Software Low Level Requirements for Analog Module of Engine Data Acquisition Unit of Airbus Helicopter Generic Vehicle Monitoring System and CMU+

Document Number: H398-003-011-ANA

Version No: 2.13

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: | Afreen P(Software Engineer, ALTEN Global Technologies Private Limited) |
| Reviewed By: | Sruthi D (Lead Engineer, ALTEN Global Technologies Private Limited)  C:\Users\afreen.p.SWSYS\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\A65C1828.tmp  Pravalika K (Quality Engineer – Quality Assurance, ALTEN Global Technologies Private Limited) |
| Approved By: | Devarani R (Manager - Quality Assurance and Management, ALTEN Global Technologies Private Limited) |
| Released By: | Madhuchandra 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 Pvt Ltd.**

**Table of Contents**

[1 Amendment Record 55](#_Toc210645476)

[2 Objective 80](#_Toc210645477)

[3 Scope 80](#_Toc210645478)

[4 References 80](#_Toc210645479)

[5 Acronyms and Definitions 81](#_Toc210645480)

[6 Document Control 82](#_Toc210645481)

[7 Responsibilities 82](#_Toc210645482)

[8 Distributions 83](#_Toc210645483)

[9 Traceability 83](#_Toc210645484)

[10 Software Low Level Requirements- Analog Application Software 83](#_Toc210645485)

[10.1 dauanaa825 83](#_Toc210645486)

[10.1.1 RxQueueRemove 84](#_Toc210645487)

[10.1.1.1 Brief Description 84](#_Toc210645488)

[10.1.1.2 List of HLRs allocated 84](#_Toc210645489)

[10.1.1.3 List of global variables accessed and modified 84](#_Toc210645490)

[10.1.1.4 Parameter list (Input/Output) 84](#_Toc210645491)

[10.1.1.5 Return Value 84](#_Toc210645492)

[10.1.1.6 Other CSUs called by this CSU 85](#_Toc210645493)

[10.1.1.7 Description of list of LLRs allocated 85](#_Toc210645494)

[10.1.1.7.1 dauanaa825-RxQueueRemove-LLR-001 85](#_Toc210645495)

[10.1.1.7.2 dauanaa825-RxQueueRemove-LLR-002 85](#_Toc210645496)

[10.1.1.7.3 dauanaa825-RxQueueRemove-LLR-003 85](#_Toc210645497)

[10.1.1.7.4 dauanaa825-RxQueueRemove-LLR-004 85](#_Toc210645498)

[10.1.1.7.5 dauanaa825-RxQueueRemove-LLR-005 86](#_Toc210645499)

[10.1.1.7.6 dauanaa825-RxQueueRemove-LLR-006 86](#_Toc210645500)

[10.1.1.7.7 dauanaa825-RxQueueRemove-LLR-007 86](#_Toc210645501)

[10.1.1.7.8 dauanaa825-RxQueueRemove-LLR-008 86](#_Toc210645502)

[10.1.1.7.9 dauanaa825-RxQueueRemove-LLR-009 87](#_Toc210645503)

[10.1.1.7.10 dauanaa825-RxQueueRemove-LLR-010 87](#_Toc210645504)

[10.1.1.7.11 dauanaa825-RxQueueRemove-LLR-011 87](#_Toc210645505)

[10.1.1.7.12 dauanaa825-RxQueueRemove-LLR-012 88](#_Toc210645506)

[10.1.1.7.13 dauanaa825-RxQueueRemove-LLR-013 88](#_Toc210645507)

[10.1.1.7.14 dauanaa825-RxQueueRemove-LLR-014 88](#_Toc210645508)

[10.1.1.7.15 dauanaa825-RxQueueRemove-LLR-015 88](#_Toc210645509)

[10.1.2 A825Receive 88](#_Toc210645510)

[10.1.2.1 Brief Description 88](#_Toc210645511)

[10.1.2.2 List of HLRs allocated 89](#_Toc210645512)

[10.1.2.3 List of global variables accessed and modified 89](#_Toc210645513)

[10.1.2.4 Parameter list (Input/Output) 89](#_Toc210645514)

