adjusting control parameters. |
| Priority | High |
| Dependencies | Depends on the integration of user authentication mechanisms. |

|  |  |
| --- | --- |
| ID | SRS\_BSW\_19204 |
| Description | Within Protected Engineering Mode, the system shall provide the ability to lock specific configuration parameters. |
| Rationale | Parameter locking prevents accidental or unauthorized changes to critical settings, improving system reliability. |
| Use Case | In an aircraft avionics system, engineers should be able to lock vital flight parameter settings during testing. |
| Priority | Medium |
| Dependencies | Depends on the development of a parameter locking feature within Engineering Mode. |

|  |  |
| --- | --- |
| ID | SRS\_BSW\_19205 |
| Description | The system shall maintain an audit trail of all changes made to configuration parameters within Engineering Mode, including user, date, and time of modification. |
| Rationale | An audit trail provides transparency and accountability for parameter changes, facilitating troubleshooting and compliance. |
| Use Case | * In a financial software application, changes to interest rate calculation parameters should be logged for regulatory purposes. |
| Priority | Medium |
| Dependencies | Depends on the implementation of an auditing system. |

|  |  |
| --- | --- |
| ID | SRS\_BSW\_19206 |
| Description | The system shall generate notifications or alerts when users enter Protected Engineering Mode or make parameter changes within Engineering Mode. |
| Rationale | Access notifications help monitor and detect unauthorized or unusual activities in Engineering Mode. |
| Use Case | In a security system, administrators should receive immediate alerts when someone enters Engineering Mode. |
| Priority | Low |
| Dependencies | Depends on the integration of a notification system. |

|  |  |
| --- | --- |
| ID | SRS\_BSW\_19207 |
| Description | The system shall store configuration parameters in two separate and distinct locations. |
| Rationale | Storing configuration parameters in two locations adds redundancy and safeguards against data loss and manipulation. |
| Use Case | In a financial application, critical interest rate settings can be stored in two locations to prevent data corruption or manipulation. |
| Priority | High |
| Dependencies | Depends on the design and implementation of dual storage locations. |

|  |  |
| --- | --- |
| ID | SRS\_BSW\_19208 |
| Description | The system shall regularly perform consistency checks between the two storage locations to ensure data integrity. |
| Rationale | Consistency checks help identify and correct discrepancies between the dual storage locations, ensuring data reliability. |
| Use Case | In an industrial control system, parameters related to machinery operation should be cross-checked between the two locations to detect any inconsistencies. |
| Priority | High |
| Dependencies | Depends on the development of a consistency-checking mechanism. |

|  |  |
| --- | --- |
| ID | SRS\_BSW\_19209 |
| Description | The system shall include a failover mechanism that allows it to switch to the secondary storage location if the primary location becomes unavailable. |
| Rationale | The failover mechanism ensures continuous system operation even if one storage location experiences a failure or becomes inaccessible. |
| Use Case | In a telecommunications system, critical network configuration parameters should be accessible from a secondary location if the primary storage location encounters issues. |
| Priority | Medium |
| Dependencies | Depends on the integration of a failover mechanism. |

|  |  |
| --- | --- |
| ID | SRS\_BSW\_19210 |
| Description | The system shall implement data synchronization mechanisms to keep the two storage locations up-to-date. |
| Rationale | Data synchronization ensures that changes made to configuration parameters in one location are mirrored in the other, maintaining consistency. |
| Use Case | In a cloud-based service, user profile settings should be synchronized between multiple data centers to ensure uniform user experiences. |
| Priority | Medium |
| Dependencies | Depends on the development of data synchronization processes. |

|  |  |
| --- | --- |
| ID | SRS\_BSW\_19211 |
| Description | Both storage locations shall be protected by appropriate security measures to prevent unauthorized access and manipulation. |
| Rationale | Security measures safeguard the integrity of configuration parameters stored in both locations. |
| Use Case | In a healthcare system, patient data privacy settings should be protected in dual storage locations to comply with regulations. |
| Priority | High |
| Dependencies | Depends on the implementation of security measures. |

