Software Requirements Specification for Configuration Module of Unit Engine Data Acquisition Unit of Airbus Helicopters Generic Vehicle Monitoring System (GVMS)

Document Number: H698-002-001-CMU

Version No: 2.4

|  |
| --- |
| Proprietary Notice  This document and the information disclosed herein are proprietary data of Howell Instruments, Inc. Neither this document nor the information contained herein shall be reproduced, used, or disclosed to others without the written authorization of Howell Instruments, Inc. |

|  |  |
| --- | --- |
| Written By: | Prajwal R (Software Engineer, ALTEN Global Technologies Private Limited) |
| Reviewed By: | Pallavi T (Software Engineer, ALTEN Global Technologies Private Limited)    Pavitra D (Quality Engineer – Quality Assurance, ALTEN Global Technologies Private Limited) |
| Approved By: | Shyamala B ( Senior Director - Quality Assurance and Management, ALTEN Global Technologies Private Limited)  cid:image001.jpg@01D61E3F.6C97D0F0 |
| Released By: | Madhu Chandra Srinivas (Associate Director- Software, ALTEN Global Technologies Private Limited)  C:\Users\afreen.p.SWSYS\Pictures\SIGNATURE.PNG |

A logo with a triangle and a triangle in a red yellow and blue rectangle

Description automatically generated

**ALTEN Global Technologies Private Limited**

**Table of Contents**

[1 Amendement Record 5](#_Toc171599824)

[2 Overview 8](#_Toc171599825)

[3 Objectives 12](#_Toc171599826)

[4 Acronyms and Definitions 12](#_Toc171599827)

[5 Scope 14](#_Toc171599828)

[6 References 14](#_Toc171599829)

[7 Assumptions 14](#_Toc171599830)

[8 Outstanding Issues 14](#_Toc171599831)

[9 Document Control 15](#_Toc171599832)

[10 Approvals 15](#_Toc171599833)

[11 Distribution 15](#_Toc171599834)

[12 Traceability 15](#_Toc171599835)

[13 Responsibilities 15](#_Toc171599836)

[14 Input 16](#_Toc171599837)

[15 Output 16](#_Toc171599838)

[16 Software High Level Requirements 16](#_Toc171599839)

[16.1 Initialization 16](#_Toc171599840)

[16.1.1 Initialization - Components 16](#_Toc171599841)

[16.1.1.1 Processor 17](#_Toc171599842)

[16.1.1.2 Watchdog timer 17](#_Toc171599843)

[16.1.1.3 System Timer 18](#_Toc171599844)

[16.1.1.4 CAN peripheral 1 18](#_Toc171599845)

[16.1.1.5 Nested Vector Interrupt Controller 18](#_Toc171599846)

[16.1.1.6 FSMC 19](#_Toc171599847)

[16.1.1.7 I2C Initialisation 19](#_Toc171599848)

[16.2 Modes of Operation 19](#_Toc171599849)

[16.2.1 Operational Modes 19](#_Toc171599850)

[16.2.2 PBIT mode 20](#_Toc171599851)

[16.2.3 Normal Mode 20](#_Toc171599852)

[16.2.4 Error Mode 20](#_Toc171599853)

[16.3 A825 Communication 21](#_Toc171599854)

[16.3.1 Logical Communication Channel 21](#_Toc171599855)

[16.3.2 Invalid A825 message 22](#_Toc171599856)

[16.3.3 NSC channel 22](#_Toc171599857)

[16.3.3.1 NSC message format 22](#_Toc171599858)

[16.3.3.2 NSC messages 23](#_Toc171599859)

[16.3.3.3 NSC message - RCI request 26](#_Toc171599860)

[16.3.3.4 NSC message - Configuration Data request 26](#_Toc171599861)

[16.3.4 NOC channel 27](#_Toc171599862)

[16.3.4.1 NOC message format 27](#_Toc171599863)

[16.3.4.2 NOC messages 28](#_Toc171599864)

[16.4 Aircraft Configuration Data 30](#_Toc171599865)

[16.4.1 Aircraft configuration data 31](#_Toc171599866)

[16.5 NVM Data Format 39](#_Toc171599867)

[16.6 BIT 39](#_Toc171599868)

[16.6.1 PBIT 39](#_Toc171599869)

[16.6.2 CBIT 39](#_Toc171599870)

[16.6.3 RAM check 40](#_Toc171599871)

[16.6.4 CPU check 40](#_Toc171599872)

[16.6.5 BIT error 41](#_Toc171599873)

[16.6.6 CRC check 41](#_Toc171599874)

[16.6.7 NVM read/write test 42](#_Toc171599875)

[16.6.8 InitNvram 42](#_Toc171599876)

[16.6.9 CMU switches to DLU 42](#_Toc171599877)

[16.7 Kernel and Scheduler 43](#_Toc171599878)

[16.7.1 Kernel/Scheduler - Initializations 43](#_Toc171599879)

[16.7.2 Scheduler - Algorithm 43](#_Toc171599880)

[16.7.3 Tasks 43](#_Toc171599881)

[16.7.4 Semaphores 44](#_Toc171599882)

[16.7.5 Task activation by semaphore 44](#_Toc171599883)

[16.7.6 Task structure 45](#_Toc171599884)

[16.8 Software Part Number 45](#_Toc171599885)

[16.8.1 Software Part Number – Application Software 45](#_Toc171599886)

[16.9 Resource Utilization 45](#_Toc171599887)

[16.9.1 Memory Usage 46](#_Toc171599888)

[16.9.2 Throughput Usage 46](#_Toc171599889)

LIST OF TABLES

[Table 1: Amendment Record 5](#_Toc171599802)

[Table 2: List of Acronyms 12](#_Toc171599803)

[Table 3: References 14](#_Toc171599804)

[Table 4: Status LED Blink Rate 21](#_Toc171599805)

[Table 5: Service Function Codes 23](#_Toc171599806)

[Table 6: Document Object Codes - NOC 28](#_Toc171599807)

[Table 7: Aircraft Configuration Data 31](#_Toc171599808)

[Table 8: Audio Configuration – Aircraft Parameters 32](#_Toc171599809)

[Table 9: Engine Data Configuration – Aircraft Parameters 33](#_Toc171599810)

[Table 10: Air Data Configuration – Aircraft Parameters 34](#_Toc171599811)

[Table 11: ARINC429 Input Data Configuration – Aircraft Parameters 34](#_Toc171599812)

[Table 12: ARINC429 Output Binary Data Configuration – Aircraft Parameters 35](#_Toc171599813)

[Table 13: ARINC429 Output Discrete Data Configuration – Aircraft Parameters 36](#_Toc171599814)

[Table 14: Exceedance Monitoring Data Configuration - Aircraft Parameters 36](#_Toc171599815)

[Table 15: Limit Data Configuration 37](#_Toc171599816)

[Table 16: Limit Others Entry Data Configuration - Exceedance Monitoring Data Configuration - Aircraft Parameters 38](#_Toc171599817)

[Table 17: Tasks and Priority 44](#_Toc171599818)

LIST OF FIGURES

[Figure 1: Functional block diagram - Engine Instrument System 8](#_Toc171599819)

[Figure 2: DAU - decomposition 11](#_Toc171599820)

[Figure 3: CMU Block diagram 12](#_Toc171599821)

[Figure 4: NSC message structure 22](#_Toc171599822)

[Figure 5: NOC message structure 27](#_Toc171599823)