[10.1.2.5 Return Value 89](#_Toc210645515)

[10.1.2.6 Other CSUs called by this CSU 89](#_Toc210645516)

[10.1.2.7 Description of list of LLRs allocated 90](#_Toc210645517)

[10.1.2.7.1 dauanaa825-A825Receive-LLR-001 90](#_Toc210645518)

[10.1.2.7.2 dauanaa825-A825Receive-LLR-002 90](#_Toc210645519)

[10.1.2.7.3 dauanaa825-A825Receive-LLR-003 90](#_Toc210645520)

[10.1.3 RxQueueInsert 90](#_Toc210645521)

[10.1.3.1 Brief Description 90](#_Toc210645522)

[10.1.3.2 List of HLRs allocated 91](#_Toc210645523)

[10.1.3.3 List of global variables accessed and modified 91](#_Toc210645524)

[10.1.3.4 Parameter list (Input/Output) 91](#_Toc210645525)

[10.1.3.5 Return Value 91](#_Toc210645526)

[10.1.3.6 Other CSUs called by this CSU 91](#_Toc210645527)

[10.1.3.7 Description of list of LLRs allocated 91](#_Toc210645528)

[10.1.3.7.1 dauanaa825-RxQueueInsert-LLR-001 92](#_Toc210645529)

[10.1.3.7.2 dauanaa825-RxQueueInsert-LLR-002 92](#_Toc210645530)

[10.1.3.7.3 dauanaa825-RxQueueInsert-LLR-003 92](#_Toc210645531)

[10.1.3.7.4 dauanaa825-RxQueueInsert-LLR-004 92](#_Toc210645532)

[10.1.4 ReceiveIntr 92](#_Toc210645533)

[10.1.4.1 Brief Description 93](#_Toc210645534)

[10.1.4.2 List of HLRs allocated 93](#_Toc210645535)

[10.1.4.3 List of global variables accessed and modified 93](#_Toc210645536)

[10.1.4.4 Parameter list (Input/Output) 93](#_Toc210645537)

[10.1.4.5 Return Value 93](#_Toc210645538)

[10.1.4.6 Other CSUs called by this CSU 93](#_Toc210645539)

[10.1.4.7 Description of list of LLRs allocated 94](#_Toc210645540)

[10.1.4.7.1 dauanaa825-ReceiveIntr-LLR-001 94](#_Toc210645541)

[10.1.4.7.2 dauanaa825-ReceiveIntr-LLR-002 94](#_Toc210645542)

[10.1.4.7.3 dauanaa825-ReceiveIntr-LLR-003 94](#_Toc210645543)

[10.1.4.7.4 dauanaa825-ReceiveIntr-LLR-004 94](#_Toc210645544)

[10.1.4.7.5 dauanaa825-ReceiveIntr-LLR-005 95](#_Toc210645545)

[10.1.4.7.6 dauanaa825-ReceiveIntr-LLR-006 95](#_Toc210645546)

[10.1.4.7.7 dauanaa825-ReceiveIntr-LLR-007 95](#_Toc210645547)

[10.1.4.7.8 dauanaa825-ReceiveIntr-LLR-008 95](#_Toc210645548)

[10.1.5 TxFromQueue 95](#_Toc210645549)

[10.1.5.1 Brief Description 96](#_Toc210645550)

[10.1.5.2 List of HLRs allocated 96](#_Toc210645551)

[10.1.5.3 List of global variables accessed and modified 96](#_Toc210645552)

[10.1.5.4 Parameter list (Input/Output) 96](#_Toc210645553)

[10.1.5.5 Return Value 96](#_Toc210645554)

[10.1.5.6 Other CSUs called by this CSU 96](#_Toc210645555)

[10.1.5.7 Description of list of LLRs allocated 97](#_Toc210645556)

[10.1.5.7.1 dauanaa825-TxFromQueue-LLR-001 97](#_Toc210645557)

[10.1.5.7.2 dauanaa825-TxFromQueue-LLR-002 97](#_Toc210645558)

[10.1.5.7.3 dauanaa825-TxFromQueue-LLR-003 97](#_Toc210645559)