|  |  |
| --- | --- |
| ID | SRS\_BSW\_19212 |
| Description | The system shall log any changes made to configuration parameters in both storage locations, including user, date, and time information. |
| Rationale | Logging changes provides an audit trail for accountability and troubleshooting. |
| Use Case | In a manufacturing system, changes to production parameters should be logged in both storage locations for quality control and compliance. |
| Priority | Low |
| Dependencies | Depends on the implementation of logging mechanisms. |

|  |  |
| --- | --- |
| ID | SRS\_BSW\_19213 |
| Description | The system shall offer a Protected Engineering Mode that provides access to protected functions. |
| Rationale | This requirement establishes the existence of the Engineering Mode, which is a fundamental feature for configuring and performing critical functions. |
| Use Case | In an automotive control system, engineers need access to Engineering Mode to adjust brake indicator brightness and calibrate non-volatile memory (NVM) data. |
| Priority | High |
| Dependencies | Depends on the design and implementation of Engineering Mode. |

|  |  |
| --- | --- |
| ID | SRS\_BSW\_19214 |
| Description | Access to Protected Engineering Mode shall require a seed key-based authentication mechanism. |
| Rationale | Seed key-based authentication ensures that only authorized individuals can access protected functions, enhancing system security. |
| Use Case | In a medical device, trained technicians must provide the correct seed key to enter Engineering Mode for critical calibration tasks. |
| Priority | High |
| Dependencies | Depends on the integration of seed key-based authentication. |

|  |  |
| --- | --- |
| ID | SRS\_BSW\_19215 |
| Description | The system shall generate and securely manage seed keys for Protected Engineering Mode access. |
| Rationale | Proper seed key generation and management are essential for maintaining the security and integrity of the authentication process. |
| Use Case | In an industrial control system, the system must generate unique seed keys for each authorized engineer for secure Engineering Mode access. |
| Priority | High |
| Dependencies | Depends on the development of seed key generation and management mechanisms. |

|  |  |
| --- | --- |
| ID | SRS\_BSW\_19216 |
| Description | The system shall allow authorized personnel to exchange seed keys securely to facilitate access to Protected Engineering Mode. |
| Rationale | Secure seed key exchange ensures that authorized users can obtain the required keys for access. |
| Use Case | In a telecommunications system, network administrators can securely exchange seed keys with colleagues to access Engineering Mode for critical network configuration. |
| Priority | Medium |
| Dependencies | Depends on the development of secure key exchange mechanisms. |

|  |  |
| --- | --- |
| ID | SRS\_BSW\_19217 |
| Description | The system shall integrate a diagnostic service within Protected Engineering Mode to facilitate configuration changes. |
| Rationale | Integration of a diagnostic service is essential for performing protected functions securely within Engineering Mode. |
| Use Case | In a manufacturing control system, engineers use the diagnostic service within Engineering Mode to fine-tune machine parameters. |
| Priority | High |
| Dependencies | Depends on the design and implementation of the diagnostic service. |

|  |  |
| --- | --- |
| ID | SRS\_BSW\_19218 |
| Description | Within Protected Engineering Mode, the system shall allow authorized users to perform non-volatile memory (NVM) calibration data changes. |
| Rationale | This requirement specifies a protected function within Engineering Mode, allowing authorized personnel to make critical NVM adjustments. |
| Use Case | In a robotics system, engineers can access Engineering Mode to perform NVM calibration data changes for precise robot arm control. |
| Priority | High |
| Dependencies | Depends on the implementation of NVM calibration data change mechanisms within Engineering Mode. |

|  |  |
| --- | --- |
| ID | SRS\_BSW\_19219 |
| Description | The system shall provide a Protected Engineering Mode for authorized personnel to access. |
| Rationale | This requirement establishes the existence of Engineering Mode, which is a fundamental functional aspect of the system for configuring and maintaining protected functions. |
| Use Case | * In an automotive control system, engineers and technicians need access to Engineering Mode to perform calibration and configuration tasks securely. |
| Priority | High |
| Dependencies | Depends on the design and implementation of Engineering Mode. |