# 1 Amendement Record

Table 1: Amendment Record

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Issue No.** | **Description of Amendment** | **Change Request No.** | **Changed By** | **Release Date** |
| 1.1 | Initial Release | NA | Nagesh Ramaiah | 24-Jan-2022 |
| 1.2 | Updated to address review comments.  Below are the sections/requirements that were updated:  Section 2 Overview  Section 4 Acronyms and Definitions  Section 6: References  H698-SRS-CMU-DRQ-29  H698-SRS-CMU-DRQ-30  H698-SRS-CMU-FNC-34  H698-SRS-CMU-FNC-36  H698-SRS-CMU-FNC-38  H698-SRS-CMU-FNC-39  H698-SRS-CMU-DRQ-45  H698-SRS-CMU-DRQ-44  H698-SRS-CMU-DRQ-19  H698-SRS-CMU-FNC-50  H698-SRS-CMU-DRQ-58  Added new requirement  H698-SRS-CMU-DRQ-65  The following requirement is deleted:  H698-SRS-CMU-FNC-42  H698-SRS-CMU-DRQ-64 | PR100025 | Nagesh Ramaiah | 01-Feb-2022 |
| 1.3 | Updated QA comments. Below are the requirement/section details which were updated:   1. In Requirement : H698-SRS-CMU-DRQ-44 – Updated table alignment for the tables – Table 9, Table 10, Table 11, Table 12, Table 13.   2) Updated Section 4 – Acronyms and Definitions. | PR100025 | Nagesh Ramaiah | 03-Feb-2022 |
| 1.4 | Updated Verification method for all requirements from ‘None’ To ‘Testing’ | PR100051 | Vijaya Bhaskar | 9-Apr-2022 |
| 1.5 | Below are the requirement/section details which are updated  H698-SRS-CMU-FNC-36  H698-SRS-CMU-FNC-41  H698-SRS-CMU-DRQ-44  H698-SRS-CMU-DRQ-47  H698-SRS-CMU-DRQ-58  H698-SRS-CMU-DRQ-60  H698-SRS-CMU-DRQ-61  The following requirements are deleted  H698-SRS-CMU-DRQ-45  H698-SRS-CMU-DRQ-46  The following requirements are Added  H698-SRS-CMU-FNC-66  H698-SRS-CMU-DRQ-67  H698-SRS-CMU-DRQ-68 | PR100073 | Divya R | 19-Dec-2022 |
| 1.6 | Updated to address review comments.  Below are the sections/requirements that were updated:  section 15 Output  section 16.1.1 Initialization Components  section 16.3 A825 Communication  section 16.3.4.1 NOC message format  H698-SRS-CMU-DRQ-67  H698-SRS-CMU-DRQ-47  H698-SRS-CMU-FNC-54  H698-SRS-CMU-FNC-50  H698-SRS-CMU-FNC-41  H698-SRS-CMU-DRQ-68  H698-SRS-CMU-FNC-38  H698-SRS-CMU-FNC-36  H698-SRS-CMU-FNC-37  H698-SRS-CMU-DRQ-22  H698-SRS-CMU-DRQ-30  The following Gereral and Derived requirements are Added  H698-SRS-CMU-GEN-69  H698-SRS-CMU-DRQ-70  H698-SRS-CMU-DRQ-71  The following derived requirement is changed to Functional requirement  H698-SRS-CMU-DRQ-30 to H698-SRS-CMU-FNC-30 | PR100073 | Divya R | 23-Dec-2022 |
| 1.7 | Updated as per QA comments and self review  General requirement  Section 16.3.4.1 NOC message format  Section 16.3.3.1 NSC message format  Updated as per self review  Functional requirment  H698-SRS-CMU-DRQ-44 | PR100073 | Divya R | 24-Dec-2022 |
| 1.8 | The below mentioned requirements are updated based on HSIT observation  1)H698-SRS-CMU-FNC-49  2)H698-SRS-CMU-FNC-51 | PR100121 | Divya R | 30-Dec-2022 |
| 1.9 | The below mentioned requirement is updated based on HSIT observation  1) H698-SRS-CMU-FNC-36 | PR100131 | Divya R | 06-01-2023 |
| 2.0 | Updated as per Airbus Spec | PR100169 | Prajwal R | 07-02-2024 |
| 2.1 | The following are updated as per review comments:  H698-SRS-CMU-DRQ-44  Section 2 Overview | PR100169 | Prajwal R | 12-02-2024 |
| 2.2 | The following sections are updated to address QA comments:  Section 6 References  Section 10 Approvals  Section 13 Responsibilities | PR100169 | Prajwal R | 14-02-2024 |
| 2.3 | The following section and tables got modified as per review comments:  Section 2: Overview  Section:10 Approvals  Table 12  Table 13 | PR100222 | Prajwal R | 10-07-2024 |
| 2.4 | The following got modified as per QA comments:  Table 12 | PR100222 | Prajwal R | 11-07-2024 |

# 2 Overview

The EIS consist of Display Unit (DU), Engine Data Acquisition Unit (EDAU), Configuration Management Unit plus NVM (CMU+) and Data Logging Unit (DLU). It will be complemented by pilot input devices and interconnecting harness. The EIS will interface with the Airbus AS532U2 engine sensors and other applicable aircraft systems.

Figure: Functional block diagram - Engine Instrument System describes the functional blocks of Engine Instrument System (EIS).

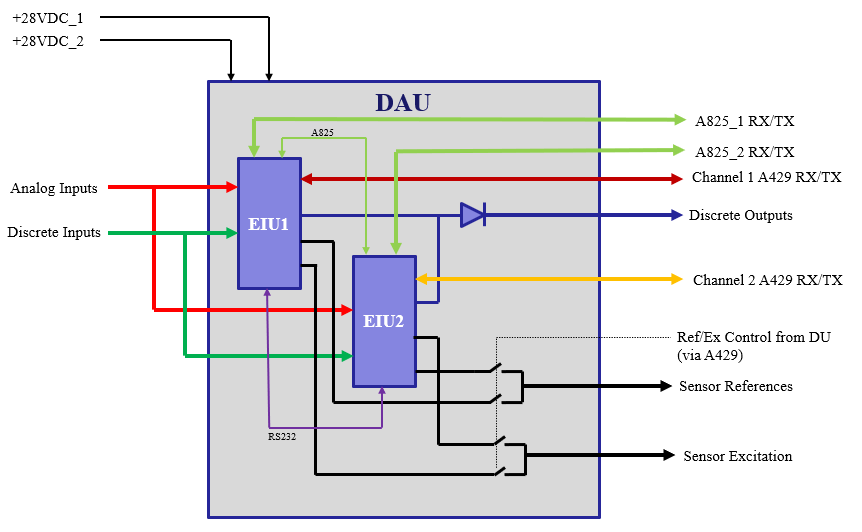
**

Figure 1: Functional block diagram - Engine Instrument System

**Engine Data Acquisition Unit (EDAU)**

The EDAU shall function to process airframe and engine data and transmit this information to be graphically presented on a cockpit display. The EDAU will process analog and discrete inputs, apply user defined logic to those inputs, and may generate aural and visual Caution and Warning alerts.

The EDAU is utilized to perform the following functions:

1. Provide analog to digital conversion of aircraft and engine systems
2. Provide discrete bit information via A429 for Crew Alerts (CAS)
3. Provide field loadable software update via maintenance port
4. Process data
5. Filter analog and digital signal data
6. Receive/Transmit data
7. Provide discrete strapping for multiple aircraft and engine configurations

To increase the reliability of the system, EDAU will have two Data Acquisition Units (DAU) packed together. Both the Data Acquisition Units (DAU) will run the data acquisition application software, process the same inputs and produce same outputs independently at the same time.

The primary function of the DAU will be acquiring analog engine parameters, discrete inputs, and other aircraft system information for processing and converting to a digital format for display via ARINC 429 to four Display Units (DU).

Each DAU will contain three Boards/Modules, each with its own CPU: The Gateway Board (BH35112) for digital buses, the Analog Board (BH35113) for analog inputs and the Discrete Board (BH35114) for discrete I/O and chip burner. Inter-board communication and communication with CMU+ / DLU will be on CAN bus based on ARINC 825 protocol.

**Configuration Module Unit Plus NVM (CMU+)**

The CMU+ stores Aircraft Configuration Data (ACD). The ACD includes options related to the engine parameters, sensor interface, and airframe configurations. The DAU accesses CMU+ on system initialization to determine the current configuration and active interfaces.

The CMU+ is utilized to perform the following functions:

1. Provide configuration data to each DAU utilizing separate, isolated hardware for each DAU
2. Support the DAU system configuration so that a replacement DAU will function the same as

the DAU being replaced after installation and initialization

c. The EDAU shall provide the capability to support different engine and aircraft configurations.

d. The CMU+ shall store item configuration data for entire EIS

e. The CMU+ shall contain system NVM

The CMU+ is internally redundant with two separate configuration sections packaged into one enclosure (separate power supplies). One section CMU+ #1 is connected to DAU#1 through ARINC825 (CAN #1), and the other section CMU+ #2 is connected to DAU#2 through ARINC825 (CAN #1). Internally CMU+ #1 and CMU+ #2 are connected through an independent ARINC825 (CAN #2).