[10.1.5.7.4 dauanaa825-TxFromQueue-LLR-004 97](#_Toc210645560)

[10.1.5.7.5 dauanaa825-TxFromQueue-LLR-005 98](#_Toc210645561)

[10.1.6 TransmitIntr 98](#_Toc210645562)

[10.1.6.1 Brief Description 98](#_Toc210645563)

[10.1.6.2 List of HLRs allocated 98](#_Toc210645564)

[10.1.6.3 List of global variables accessed and modified 98](#_Toc210645565)

[10.1.6.4 Parameter list (Input/Output) 98](#_Toc210645566)

[10.1.6.5 Return Value 99](#_Toc210645567)

[10.1.6.6 Other CSUs called by this CSU 99](#_Toc210645568)

[10.1.6.7 Description of list of LLRs allocated 99](#_Toc210645569)

[10.1.6.7.1 dauanaa825-TransmitIntr-LLR-001 99](#_Toc210645570)

[10.1.6.7.2 dauanaa825-TransmitIntr-LLR-002 99](#_Toc210645571)

[10.1.7 TxQueueInsert 99](#_Toc210645572)

[10.1.7.1 Brief Description 100](#_Toc210645573)

[10.1.7.2 List of HLRs allocated 100](#_Toc210645574)

[10.1.7.3 List of global variables accessed and modified 100](#_Toc210645575)

[10.1.7.4 Parameter list (Input/Output) 100](#_Toc210645576)

[10.1.7.5 Return Value 100](#_Toc210645577)

[10.1.7.6 Other CSUs called by this CSU 100](#_Toc210645578)

[10.1.7.7 Description of list of LLRs allocated 101](#_Toc210645579)

[10.1.7.7.1 dauanaa825-TxQueueInsert-LLR-001 101](#_Toc210645580)

[10.1.7.7.2 dauanaa825-TxQueueInsert-LLR-002 101](#_Toc210645581)

[10.1.7.7.3 dauanaa825-TxQueueInsert-LLR-003 101](#_Toc210645582)

[10.1.7.7.4 dauanaa825-TxQueueInsert-LLR-004 102](#_Toc210645583)

[10.1.8 BuildBroadcastExtId 102](#_Toc210645584)

[10.1.8.1 Brief Description 102](#_Toc210645585)

[10.1.8.2 List of HLRs allocated 102](#_Toc210645586)

[10.1.8.3 List of global variables accessed and modified 102](#_Toc210645587)

[10.1.8.4 Parameter list (Input/Output) 102](#_Toc210645588)

[10.1.8.5 Return Value 103](#_Toc210645589)

[10.1.8.6 Other CSUs called by this CSU 103](#_Toc210645590)

[10.1.8.7 Description of list of LLRs allocated 103](#_Toc210645591)

[10.1.8.7.1 dauanaa825-BuildBroadcastExtId-LLR-001 103](#_Toc210645592)

[10.1.8.7.2 dauanaa825-BuildBroadcastExtId-LLR-002 104](#_Toc210645593)

[10.1.9 BuildPeerToPeerExtId 104](#_Toc210645594)

[10.1.9.1 Brief Description 104](#_Toc210645595)

[10.1.9.2 List of HLRs allocated 104](#_Toc210645596)

[10.1.9.3 List of global variables accessed and modified 104](#_Toc210645597)

[10.1.9.4 Parameter list (Input/Output) 104](#_Toc210645598)

[10.1.9.5 Return Value 104](#_Toc210645599)

[10.1.9.6 Other CSUs called by this CSU 105](#_Toc210645600)

[10.1.9.7 Description of list of LLRs allocated 105](#_Toc210645601)

[10.1.9.7.1 dauanaa825-BuildPeerToPeerExtId-LLR-001 105](#_Toc210645602)

[10.1.9.7.2 dauanaa825-BuildPeerToPeerExtId-LLR-002 105](#_Toc210645603)

[10.1.10 A825Transmit 106](#_Toc210645604)

[10.1.10.1 Brief Description 106](#_Toc210645605)

[10.1.10.2 List of HLRs allocated 106](#_Toc210645606)