|  |  |
| --- | --- |
| ID | SRS\_BSW\_19220 |
| Description | Access to Protected Engineering Mode shall require a robust authentication mechanism. |
| Rationale | Robust authentication mechanisms ensure that only authorized individuals can access Engineering Mode, enhancing system security. |
| Use Case | In a healthcare device, healthcare professionals must authenticate themselves securely to access Engineering Mode for equipment calibration and setup. |
| Priority | High |
| Dependencies | Depends on the implementation of authentication mechanisms. |

|  |  |
| --- | --- |
| ID | SRS\_BSW\_19221 |
| Description | The system shall enforce role-based access control to Engineering Mode. |
| Rationale | Role-based access control ensures that users are granted appropriate privileges within Engineering Mode based on their roles, responsibilities, and permissions. |
| Use Case | In a manufacturing facility, different personnel, such as operators and engineers, should have different levels of access to Engineering Mode. |
| Priority | Medium |
| Dependencies | Depends on the development of role-based access control features. |

|  |  |
| --- | --- |
| ID | SRS\_BSW\_19222 |
| Description | The authentication mechanism for Engineering Mode shall include password-based authentication. |
| Rationale | Password-based authentication adds an additional layer of security by requiring a secure password for access. |
| Use Case | In a financial application, administrators must enter a secure password to access Engineering Mode for configuration changes. |
| Priority | Medium |
| Dependencies | Depends on the integration of password-based authentication mechanisms. |

**9.2.2. Non-Functional\_SRS**

|  |  |
| --- | --- |
| ID | SRS\_BSW\_19223 |
| Description | The system shall implement appropriate security measures to safeguard access to Engineering Mode. |
| Rationale | Security measures, including encryption and intrusion detection, ensure the integrity and confidentiality of Engineering Mode access. |
| Use Case | In an industrial control system, security measures are essential to protect against unauthorized access and data breaches in Engineering Mode. |
| Priority | High |
| Dependencies | Depends on the implementation of security measures. |

|  |  |
| --- | --- |
| ID | SRS\_BSW\_19224 |
| Description | The system shall maintain an audit trail of access attempts, including successful and unsuccessful attempts, user information, and timestamps. |
| Rationale | Logging access attempts provides transparency, accountability, and aids in security monitoring and forensic analysis. |
| Use Case | In a critical infrastructure control system, all access attempts to Engineering Mode should be logged for regulatory compliance and security monitoring. |
| Priority | Low |
| Dependencies | Depends on the implementation of logging mechanisms. |

## 10. Requirements Validation and Testing

### 10.1 Validation Strategies

Validation strategies for the DriveSync system requirements are outlined, ensuring that each requirement is rigorously tested and validated throughout the development process.

### 10.2 Integration and Validation

**10.2.1. Functional\_SRS**

|  |  |
| --- | --- |
| ID | SRS\_BSW&AP&RTE\_11201 |
| Description | The system shall define and follow a structured integration strategy to ensure the seamless combination of system components and modules. |
| Rationale | A clear integration strategy ensures that all system parts work harmoniously together, reducing the risk of integration issues during development and deployment. |
| Use Case | * In a complex software application, the integration strategy guides how different modules are combined to create the final product. |
| Priority | High |
| Dependencies | Depends on the establishment and documentation of the integration strategy. |

|  |  |
| --- | --- |
| ID | SRS\_BSW&AP\_11202 |
| Description | The system shall define and adhere to a validation strategy to systematically verify that the system meets its requirements and specifications. |
| Rationale | A validation strategy ensures that the system is thoroughly tested and validated to ensure it meets its intended purpose and requirements. |
| Use Case | * In a healthcare software system, the validation strategy outlines how medical device functionality is rigorously tested and validated before clinical use. |
| Priority | High |
| Dependencies | Depends on the establishment and documentation of the validation strategy. |