The CMU+ installed in position #1 will have a USB device interface for programming/accessing configuration information in Maintenance Mode.

Figure: DAU - decomposition illustrates the decomposition of the DAU into its various internal modules:

A429

Input

Discrete Input

RS232

Input

Analog Module

Discrete Module

Analog Input

Discrete Input

28V

power Input

Discrete Input

28V

Power Input

Discrete Output

Gateway Module

A429

Output

RS232

Output

CMU

+ Module /

DLU

Module

USB

Input/Output

ARINC825

28V

Power Input

ARINC825

EDAU

Figure 2: DAU - decomposition

The CMU stores the various data related to various aircraft and engine configurations. It is responsible for sending this configuration data to the Gateway. Figure: CMU Block diagram depicts the CMU.

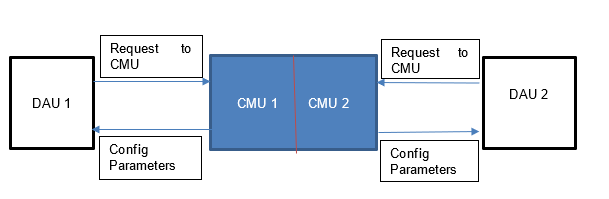


Figure 3: CMU Block diagram

# 3 Objectives

The objective of this document is to specify the Software High Level Requirements for the Configuration Module Unit.

# 4 Acronyms and Definitions

Table 2: List of Acronyms

|  |  |
| --- | --- |
| **Terms** | **Definitions** |
| A825 | ARINC Specification 825 |
| ACD | Aircraft Configuration Data |
| ACK | Acknowledgement |
| ARINC | Aeronautical Radio, Incorporated |
| ASCII | American Standard Code for Information Interchange |
| BIT | Built-In Test |
| CAN | Controller Area Network |
| CBIT | Continuous BIT |
| CMU | Configuration Management Unit |
| CPU | Central Processing Unit |
| CRC | Cyclic Redundancy Check |
| DAU | Data Acquisition Unit |
| DLC | Data Length Count |
| DOC | Data Object Code |
| DRQ | Derived Requirement |
| EDAU | Engine Data Acquisition Unit |
| EIU | Engine Instrument System |
| FID | Function Code Identifier |
| FIFO | First In First Out |
| FSMC | Flexible Static Memory Controller |
| FTP | File Transfer Protocol |
| GPIO | General Purpose Input/Output |
| I/O | Input Output |
| LCC | Logical Communication Channel |
| LCL | Local Bit |
| LED | Light Emitting Diode |
| MOPS | Minimum Operational Performance Specification |
| NOC | Normal Operation Channel |
| NSC | Node Service Channel |
| NVM | Non Volatile Memory |
| NVIC | Nested Vector Interrupt Control |
| PBIT | Power-On BIT |
| PVT | Private Bit |
| QA | Quality Assurance |
| RCI | Redundancy Channel Identifier |
| RSD | Reserved Bit |
| RTC | Real Time Clock |
| RTCA | Radio Technical Commission for Aeronautics |
| Rx | Receive |
| SFC | Service Function Code |
| SMT | Service Message Type |
| SRAM | Static Random Access Memory |
| SVN | SubVersion |
| Tx | Transmit |

# 5 Scope

The Scope of this document is limited to specifying the Software High Level Requirements of the Configuration Module Unit of the DAU.

# 6 References

Table 3: References

|  |  |  |
| --- | --- | --- |
| **Source** | **Document No.** | **Title** |
| RTCA | DO-178B | Software Considerations in Airborne Systems and Equipment Certification |
| Howell | SYS2160SRS | Airbus Helicopter AS532U2 Retrofit Engine Instrument System Requirements Specification |
| Howell Instruments, Inc. | HE0698HA | HARDWARE ARCHITECTURE H698[]-[] CMU+ Series |
| Aeronautical Radio, Inc. | ARINC SPECIFICATION 825-2 | GENERAL STANDARDIZATION OF CAN (CONTROLLER AREA NETWORK) BUS PROTOCOL FOR AIRBORNE USE |
| ALTEN Global Technologies Private Limited | H398-001-002 | Software Development Plan |
| ALTEN Global Technologies Private Limited | H398-001-006 | Software Requirements Standards |
| ALTEN Global Technologies Private Limited | H398-001-004 | Software Configuration Management Plan |

# 7 Assumptions

None

# 8 Outstanding Issues

None

# 9 Document Control

This document is under change control. After baseline, any changes to this document shall be carried out in accordance with Software Configuration Management Plan (H398-001-004).

# 10 Approvals

The document will be reviewed to meet the objectives of RTCA DO-178B level A and approved by ALTEN GT QA.

# 11 Distribution

This document will be distributed over a secure FTP server to Howell Instruments, Inc.

# 12 Traceability

Traceability to SYS2160SRS is provided in Traceability Matrix from SRS to SPEC (H698-002-002-CMU).

# 13 Responsibilities

1. Software Development Lead shall write the Software High Level Requirements in ReMa.
2. Reviewers shall review these Software High Level Requirements based on the RTCA DO-178B SRS check points provided as attributes in ReMa. Developers shall respond to these comments and then change the status to <Looked Into> in ReMa.
3. Once all the comments from Reviewer are made to 'Looked Into' and the status of all Software High Level Requirements in ReMa is <Closed>, the QA shall change the QA Review Status Requirement to "QA Approved".
4. Engineering Manager is responsible for baselining the document to SVN through ReMa as well as checking in the exported document into the path

<http://192.168.1.229/svn/A21HOWBLDAU/V2_H398/ACCORD/SW/Trunk/Documents/SRS>

# 14 Input

* SYS2160SRS
* Software Development Plan (H398-001-002)
* Software Requirements Standards (H398-001-006)

# 15 Output

* Software High Level Requirements Specification of Configuration Module Unit (H698-002-001-CMU)
* Traceability Matrix (H698-002-002-CMU)

# 16 Software High Level Requirements

Software High Level Requirements section explains the Software High Level Requirements of Configuration Module Unit.

## 16.1 Initialization

Initialization section specifies the Software High Level Requirements for the initialization of the CMU.

### 16.1.1 Initialization - Components

The CMU initializes the following components after a Power On:

1. Processor
2. Watchdog timer
3. System Timer
4. Initialize external NVRAM and NOR Flash
5. Onchip and External CAN peripheral configuration
6. Initialize the serial I2C bus
7. Initialize the Real-time Clock
8. Initialize the Continuous Built-In-Test task
9. Enable GPIOB Clock and configures LED and chip select
10. Nested Vector Interrupt Controller (NVIC)

#### 16.1.1.1 Processor

Requirement ID: H698-SRS-CMU-DRQ-19

MOPS: No

Safety Requirement: No

Rationale: The Processor initialization required to execute the software is derived from hardware architecture document HE0698HA and to execute the application software in normal mode

Verification Method: Testing

The CMU shall initialize the processor as follows:

1. Enable HSE (high speed external clock).

2. Enable power interface clock

3. Configure power regulator voltage scale to 1

4. Set CPU clock as System Clock

5. Set PCLK1 = HCLK clock / 4

6. Set PCLK2 = HCLK clock/ 2

7. Set 5 wait states as the latency period on Flash

8. Enable instruction cache.

9. Enable data cache

10. Enable flash prefetch buffer

11. Configure the PLL clock source to 168 MHz for CMU Flight Application Software.

12. Enable PLL.

13. Select PLL as system clock source.

14. Initialize the interrupt vector table in RAM at address 0x00.

15. Set the Vector Table base address at RAM

16. Configure the group priority equal to sixteen and sub-priority to zero

17. Set the priority of the system tick interrupt as 15

18. Setting up the PendSV interrupt into vector table in RAM and set the priority of the PendSV interrupt as 14

19. Enable CRC peripheral clock

20. Initializing the FPU context save not in lazy mode

21. Enable floating point coprocessor

22. Enable Memory Management Fault, Bus Fault and Usage Fa1ult exceptions are enabled by

writing to enable bits(16,17 and 18) of SCB->SHCSR register.

23. Enable Usage Fault when the processor executes a DIV instruction with a divisor

of 0 by setting bit 4 of Configuration Control Register.