[10.1.10.3 List of global variables accessed and modified 106](#_Toc210645607)

[10.1.10.4 Parameter list (Input/Output) 106](#_Toc210645608)

[10.1.10.5 Return Value 106](#_Toc210645609)

[10.1.10.6 Other CSUs called by this CSU 107](#_Toc210645610)

[10.1.10.7 Description of list of LLRs allocated 107](#_Toc210645611)

[10.1.10.7.1 dauanaa825-A825Transmit-LLR-001 107](#_Toc210645612)

[10.1.10.7.2 dauanaa825-A825Transmit-LLR-002 107](#_Toc210645613)

[10.1.10.7.3 dauanaa825-A825Transmit-LLR-003 107](#_Toc210645614)

[10.1.10.7.4 dauanaa825-A825Transmit-LLR-004 108](#_Toc210645615)

[10.1.10.7.5 dauanaa825-A825Transmit-LLR-005 108](#_Toc210645616)

[10.1.10.7.6 dauanaa825-A825Transmit-LLR-006 108](#_Toc210645617)

[10.1.10.7.7 dauanaa825-A825Transmit-LLR-007 108](#_Toc210645618)

[10.1.11 InitCAN1 108](#_Toc210645619)

[10.1.11.1 Brief Description 109](#_Toc210645620)

[10.1.11.2 List of HLRs allocated 109](#_Toc210645621)

[10.1.11.3 List of global variables accessed and modified 109](#_Toc210645622)

[10.1.11.4 Parameter list (Input/Output) 109](#_Toc210645623)

[10.1.11.5 Return Value 109](#_Toc210645624)

[10.1.11.6 Other CSUs called by this CSU 109](#_Toc210645625)

[10.1.11.7 Description of list of LLRs allocated 110](#_Toc210645626)

[10.1.11.7.1 dauanaa825-InitCAN1-LLR-001 110](#_Toc210645627)

[10.1.11.7.2 dauanaa825-InitCAN1-LLR-002 110](#_Toc210645628)

[10.1.11.7.3 dauanaa825-InitCAN1-LLR-003 110](#_Toc210645629)

[10.1.11.7.4 dauanaa825-InitCAN1-LLR-004 110](#_Toc210645630)

[10.1.11.7.5 dauanaa825-InitCAN1-LLR-005 111](#_Toc210645631)

[10.1.11.7.6 dauanaa825-InitCAN1-LLR-006 111](#_Toc210645632)

[10.1.11.7.7 dauanaa825-InitCAN1-LLR-007 111](#_Toc210645633)

[10.1.11.7.8 dauanaa825-InitCAN1-LLR-008 111](#_Toc210645634)

[10.1.11.7.9 dauanaa825-InitCAN1-LLR-009 112](#_Toc210645635)

[10.1.11.7.10 dauanaa825-InitCAN1-LLR-014 112](#_Toc210645636)

[10.1.12 NvicConfig 112](#_Toc210645637)

[10.1.12.1 Brief Description 112](#_Toc210645638)

[10.1.12.2 List of HLRs allocated 112](#_Toc210645639)

[10.1.12.3 List of global variables accessed and modified 113](#_Toc210645640)

[10.1.12.4 Parameter list (Input/Output) 113](#_Toc210645641)

[10.1.12.5 Return Value 113](#_Toc210645642)

[10.1.12.6 Other CSUs called by this CSU 113](#_Toc210645643)

[10.1.12.7 Description of list of LLRs allocated 113](#_Toc210645644)

[10.1.12.7.1 dauanaa825-NvicConfig-LLR-001 114](#_Toc210645645)

[10.1.12.7.2 dauanaa825-NvicConfig-LLR-002 114](#_Toc210645646)

[10.1.12.7.3 dauanaa825-NvicConfig-LLR-003 114](#_Toc210645647)

[10.1.12.7.4 dauanaa825-NvicConfig-LLR-004 114](#_Toc210645648)

[10.1.13 A825Init 114](#_Toc210645649)

[10.1.13.1 Brief Description 115](#_Toc210645650)

[10.1.13.2 List of HLRs allocated 115](#_Toc210645651)