|  |  |
| --- | --- |
| ID | SRS\_AP\_11205 |
| Description | The system shall validate each requirement during the integration process to ensure that it has been correctly implemented and meets its specifications. |
| Rationale | Requirement validation during integration is essential to confirm that the system components work as intended and collectively fulfill the requirements. |
| Use Case | * In a software application, this requirement ensures that each feature or functionality specified in the requirements is thoroughly tested during integration. |
| Priority | High |
| Dependencies | Depends on the integration process and the availability of test cases for each requirement. |

|  |  |
| --- | --- |
| ID | SRS\_BSW&RTE\_11206 |
| Description | The system shall implement error handling mechanisms to capture and manage errors and exceptions that occur during integration and runtime. |
| Rationale | Error handling mechanisms are crucial for gracefully managing unexpected situations, preventing system crashes, and facilitating debugging and troubleshooting. |
| Use Case | * In a web application, error handling mechanisms help handle issues such as database connection failures or server timeouts. |
| Priority | High |
| Dependencies | Depends on the design and implementation of error handling components. |

|  |  |
| --- | --- |
| ID | SRS\_BSW\_11207 |
| Description | The system shall incorporate fault detection mechanisms to identify and respond to faults or anomalies in the system's behavior during integration and operation. |
| Rationale | Fault detection mechanisms enhance system reliability and availability by promptly identifying and addressing issues that may lead to system failures. |
| Use Case | * In an autonomous vehicle system, fault detection mechanisms can detect sensor failures or abnormal vehicle behavior. |
| Priority | High |
| Dependencies | Depends on the design and implementation of fault detection components. |

**10.2.2. Functional\_SRS**

|  |  |
| --- | --- |
| ID | SRS\_BSW\_11203 |
| Description | The system shall adhere to a predefined static architecture that defines the structure of the system components and their relationships. |
| Rationale | A predefined static architecture provides a clear blueprint for system design, which helps in maintaining consistency and scalability. |
| Use Case | * In an e-commerce platform, a predefined static architecture ensures consistent layout and structure across all pages and components. |
| Priority | High |
| Dependencies | Depends on the establishment and documentation of the static architecture. |

|  |  |
| --- | --- |
| ID | SRS\_AP\_11204 |
| Description | The system shall adhere to a predefined dynamic architecture that defines the behavior and interactions of system components during runtime. |
| Rationale | A predefined dynamic architecture ensures that the system functions as intended during execution and handles interactions correctly. |
| Use Case | * In a real-time financial trading system, a predefined dynamic architecture ensures proper order execution and transaction handling. |
| Priority | High |
| Dependencies | Depends on the establishment and documentation of the dynamic architecture. |

|  |  |
| --- | --- |
| ID | SRS\_RTE\_11208 |
| Description | Error handling mechanisms shall respond to errors and exceptions in a timely manner to minimize system downtime. |
| Rationale | Timely error handling ensures that system disruptions are brief and do not unduly affect system availability and user experience. |
| Use Case | * In an e-commerce platform, timely error handling prevents checkout failures and lost sales due to temporary issues. |
| Priority | Medium |
| Dependencies | Depends on the efficiency of the error handling mechanisms and the complexity of error resolution. |

|  |  |
| --- | --- |
| ID | SRS\_BSW\_11209 |
| Description | Fault detection mechanisms shall be highly reliable in accurately identifying faults and minimizing false positives. |
| Rationale | Reliable fault detection ensures that real issues are addressed promptly while minimizing unnecessary alarms and disruptions. |
| Use Case | * In a power distribution system, reliable fault detection prevents unnecessary shutdowns due to false alarms while promptly addressing actual faults. |
| Priority | Medium |
| Dependencies | Depends on the accuracy and effectiveness of the fault detection algorithms and sensors. |

**11. Appendices**

**11.1 Glossary of Terms**

This section contains a comprehensive glossary of terms used throughout the SRS document, ensuring a shared understanding of key terminology.

**11.2 Acronyms and Abbreviations**

A list of all acronyms and abbreviations used in the document is provided for quick reference.

**11.3 Document Revision History**

This section maintains a record of document revisions, detailing version numbers, dates of modification, and a brief description of changes made, ensuring traceability and transparency in document management.