24. Configure system select as switch input

25. Set GPIO Pin#14,15 of Port C for GREEN and RED status LED respectively (speed = fast, mode = output mode, operating output type = push pull type, pull-up/pull-down as pull down)

26. Set GPIO Pin#2 Port B for Heartbeat LED (speed = fast, mode = output mode, operating output type = push pull type, pull-up/pull-down as no pull )

#### 16.1.1.2 Watchdog timer

Requirement ID: H698-SRS-CMU-DRQ-20

MOPS: No

Safety Requirement: No

Rationale: The Watchdog Timer initialization is required to monitor the processor idle state

Verification Method: Testing

The CMU Application Software shall initialize the Watchdog timer to 2 seconds.

#### 16.1.1.3 System Timer

Requirement ID: H698-SRS-CMU-DRQ-21

MOPS: No

Safety Requirement: No

Rationale: The System Timer initialization is required to perform periodic scheduling of tasks

Verification Method: Testing

The CMU Application Software shall initialize the System Timer to generate 10000 interrupts every second.

#### 16.1.1.4 CAN peripheral 1

Requirement ID: H698-SRS-CMU-DRQ-22

MOPS: No

Safety Requirement: No

Rationale: The initialization of CAN peripheral is required to communicate to Gateway

Verification Method: Testing

The CMU Application Software shall initialize the CAN1 peripheral as follows:

1. Connect GPIO Pins #8 and #9 of Port B to Alternate Function#9

2. Configure mode, speed, output type and pull-up/pull-down type of GPIO Pins #8 and #9 as Alternate Function mode, fast speed, Output push-pull, and pull-up respectively

3. Enable CAN Clock.

4. Initialize CAN register

5. Disable the triggered communication mode, automatic wake-up mode, non-automatic retransmission mode, Receive FIFO Locked mode and Transmit FIFO priority

6. Enable Bus-Off recovery

7. Set up filter #0 to allow all messages and attach the filter to FIFO #0

#### 16.1.1.5 Nested Vector Interrupt Controller

Requirement ID: H698-SRS-CMU-DRQ-24

MOPS: No

Safety Requirement: No

Rationale: The initialization of Nested Vector Interrupt Controller required to execute the software

Verification Method: Testing

The CMU Application Software shall set up the nested vectored interrupt controller to respond to CAN1 transmit and receive interrupts as follows:

1. Enable NVIC channel#19 and set its Preemption priority and SubPriority to 0

2. Enable NVIC channel#20 and set its Preemption priority and SubPriority to 0

#### 16.1.1.6 FSMC

Requirement ID: H698-SRS-CMU-DRQ-25

MOPS: No

Safety Requirement: No

Rationale: The initialization of FSMC required to execute the software

Verification Method: Testing

The CMU Module shall initialize the FSMC to access the non-volatile SRAM.

#### 16.1.1.7 I2C Initialisation

Requirement ID: H698-SRS-CMU-DRQ-65

MOPS: No

Safety Requirement: No

Rationale: The initialization of I2C peripheral is required to access RTC module.

Verification Method: Testing

The CMU Module shall initialize the I2C peripheral and RTC hardware module and also as part of RTC hardware initialisation module the RTC Battery level and Integrity are checked whether they are good or not.

## 16.2 Modes of Operation

Modes of Operation section specifies the Software High Level Requirements for the Operational Modes of the CMU.

### 16.2.1 Operational Modes

The CMU have the following Operational Modes

1. Power-On BIT mode

2. Normal Mode

3. Error Mode

### 16.2.2 PBIT mode

Requirement ID: H698-SRS-CMU-DRQ-28

MOPS: No

Safety Requirement: No

Rationale: This is required to identify the PBIT mode of application software

Verification Method: Testing

The CMU shall perform the Power-on Built-in Tests during PBIT mode.

### 16.2.3 Normal Mode

Requirement ID: H698-SRS-CMU-DRQ-29

MOPS: No

Safety Requirement: No

Rationale: This is required to identify the activities to be done in Normal mode of application software

Verification Method: Testing

The CMU shall perform the following activities in Normal mode:

1. Establish a connection with Gateway module.
2. Transfer Aircraft configuration data, NVM parameters and Software part numbers along with CRCs to Gateway module over A825
3. Perform CBIT

4. Flash Heartbeat LED to indicate the execution health status of the CMU

### 16.2.4 Error Mode

Requirement ID: H698-SRS-CMU-FNC-30

MOPS: No

Safety Requirement: No

Rationale: NA

Verification Method: Testing

In Error mode, the CMU shall perform the following:

1. Cease all current activities

2. Switch the HeartBeat LED as Steady On.

3. Check the CRC of Error Flash section and write error to Flash.

Table 4: Status LED Blink Rate

|  |  |
| --- | --- |
| **Error Type** | **Description** |
| Aircraft Config Data CRC Error | Mismatch in the computed and Stored ACD Application CRC value or DLU CRC fail. |
| Memory Error | Failure of RAM or NVM Data CRC test in PBIT. |
| CPU Error | Failure of CPU test in PBIT. |
| INTR Error | Failure due to following Fault interrupts: NonMaskable, HardFault, MemManage, BusFault UsageFault and Spurious Fault |
| Stack Error | Failure of Stack test in CBIT. |
| OS Kernel Error | Failure of OS Kernel |
| TBase Error | Failure of OS Task |

Note:

The LED Code represents the code displayed on the front panel of the CMU+ when a fatal system error has occurred. It consists of three groups of four bits (red (R) or green (G) illumination) separated by a short period of no illumination. Each four bits is the binary equivalent of a decimal number.

Forst 4 bits represents Application ID which is 5 for CMU application software.

Next 4 bits represents the error code.

Next 4 bits represents the sub error code.

For example, the pattern "R-G-R-G R-G-R-R G-R-G-G" represents the code 5 4 11.

## 16.3 A825 Communication

This section specifies the Software High Level Requirements for the A825 communication.

The CMU is internally redundant with two separate CAN sections packaged into one enclosure. One section CMU #1 is connected to EIU #1 through ARINC825 (CAN #1) and the other section CMU #2 is connected to EIU #2 through ARINC825 (CAN #1).

Note: Refer ARINC SPECIFICATION 825-2 for A825 communication protocol.

### 16.3.1 Logical Communication Channel

Requirement ID: H698-SRS-CMU-FNC-32

MOPS: No

Safety Requirement: No

Rationale: NA

Verification Method: Testing

The CMU Application Software shall use the following Logical Communication Channels to communicate over the A825 bus

1. Node Service Channel (NSC)

2. Normal Operation Channel (NOC)

### 16.3.2 Invalid A825 message

Requirement ID: H698-SRS-CMU-FNC-33

MOPS: No

Safety Requirement: No

Rationale: NA

Verification Method: Testing

The CMU Application Software shall ignore an A825 message if it is invalid.

Note: An invalid message is one which does not satisfy the specified NSC/NOC message formats specified in NSC and NOC message format.

### 16.3.3 NSC channel

This section specifies the Software High Level Requirements for messages sent over NSC channel.

The Node Service Channel is used for peer-to-peer communication.