[10.1.13.3 List of global variables accessed and modified 115](#_Toc210645652)

[10.1.13.4 Parameter list (Input/Output) 115](#_Toc210645653)

[10.1.13.5 Return Value 115](#_Toc210645654)

[10.1.13.6 Other CSUs called by this CSU 115](#_Toc210645655)

[10.1.13.7 Description of list of LLRs allocated 116](#_Toc210645656)

[10.1.13.7.1 dauanaa825-A825Init-LLR-001 116](#_Toc210645657)

[10.1.13.7.2 dauanaa825-A825Init-LLR-002 116](#_Toc210645658)

[10.2 dauanaa825comm 116](#_Toc210645659)

[10.2.1 A825CommTask 116](#_Toc210645660)

[10.2.1.1 Brief Description 116](#_Toc210645661)

[10.2.1.2 List of HLRs allocated 117](#_Toc210645662)

[10.2.1.3 List of global variables accessed and modified 117](#_Toc210645663)

[10.2.1.4 Parameter list (Input/Output) 117](#_Toc210645664)

[10.2.1.5 Return Value 117](#_Toc210645665)

[10.2.1.6 Other CSUs called by this CSU 117](#_Toc210645666)

[10.2.1.7 Description of list of LLRs allocated 117](#_Toc210645667)

[10.2.1.7.1 dauanaa825comm-A825CommTask-LLR-001 118](#_Toc210645668)

[10.2.1.7.2 dauanaa825comm-A825CommTask-LLR-002 118](#_Toc210645669)

[10.2.1.7.3 dauanaa825comm-A825CommTask-LLR-003 118](#_Toc210645670)

[10.2.1.7.4 dauanaa825comm-A825CommTask-LLR-004 118](#_Toc210645671)

[10.2.1.7.5 dauanaa825comm-A825CommTask-LLR-005 119](#_Toc210645672)

[10.2.1.7.6 dauanaa825comm-A825CommTask-LLR-006 119](#_Toc210645673)

[10.2.1.7.7 dauanaa825comm-A825CommTask-LLR-007 119](#_Toc210645674)

[10.2.2 A825CommInit 119](#_Toc210645675)

[10.2.2.1 Brief Description 120](#_Toc210645676)

[10.2.2.2 List of HLRs allocated 120](#_Toc210645677)

[10.2.2.3 List of global variables accessed and modified 120](#_Toc210645678)

[10.2.2.4 Parameter list (Input/Output) 120](#_Toc210645679)

[10.2.2.5 Return Value 120](#_Toc210645680)

[10.2.2.6 Other CSUs called by this CSU 120](#_Toc210645681)

[10.2.2.7 Description of list of LLRs allocated 121](#_Toc210645682)

[10.2.2.7.1 dauanaa825comm-A825CommInit-LLR-001 121](#_Toc210645683)

[10.2.2.7.2 dauanaa825comm-A825CommInit-LLR-003 121](#_Toc210645684)

[10.2.2.7.3 dauanaa825comm-A825CommInit-LLR-002 121](#_Toc210645685)

[10.3 dauanaapp 122](#_Toc210645686)

[10.3.1 AppGetA825RCI 122](#_Toc210645687)

[10.3.1.1 Brief Description 122](#_Toc210645688)

[10.3.1.2 List of HLRs allocated 122](#_Toc210645689)

[10.3.1.3 List of global variables accessed and modified 122](#_Toc210645690)

[10.3.1.4 Parameter list (Input/Output) 122](#_Toc210645691)

[10.3.1.5 Return Value 123](#_Toc210645692)

[10.3.1.6 Other CSUs called by this CSU 123](#_Toc210645693)

[10.3.1.7 Description of list of LLRs allocated 123](#_Toc210645694)

[10.3.1.7.1 dauanaapp-AppGetA825RCI-LLR-001 123](#_Toc210645695)

[10.3.2 ReadChannelBIT 123](#_Toc210645696)

[10.3.2.1 Brief Description 123](#_Toc210645697)

[10.3.2.2 List of HLRs allocated 123](#_Toc210645698)

[10.3.2.3 List of global variables accessed and modified 124](#_Toc210645699)

[10.3.2.4 Parameter list (Input/Output) 124](#_Toc210645700)

[10.3.2.5 Return Value 124](#_Toc210645701)

[10.3.2.6 Other CSUs called by this CSU 124](#_Toc210645702)

[10.3.2.7 Description of list of LLRs allocated 124](#_Toc210645703)

[10.3.2.7.1 dauanaapp-ReadChannelBIT-LLR-001 124](#_Toc210645704)

[10.3.2.7.2 dauanaapp-ReadChannelBIT-LLR-002 125](#_Toc210645705)

[10.3.2.7.3 dauanaapp-ReadChannelBIT-LLR-003 125](#_Toc210645706)

[10.3.3 ScaleChannel 125](#_Toc210645707)

[10.3.3.1 Brief Description 125](#_Toc210645708)

[10.3.3.2 List of HLRs allocated 125](#_Toc210645709)

[10.3.3.3 List of global variables accessed and modified 126](#_Toc210645710)

[10.3.3.4 Parameter list (Input/Output) 126](#_Toc210645711)

[10.3.3.5 Return Value 126](#_Toc210645712)

[10.3.3.6 Other CSUs called by this CSU 126](#_Toc210645713)

[10.3.3.7 Description of list of LLRs allocated 126](#_Toc210645714)

[10.3.3.7.1 dauanaapp-ScaleChannel-LLR-001 127](#_Toc210645715)

[10.3.3.7.2 dauanaapp-ScaleChannel-LLR-002 127](#_Toc210645716)

[10.3.3.7.3 dauanaapp-ScaleChannel-LLR-003 127](#_Toc210645717)

[10.3.3.7.4 dauanaapp-ScaleChannel-LLR-004 127](#_Toc210645718)

[10.3.3.7.5 dauanaapp-ScaleChannel-LLR-011 128](#_Toc210645719)

[10.3.3.7.6 dauanaapp-ScaleChannel-LLR-012 128](#_Toc210645720)

[10.3.3.7.7 dauanaapp-ScaleChannel-LLR-013 128](#_Toc210645721)

[10.3.3.7.8 dauanaapp-ScaleChannel-LLR-014 128](#_Toc210645722)

[10.3.3.7.9 dauanaapp-ScaleChannel-LLR-015 129](#_Toc210645723)

[10.3.3.7.10 dauanaapp-ScaleChannel-LLR-005 129](#_Toc210645724)

[10.3.3.7.11 dauaanapp-ScaleChannel-LLR-016 129](#_Toc210645725)

[10.3.3.7.12 dauanaapp-ScaleChannel-LLR-017 130](#_Toc210645726)

[10.3.3.7.11 dauanaapp-ScaleChannel-LLR-006 130](#_Toc210645727)

[10.3.3.7.12 dauanaapp-ScaleChannel-LLR-007 130](#_Toc210645728)

[10.3.3.7.13 dauanaapp-ScaleChannel-LLR-008 131](#_Toc210645729)

[10.3.3.7.14 dauanaapp-ScaleChannel-LLR-009 131](#_Toc210645730)

[10.3.3.7.15 dauanaapp-ScaleChannel-LLR-010 131](#_Toc210645731)

[10.3.4 ScaleColdJunction 132](#_Toc210645732)

[10.3.4.1 Brief Description 132](#_Toc210645733)

[10.3.4.2 List of HLRs allocated 132](#_Toc210645734)

[10.3.4.3 List of global variables accessed and modified 132](#_Toc210645735)

[10.3.4.4 Parameter list (Input/Output) 132](#_Toc210645736)

[10.3.4.5 Return Value 132](#_Toc210645737)

[10.3.4.6 Other CSUs called by this CSU 133](#_Toc210645738)

[10.3.4.7 Description of list of LLRs allocated 133](#_Toc210645739)

[10.3.4.7.1 dauanaapp-ScaleColdJunction-LLR-001 133](#_Toc210645740)

[10.3.4.7.2 dauanaapp-ScaleColdJunction-LLR-002 133](#_Toc210645741)

[10.3.4.7.3 dauanaapp-ScaleColdJunction-LLR-003 133](#_Toc210645742)