#### 16.3.3.1 NSC message format

The CMU Application Software will use the message structure depicted in Figure: NSC message structure while using the NSC channel

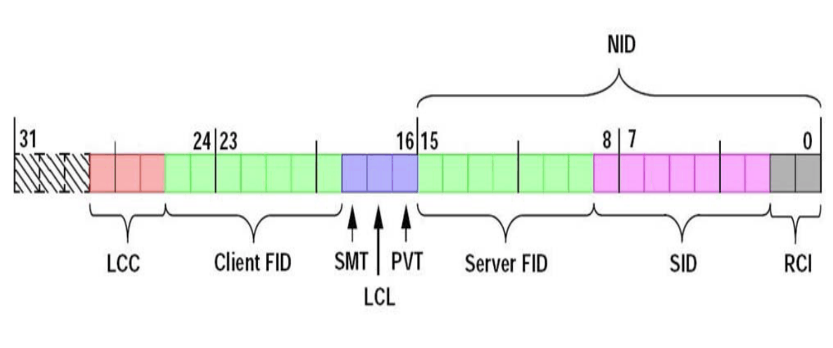


Figure 4: NSC message structure

Note:

i. LCC - Indicates the Logical Communication Channel

ii. Client FID - always 42

iii. SMT - Indicates whether message is Request (1) or Response (0)

iv. LCL - messages is designated only for the network the transmitting node resides in - always 1

v. PVT - messages which have no meaning to nodes other than those which are specifically programmed to use them - always 1. The “Private” (PVT) bit in the CAN message ID is set to zero (0) for all service control messages (non-data transfer messages) and set to one (1) for all data transfer messages.

vi. Server FID - Always 42

vii. Server ID - Specifies the recipient node for the message

viii. RCI - Redundancy Channel Identifier - indicates whether message is Onside (0) or Offside (1)

#### 16.3.3.2 NSC messages

Requirement ID: H698-SRS-CMU-FNC-36

MOPS: No

Safety Requirement: No

Rationale: NA

Verification Method: Testing

Table 5: Service Function Codes

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SFC** | **Client FID** | **SMT** | **LCL** | **PVT** | **Server FID** | **Server ID** | **Rx/Tx** | **RCI** | **DLC** | **Description** | **Payload** |
| 49155 | 42 | 1 | 1 | 1 | 42 | 6 | Rx | 0 or 1 | 2 | Gateway commands to the CMU to send its RCI, operation mode, and data validity. | The first two bytes of the payload contain the SFC. |
| 49155 | 42 | 0 | 1 | 1 | 42 | 6 | Tx | 0 | 5 | Response by CMU to the SFC = 49155 request | The first two bytes of the payload contain the SFC. The third byte contains the RCI (0 or 1), the fourth byte contains always value 1, the fifth contains the result of a CRC on the data store (0 = FAIL, 1 = PASS). |
| 3 | 42 | 1 | 1 | 0 | 42 | 6 | Rx | 0 | 8 | Gateway request to CMU to transmit Aircraft Configuration Data Table. | The first two bytes of the payload contain the SFC. The third and fourth byte together contains value 512.Other bytes contain 0. |
| 3 | 42 | 0 | 1 | 0 | 42 | 6 | Tx | 0 | 8 | CMU responds to the Gateway with the number of ACD bytes to be downloaded. | The first two bytes of the payload contain the SFC. The third, fourth byte contains 1. Fifth and sixth byte contains the number of bytes acknowledged to the Gateway (8k).  Other bytes contain 0. |
| 3 | 42 | 1 | 1 | 0 | 42 | 6 | Rx | 0 | 8 | Gateway request to CMU with the download size request (8k) | The first two bytes of the payload contain the SFC. The third byte contains 1, fourth byte contains 0. Fifth and sixth byte contains the number of downloaded bytes requested by the Gateway to CMU.  Other bytes contain 0. |
| 3 | 42 | 0 | 1 | 0 | 42 | 6 | Tx | 0 | 8 | CMU response to Gateway with the download size (8k) | The first two bytes of the payload contain the SFC. The third byte contains 1, fourth byte contains 0. Fifth and sixth byte contains the number of downloaded bytes acknowledged by the CMU to Gateway.  Other bytes contain 0. |
| Start of ACD transfer begins | 42 | 0 | 1 | 0 | 42 | 6 | Tx | 0 | 8 | Start of ACD Transfer from CMU side to Gateway | CMU transfer 8k bytes of ACD data to Gateway |
| 3 | 42 | 1 | 1 | 0 | 42 | 6 | Rx | 0 | 3 | Gateway acknowledgement to CMU after finish of ACD transfer from CMU | The first two bytes of the payload contain the SFC. The third byte can be 0 or 1 based on receive success or fail. |

Note:

1. Rx/Tx - Indicates whether CMU module Receives (Rx) or Transmits (Tx) the message

2. DLC - Data Length Count - size of the payload in bytes

3. RCI is 0 for CMU1 and CMU2

4. Refer NSC Message Format

#### 16.3.3.3 NSC message - RCI request

Requirement ID: H698-SRS-CMU-FNC-37

MOPS: No

Safety Requirement: No

Rationale: NA

Verification Method: Testing

For any A825 message on NSC channel with SFC = 49155 (RCI request) the CMU Application Software shall respond with the SFC = 49155 response message on the same channel.

Note: Refer H698-CMU-SRS-FNC-36 for the request and response message.

#### 16.3.3.4 NSC message - Configuration Data request

Requirement ID: H698-SRS-CMU-FNC-38

MOPS: No

Safety Requirement: No

Rationale: NA

Verification Method: Testing

After successful RCI request and response, for any A825 message on NSC channel with SFC = 3 (Aircraft Configuration Data request) received on the ARINC825 port (CAN #1), the CMU Application Software shall respond as follows:

1. Transmit response message for SFC = 3 message (Configuration Data response) on the same ARINC825 port (CAN #1).
2. Wait for ACK from corresponding Gateway Module.
3. On receipt of ACK from Gateway, transmit the configuration data (1024 blocks of 8 bytes) with the ARINC825 message fields set as mentioned below :

     - LCC to NSC

     - Client Functional Code Identifier to 42

     - SMT to 0.

     - LCL to 1

     - PVT to 1

     - Server Functional Code Identifier to 42

     - Server Identifier to 6

     - RCI is 0 for both CMU1 and CMU2

     - DLC to 8.

### 16.3.4 NOC channel

This section specifies Software High Level Requirements for messages sent over NOC channel.

The Normal Operation Channel is used for one-to-many communication and/or for sending aircraft operational data.

#### 16.3.4.1 NOC message format

The CMU Application Software will use the message structure depicted in Figure: NOC message structure while using the NOC channel



Figure 5: NOC message structure

Note:

i. LCC - Indicates the Logical Communication Channel

ii. Source FID - always 42

iii. RSD - Reserved - always 0

iv. LCL - messages is designated only for the network the transmitting node resides in - always 1

v. PVT - messages which have no meaning to nodes other than those which are specifically programmed to use them - always 1

vi. DOC - Data Object Code

vii. RCI - Redundancy Channel Identifier - indicates whether message is Offside(0)

#### 16.3.4.2 NOC messages

Requirement ID: H698-SRS-CMU-FNC-41

MOPS: No

Safety Requirement: No

Rationale: NA

Verification Method: Testing

The CMU Application Software shall use the Document Object Codes specified in Table**:** Document Object Codes - NOC in its A825 communication over the Normal Operation Channel:

Table 6: Document Object Codes - NOC

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **DOC** | **Source FID** | **RSD** | **LCL** | **PVT** | **RCI** | **Rx/Tx** | **DLC** | **Description** | **Payload** |
| 2700 | 42 | 0 | 1 | 1 | 0 or 1 | Rx | 0 | Gateway triggers the CMU to send its Bootloader Application, Bootloader Configuration, Flight Application’s Part Number and CRC along with Aircraft Configuration Data’s CRC | NA |
| 2806 | 42 | 0 | 1 | 1 | 0 or 1 | Tx | 8 | Bootloader Application Part number first 8 bytes | Each ASCII character (1-8) of the Bootloader Application Part Number is reported as 1 byte of the payload. |
| 2807 | 42 | 0 | 1 | 1 | 0 or 1 | Tx | 8 | Bootloader Application Part number last 8 bytes | Each ASCII character (9-16) of the Bootloader Application Part Number is reported as 1 byte of the payload. |
| 2808 | 42 | 0 | 1 | 1 | 0 or 1 | Tx | 4 | Bootloader Application CRC (32-Bit) | 8 Bits of the Bootloader CRC is reported as 1 byte of the payload. Least significant 8 Bits of the Bootloader CRC appears as the first byte of the payload. Next 8 Bits as second byte of the payload and so on. |
| 2812 | 42 | 0 | 1 | 1 | 0 or 1 | Tx | 8 | Bootloader Configuration Part number first 8 bytes | Each ASCII character (1-8) of the Bootloader Configuration Part Number is reported as 1 byte of the payload. |
| 2813 | 42 | 0 | 1 | 1 | 0 or 1 | Tx | 8 | Bootloader Configuration Part number last 8 bytes | Each ASCII character (9-16) of the Bootloader Configuration Part Number is reported as 1 byte of the payload. |
| 2814 | 42 | 0 | 1 | 1 | 0 or 1 | Tx | 4 | Bootloader Configuration CRC (32-Bit) | 8 Bits of the Bootloader Configuration CRC is reported as 1 byte of the payload. Least significant 8 Bits of the Bootloader CRC appears as the first byte of the payload. Next 8 Bits as second byte of the payload and so on. |
| 2815 | 42 | 0 | 1 | 1 | 0 or 1 | Tx | 8 | Flight Application Part number first 8 bytes | Each ASCII character (1-8) of the Flight Application Part Number is reported as 1 byte of the payload. |
| 2816 | 42 | 0 | 1 | 1 | 0 or 1 | Tx | 8 | Flight Application Part number last 8 bytes | Each ASCII character (9-16) of the Flight application Part Number is reported as 1 byte of the payload. |
| 2817 | 42 | 0 | 1 | 1 | 0 or 1 | Tx | 4 | Flight Application CRC (32-Bit) | 8 Bits of the Flight Application CRC is reported as 1 byte of the payload. Least significant 8 Bits of the Bootloader CRC appears as the first byte of the payload. Next 8 Bits as second byte of the payload and so on. |
| 2818 | 42 | 0 | 1 | 1 | 0 or 1 | Tx | 4 | Aircraft Configuration Data CRC (32-Bit) | 8 Bits of the Aircraft Configuration Data CRC is reported as 1 byte of the payload. Least significant 8 Bits of the Configuration CRC appears as the first byte of the payload. Next 8 Bits as second byte of the payload and so on. |
| 3000 | 42 | 0 | 1 | 1 | 0 or 1 | Rx | 0 | Trigger for CMU+ to send NVM data to Gateway | NA |
| 3001 to 3022 | 42 | 0 | 1 | 1 | 0 or 1 | Tx | Number of doc Ids \* 8 bytes | NVM data sent from CMU+ to Gateway | CMU Sends NVM Parameters to Gateway. Refer H698-SRS-CMU-DRQ-47 for NVM structure details. |

**Note**:

1. Rx/Tx - Indicates whether CMU Receives (Rx) or Transmits (Tx) the message.

2. DLC - Data Length Count - size of the payload in bytes.

3. RCI is 0 for CMU1 and CMU2.

4. Refer NOC Message Format

## 16.4 Aircraft Configuration Data

This section specifies the Configuration data that is provided by CMU whenever the Gateway requests for Configuration data On Power-On. Aircraft Configuration Data section provides the Software High Level Requirements for the Aircraft Configuration Data (ACD) for the CMU. The CMU contains the Aircraft Configuration Data used for Normal operation. On power-on, the EIU requests for this data from the CMU.

### 16.4.1 Aircraft configuration data

Requirement ID: H698-SRS-CMU-DRQ-44

MOPS: No

Safety Requirement: No

Rationale: Required to define the various sections of Aircraft Configuration Data used to make the software configurable

Verification Method: Testing

The CMU Application Software shall read the Aircraft Configuration Data stored in location 0x0800C000 as per the below format and transmit to Gateway in NSC messages as mentioned in H698-SRS-CMU-FNC-36

Table 7: Aircraft Configuration Data

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Description** | **Range** |
| Aircraft Configuration Software revision Number | Aircraft Configuration(ACD) Part Number | Format:<H108E-713><space><-><x><.><yy><space>  where,  x is major release number  yy is minor release number.  16-character string(16 Bytes) |
| Ohm Offset Channel, 8 | Analog Calculation ohm offset channel | 0x00 to 0xFF (Each 1 Byte) |
| Ohm Offset Value, 8 | Analog Calculation ohm offset value to be used | 0x0000 to 0xFFFF (Each 2 Bytes) |
| VAC Offset Signal, 8 | Analog Calculation VAC offset channel | 0x00 to 0xFF(Each 1 Byte) |
| VAC Offset Reference, 8 | Analog Calculation VAC offset Reference value | 0x00 to 0xFF(Each 1 Byte) |
| Table offset, 128 | Table offset value | 0x00 to 0xFF(Each 1 Byte) |
| Audio Config, 8 | Audio Configuration | Refer Audio Configuration – Aircraft Parameters section |
| CMU ACD Version | Version to track structure changes | 4 Byte version number |
| Offside Analog Offset | Indicates Offside Analog Offset Value | 4 Bytes |
| Number OF Engines | Indicates number of engines installed on the aircraft | 1-2(4 Bytes) |
| Engine Data | Indicates Engine Parameter data | Refer Engine Data – Aircraft Parameters section |
| Air Data | Indicates Air data parameters | Refer Air Data – Aircraft Parameters section |
| Number of ARINC 429 inputs | Indicates Number of ARINC 429 inputs | 0-255 (1 Byte) |
| ARINC429 Input Data | ARINC 429 Input Data parameters | Refer ARINC 429 Input Data – Aircraft Parameters section |
| Number of ARINC 429 Binary Outputs | Indicates Number of ARINC 429 Binary Outputs | 0-255 (1 Byte) |
| ARINC 429 Output Binary Data | ARINC 429 Output Binary Data parameters | Refer ARINC 429 Output Binary Data – Aircraft Parameters section |
| Number of ARINC 429 Discrete Outputs | Indicates Number of ARINC 429 Discrete Outputs | 0-255 (1 Byte) |
| ARINC 429 Output Discrete Data | ARINC 429 Output Discrete Data Parameters | Refer ARINC 429 Output Discrete Data – Aircraft Parameters section |
| Number of exceedance monitoring data | Indicates Number of exceedance monitoring data | 0-255 (1 Byte) |
| Exceedance monitoring data | Indicates Exceedance monitoring Parameters data | Refer Exceedance monitoring data – Aircraft Parameters section |

The Audio Configuration section shall have the parameters specified in Table : Audio Configuration - Aircraft Parameters for each Audio Alerts

Table 8: Audio Configuration – Aircraft Parameters

|  |  |
| --- | --- |
| **Parameter** | **Range** |
| Enable | 0 = Disable  1 = Enable |
| Type | 0 = None  1 = Bell  2 = Cavalry  3 = Cricket  4 = Steady sound  5 = Lyre bird sound  6 = Master Warning sound  7 = Stall warning sound |
| Enable AI | 0 = Disable  1 = Enable |
| Audio inhibit | 0 = Disable  1 = Enable |
| Enable Mute | 0 = Disable  1 = Enable |
| Mute | 0 = Sound is muted in this aircraft configuration  1 = Sound is not muted in this aircraft configuration |

The Engine Data Configuration section shall have the parameters specified in Table : Engine Data Configuration - Aircraft Parameters for each of the engine parameters(TOT, EOP\_SIG, EOT, TRQ, NG, MGBT, LRHP, MGBOP, TOTR)

Table 9: Engine Data Configuration – Aircraft Parameters

|  |  |
| --- | --- |
| **Parameter** | **Range** |
| Enable | 0 = Disable  1 = Enable |
| Input Location | 0 = None  1 = Discrete Input  2 = Discrete Output  3 = Analog Input  4 = A429 Input  5 = A422 Input  6 = Derived signal  7 = Total Loss of Channel |
| Location Number | 1 - 100 |
| Resolution | {1, 10, 100, 1000} |
| Table Number | 0=NONE  1=TOT  2=EOP\_SIG  3=EOT  4=TRQ  5=FP  6=NG  7=MGBT  8=LRHP  9=MGBOP  10=TOTR  11=MAX |
| Filter Type | 0 = None  1 = First Order filter  2 = Averaging filter |
| Filter Constant | 0 to 1 |
| Start Range | -10,000 to +10,000 |
| End Range | -10,000 to +10,000 |

The Air Data Configuration section shall have the parameters specified in Table : Air Data Configuration - Aircraft Parameters for each of the Air Data parameters(MGBT, LHP, RHP, MGBOP)

Table 10: Air Data Configuration – Aircraft Parameters

|  |  |
| --- | --- |
| **Parameter** | **Range** |
| Enable | 0 = Disable  1 = Enable |
| Input Location | 0 = None  1 = Discrete Input  2 = Discrete Output  3 = Analog Input  4 = A429 Input  5 = A422 Input  6 = Derived signal  7 = Total Loss of Channel |
| Location Number | 1 - 100 |
| Resolution | {1, 10, 100, 1000} |
| Table Number | 0=NONE  1=TOT  2=EOP\_SIG  3=EOT  4=TRQ  5=FP  6=NG  7=MGBT  8=LRHP  9=MGBOP  10=TOTR  11=MAX |
| Filter Type | 0 = None  1 = First Order filter  2 = Averaging filter |
| Filter Constant | 0 to 1 |
| Start Range | -10,000 to +10,000 |
| End Range | -10,000 to +10,000 |

The ARINC 429 Input Data Configuration section shall have the parameters specified in Table : ARINC429 Input Data Configuration - Aircraft Parameters for each of the ARINC429 Input Data as specified in the Number of ARINC 429 Inputs.

Table 11: ARINC429 Input Data Configuration – Aircraft Parameters

|  |  |
| --- | --- |
| **Parameter** | **Range** |
| Label | 0 - 255 |
| Channel | 0 – 10 |
| Label Type | 0 = Discrete  1 = BNR  2 = Others |
| Sign Bit | 0 = Used as Sign bit  1 = Used as Data bit  2 = Used for direction  3 = Others |
| sig\_bits | 0-20 |
| Update Rate | 0 – 1000 |
| SDI | 0 = SDI\_0  1 = SDI\_1  2 = SDI\_2  3 = SDI\_3  4 = SDI\_ALL |
| Parity | 0, 1 |
| Resolution | 1-1000 |
| Offset Value | 0 – 10000 |
| Missing Time | 0 – 1000 |
| Default Value | 0 – 10000 |
| Source | 0 = None  1 = Display  2 = Others |

The ARINC 429 Output Binary Data Configuration section shall have the parameters specified in Table : ARINC429 Output Binary Data Configuration - Aircraft Parameters for each of the ARINC429 Output Binary Data as specified in the Number of ARINC 429 Output Binary.

Table 12: ARINC429 Output Binary Data Configuration – Aircraft Parameters

|  |  |
| --- | --- |
| **Parameter** | **Range** |
| Label | 0 - 255 |
| Channel | 0 – 10 |
| Label Type | 0 = Discrete  1 = BNR  2 = Others |
| Sign Bit | 0 = Used as Sign bit  1 = Used as Data bit  2 = Used for direction  3 = Others |
| Significant Bits | 0-20 |
| SDI | 0 – 4 |
| Parity | 0, 1 |
| Resolution | 1 – 1000 |
| Offset Value | 0 – 10000 |
| Update Rate | 0-1000 |
| Default Value | 0 – 10000 |
| Source | 0 = None  1 = Display  2 = Others |
| Raw data packet resolution | 0-10000 |

The ARINC 429 Output Discrete Data Configuration section shall have the parameters specified in Table : ARINC429 Output Discrete Data Configuration - Aircraft Parameters for each of the ARINC429 Output Discrete Data as specified in the Number of ARINC 429 Output Discrete.

Table 13: ARINC429 Output Discrete Data Configuration – Aircraft Parameters

|  |  |
| --- | --- |
| **Parameter** | **Range** |
| Label | 0 - 255 |
| Channel | 0 – 10 |
| Label Type | 0 = Discrete  1 = BNR  2 = Others |
| Sign Bit | 0 = Used as Sign bit  1 = Used as Data bit  2 = Used for direction  3 = Others |
| Significant Bits | 0-20 |
| SDI | 0 – 4 |
| Parity | 0, 1 |
| Resolution | 1-1000 |
| Offset Value | 0 – 10000 |
| Update Rate | 0-1000 |
| Default Value | 0 – 10000 |
| Source | 0 = None  1 = Display  2 = Others |
| Raw data packet resolution | 0-10000 |

The Exceedance Monitoring Data Configuration section shall have the parameters specified in Table : Exceedance Monitoring Data Configuration - Aircraft Parameters for each of the Exceedance Monitoring Parameters.

Table 14: Exceedance Monitoring Data Configuration - Aircraft Parameters

|  |  |
| --- | --- |
| **Parameter** | **Range** |
| CAS Message | 0 = No CAS  1 = ENG1 Exceed  2 = ENG2 Exceed  3 = RTR Overspeed  4 = XMSN OVERTQ  5 = OEI Time Used 1  6 = OEI Time Used 2  7 = Number Of CAS  EOT = Number Of CAS |
| Event ID | 0 = None  1 = 30 sec  2 = 2-5 min  3 = 2 min  4 = OEI Usage  5 = 30 min  7 = Limit Exceed  8 = Start Exceed  10 = 5 min  12 = AEO Exceed  13 = Number of event |
| Parameter ID | 0 = NPNR  1 = TRQ  2 = MGT  3 = NG |
| System ID | 0 = XMSN  1 = ENG1  2 = ENG2  3 = RTR |
| Limit Data | Refer Limit Data Configuration - Exceedance Monitoring Data Configuration - Aircraft Parameters |
| Exceed Flag | 0 = FALSE  1 = TRUE |
| Previous Exceed Flag | 0 = FALSE  1 = TRUE |
| Start Time | 0 |
| End Time | 0 |
| Date | 0 |
| Peak | 0 |
| Duration | 0 |
| Limit Others Entry Data | Refer Limit Others Entry Data Configuration - Exceedance Monitoring Data Configuration - Aircraft Parameters |

Table 15: Limit Data Configuration

|  |  |
| --- | --- |
| **Parameter** | **Range** |
| Limit Type | 0 = High Low  1 = Others |
| Limit Range | 0 = Above  1 = Above Or Equal  2 = Below  3 = Below Or Equal |
| Limit Condition | 0 = STATE\_ALWAYS  1 = STATE\_SHUTDOWN  2 = STATE\_START  3 = STATE\_AEO  4 = STATE\_TAKEOFF  5 = STATE\_OEI1  6 = STATE\_OEI2  7 = STATE\_TAKEOFF\_AEO |
| Limit Value | 0-10000 |
| High Limit Value | 0-10000 |
| Low Value Range | 0-10000 |
| High Value Range | 0-10000 |
| Range Check Variable | 0 = NR  1 = QM  2 = NP1  3 = NG1  4 = QE1  5 = MGT1  6 = NP2  7 = NG2  8 = QE2  9 = MGT2  10 = QE  11 = BLANK |
| Maximum Time | 0 - 300 |
| P Timer | 0 |
| X Timer | 0 |
| Upper Limit Exceed | 0 = FALSE  1 = TRUE |

Table 16: Limit Others Entry Data Configuration - Exceedance Monitoring Data Configuration - Aircraft Parameters

|  |  |
| --- | --- |
| **Parameter** | **Range** |
| Limit Others Type | 0 = Fixed  1 = Range  2 = Nil |
| Limit Others Start Time | 0-300 |
| Limit Others End Time | 0-300 |
| Limit Others value | 0-10000 |
| Limit Others conversion value | -100 to 100 |

## 16.5 NVM Data Format

Requirement ID: H698-SRS-CMU-DRQ-47

MOPS: No

Safety Requirement: No

Rationale: Required to define the various sections of NVM Data structure

Verification Method: Testing

CMU shall transfer the NVM data present in NVM working Location to Gateway (as part of Doc ID 3001 to 3022) when the NVM read/write test passed as per H698-SRS-CMU-FNC-66.

## 16.6 BIT

This section specifies Software High Level Requirements for Built-In Tests.

The CMU Application Software performs tests to check its health. Some tests are performed only once during power-on and some tests are performed continuously at specific intervals during normal operation.

### 16.6.1 PBIT

Requirement ID: H698-SRS-CMU-FNC-49

MOPS: No

Safety Requirement: No

Rationale: NA

Verification Method: Testing

The CMU shall do the following tests as part of PBIT

1. CPU Test as per H698-SRS-CMU-FNC-52
2. CRC Test as per H698-SRS-CMU-FNC-54
3. NVM Test as per H698-SRS-CMU-DRQ-67
4. RAM Test as per H698-SRS-CMU-FNC-51

### 16.6.2 CBIT

Requirement ID: H698-SRS-CMU-FNC-50

MOPS: No

Safety Requirement: No

Rationale: NA

Verification Method: Testing

As part of CBIT, the CMU shall perform the Stack test as follows:

1. Read the location at 70% of the stack length

2. If the value read is not equal to 0xDEADDEAD then the test is failed else it is passed

Note: Stack Test of the following tasks:

A825CommTask

Init Task

Continuous Bit Task

Idle Task

### 16.6.3 RAM check

Requirement ID: H698-SRS-CMU-FNC-51

MOPS: No

Safety Requirement: No

Rationale: NA

Verification Method: Testing

The CMU shall perform a RAM check as follows:

1. Write the pattern 0xAAAAAAAA to RAM memory

2. Read back the values from the RAM memory

3. Write the pattern 0x55555555 to RAM memory

4. Read back the values from the RAM memory

5. Write the pattern 0xFFFFFFFF to RAM memory

6. Read back the values from the RAM memory

7. Write the pattern 0x00000000 to the same RAM memory

8. Read back the values from the RAM memory

9. If all the values read in Steps #2, #4, #6 and #8 match with the values written in the previous check then the check is a pass else check is failed.

Note: This test is performed on RAM and CCRAM memory.

### 16.6.4 CPU check

Requirement ID: H698-SRS-CMU-FNC-52

MOPS: No

Safety Requirement: No

Rationale: NA

Verification Method: Testing

The CMU Application Software shall perform a CPU test as follows:

1. Perform the following operations with known operands

Arithmetic Operators - addition, subtraction, multiplication, division

Logical Operators - AND, OR, XOR

2. If the obtained result matches with the known result, then the test is pass else it is fail.

Note: Flags - Carry, Overflow, Zero flags are also tested while performing the above operations.

### 16.6.5 BIT error

Requirement ID: H698-SRS-CMU-FNC-53

MOPS: No

Safety Requirement: No

Rationale: NA

Verification Method: Testing

The CMU Application Software shall enter the Error mode if any of the tests in PBIT/CBIT fail.

Note:

Failure such as CRC failure, CPU failure, Stack overflow are considered as critical errors and during such failure conditions the CMU should stop its normal operational activities and enter into Error Mode.

### 16.6.6 CRC check

Requirement ID: H698-SRS-CMU-FNC-54

MOPS: No

Safety Requirement: No

Rationale: NA

Verification Method: Testing

The CMU Application Software shall verify the CRC of the CMU Aircraft Configuration data as part of PBIT

CRC test is performed as follows:

1. Calculate the CRC over the required memory region. (CRC polynomial is provided below)

2. Compare the calculated CRC with the CRC present in the memory.

3. If both values match, then CRC test is pass, else the test is a fail.

Note: The CRC polynomial is X^32+X^26+X^23+X^22+X^16+X^12+X^11+X^10+X^8+X^7+X^5+X^4+X^2+X^1+X^0

Aircraft Configuration data present in location 0x0800C000 and CRC of Aircraft Configuration data present in location 0x0800DFFC

### 16.6.7 NVM read/write test

Requirement ID: H698-SRS-CMU-FNC-66

MOPS: No

Safety Requirement: No

Rationale: NA

Verification Method: None

The CMU Application Software shall perform the NVRAM Read/Write test and send the status of the same to Gateway as part of NVM data over A825.

### 16.6.8 InitNvram

Requirement ID: H698-SRS-CMU-DRQ-67

MOPS: No

Safety Requirement: No

Rationale: This requirement has been derived to ensure the CMU module should copy NVM data with valid CRC.

Verification Method: Testing

The CMU module shall copy NVM data with valid CRC and with the latest timestamp to NVM working location.

Enters into error mode, if there are no valid NVM data(No valid CRC for all 3 locations).

Note:

NVM data present in the following 3 areas

Working Location with starting address as 0x60008000

Backup 1 with starting address as 0x60010000

Backup 2 with starting address as 0x60018000

### 16.6.9 CMU switches to DLU

Requirement ID: H698-SRS-CMU-DRQ-68

MOPS: No

Safety Requirement: No

Rationale: This requirement has been derived to ensure the CMU Application Software should switch to DLU and should check computed CRC against stored CRC for DLU flight application before switching.

Verification Method: Testing

CMU Application Software shall check computed CRC against stored CRC for DLU flight application software on any of the following conditions

i) After completing transmission of NVM parameters to Gateway module.

ii) After 7 seconds of powerup if CMU not received any of the ACD,NVM and Part Number request message from gateway.

switch to DLU flight application software if the CRC check matches.

otherwise enters into Error mode.

## 16.7 Kernel and Scheduler

This section specifies the Software High Level Requirements for the Kernel and Scheduler used in the CMU Module.

### 16.7.1 Kernel/Scheduler - Initializations

Requirement ID: H698-SRS-CMU-DRQ-56

MOPS: No

Safety Requirement: No

Rationale: This requirement has been derived to ensure proper initialisation so that all tasks are functioning correctly

Verification Method: Testing

The CMU Application Software Module shall initialize the data structures used by the Kernel and Scheduler after PBIT.

### 16.7.2 Scheduler - Algorithm

Requirement ID: H698-SRS-CMU-DRQ-57

MOPS: No

Safety Requirement: No

Rationale: This requirement has been derived to ensure that all tasks are functioning correctly

Verification Method: Testing

The CMU Application Software Module shall use a priority based pre-emptive Scheduler.

### 16.7.3 Tasks

Requirement ID: H698-SRS-CMU-DRQ-58

MOPS: No

Safety Requirement: No

Rationale: This requirement has been derived to differentiate the activities that are to be performed by different tasks

Verification Method: Testing

The CMU Application Software Module shall create the tasks described in Table: Tasks and Priority during initialization.

Table 17: Tasks and Priority

|  |  |  |  |
| --- | --- | --- | --- |
| **Tasks** | **Frequency** | **Priority** | **Description** |
| A825CommTask | 50ms/invoked by ISR | 4 | Monitor Data Download Service timers and toggle status LED |
| Init Task | 500ms | 15 | This task initializes the peripherals, scheduler, kernel and creates other tasks. |
| Continuous Bit Task | 60ms | 16 | This Task is responsible for the CBIT for the stack overflow condition.  It is performed on all created Tasks to check 70% on stack usage |
| Idle Task | - | 63 | It is executed when there are no other Ready tasks. |

### 16.7.4 Semaphores

Requirement ID: H698-SRS-CMU-DRQ-59

MOPS: No

Safety Requirement: No

Rationale: This requirement has been derived to specify the mechanism used for task activation

Verification Method: Testing

The CMU Application Software Module shall use a separate binary semaphore for each task to activate the task.

### 16.7.5 Task activation by semaphore

Requirement ID: H698-SRS-CMU-DRQ-60

MOPS: No

Safety Requirement: No

Rationale: This requirement has been derived to specify the mechanism used for task activation

Verification Method: Testing

The CMU Application Software Module shall use the System Timer interrupt to release the semaphore for tasks as per the timing mentioned in H698-SRS-CMU-DRQ-58.

### 16.7.6 Task structure

Requirement ID: H698-SRS-CMU-DRQ-61

MOPS: No

Safety Requirement: No

Rationale: This has been derived to ensure that all tasks have a common structure for executing their defined activities

Verification Method: Testing

The CMU Module shall behave as per the following steps to execute the tasks mentioned in H698-SRS-CMU-DRQ-58:

1. Wait for the semaphore

2. On semaphore release, perform the task activities once

3. Infinitely repeat Steps#1 and #2.

Note: The Idle task is an exception. It is activated when there are no other tasks which are ready.

## 16.8 Software Part Number

The CMU binaries has a unique part number to identify itself.

### 16.8.1 Software Part Number – Application Software

Requirement ID: H698-SRS-CMU-DRQ-63

MOPS: No

Safety Requirement: No

Rationale: Required to define the Software Part Number to identify the Application Software

Verification Method: Testing

The CMU Application Software shall define a 16-character string at a prefixed location in memory (0x0803FFEC) to identify the Application Software Part Number.

Format: < H108E-658><space><-><x><.><yy><space>

where,

x is major release number

yy is minor release number.

## 16.9 Resource Utilization

This section specifies the Software High Level Requirements for the resource utilization of the CMU Module.

### 16.9.1 Memory Usage

Requirement ID: H698-SRS-CMU-DRQ-70

MOPS: No

Safety Requirement: No

Rationale: This margin is required for future enhancements

Verification Method: Analysis

The CMU Module software shall not utilise more than 80% of the available FLASH and RAM memory for the initial certification of the product.

### 16.9.2 Throughput Usage

Requirement ID: H698-SRS-CMU-DRQ-71

MOPS: No

Safety Requirement: No

Rationale: This margin is required for future enhancements

Verification Method: Analysis

The CMU Module software shall not utilise more than 80% of the available processor throughput for the initial certification of the product.