[10.3.4.7.4 dauanaapp-ScaleColdJunction-LLR-004 134](#_Toc210645743)

[10.3.4.7.5 dauanaapp-ScaleColdJunction-LLR-005 134](#_Toc210645744)

[10.3.4.7.6 dauanaapp-ScaleColdJunction-LLR-006 134](#_Toc210645745)

[10.3.4.7.7 dauanaapp-ScaleColdJunction-LLR-007 135](#_Toc210645746)

[10.3.5 SendReadings 135](#_Toc210645747)

[10.3.5.1 Brief Description 135](#_Toc210645748)

[10.3.5.2 List of HLRs allocated 135](#_Toc210645749)

[10.3.5.3 List of global variables accessed and modified 135](#_Toc210645750)

[10.3.5.4 Parameter list (Input/Output) 135](#_Toc210645751)

[10.3.5.5 Return Value 136](#_Toc210645752)

[10.3.5.6 Other CSUs called by this CSU 136](#_Toc210645753)

[10.3.5.7.1 dauanaapp-SendReadings-LLR-002 136](#_Toc210645754)

[10.3.5.7.2 dauanaapp-SendReadings-LLR-003 136](#_Toc210645755)

[10.3.6 SendCrcPn 137](#_Toc210645756)

[10.3.6.1 Brief Description 137](#_Toc210645757)

[10.3.6.2 List of HLRs allocated 137](#_Toc210645758)

[10.3.6.3 List of global variables accessed and modified 137](#_Toc210645759)

[10.3.6.4 Parameter list (Input/Output) 137](#_Toc210645760)

[10.3.6.5 Return Value 137](#_Toc210645761)

[10.3.6.6 Other CSUs called by this CSU 138](#_Toc210645762)

[10.3.6.7 Description of list of LLRs allocated 138](#_Toc210645763)

[10.3.6.7.1 dauanaapp-SendCrcPn-LLR-001 138](#_Toc210645764)

[10.3.7 ScaleReadings 140](#_Toc210645765)

[10.3.7.1 Brief Description 140](#_Toc210645766)

[10.3.7.2 List of HLRs allocated 140](#_Toc210645767)

[10.3.7.3 List of global variables accessed and modified 140](#_Toc210645768)

[10.3.7.4 Parameter list (Input/Output) 141](#_Toc210645769)

[10.3.7.5 Return Value 141](#_Toc210645770)

[10.3.7.6 Other CSUs called by this CSU 141](#_Toc210645771)

[10.3.7.7 Description of list of LLRs allocated 141](#_Toc210645772)

[10.3.7.7.1 dauanaapp-ScaleReadings-LLR-001 141](#_Toc210645773)

[10.3.7.7.2 dauanaapp-ScaleReadings-LLR-002 142](#_Toc210645774)

[10.3.7.7.3 dauanaapp-ScaleReadings-LLR-003 142](#_Toc210645775)

[10.3.7.7.4 dauanaapp-ScaleReadings-LLR-004 143](#_Toc210645776)

[10.3.7.7.5 dauanaapp-ScaleReadings-LLR-005 144](#_Toc210645777)

[10.3.8 AppMessageNOC 145](#_Toc210645778)

[10.3.8.1 Brief Description 145](#_Toc210645779)

[10.3.8.2 List of HLRs allocated 145](#_Toc210645780)

[10.3.8.3 List of global variables accessed and modified 145](#_Toc210645781)

[10.3.8.4 Parameter list (Input/Output) 145](#_Toc210645782)

[10.3.8.5 Return Value 145](#_Toc210645783)

[10.3.8.6 Other CSUs called by this CSU 146](#_Toc210645784)

[10.3.8.7 Description of list of LLRs allocated 146](#_Toc210645785)

[10.3.8.7.1 dauanaapp-AppMessageNOC-LLR-001 146](#_Toc210645786)

[10.3.8.7.2 dauanaapp-AppMessageNOC-LLR-002 146](#_Toc210645787)

[10.3.8.7.3 dauanaapp-AppMessageNOC-LLR-004 146](#_Toc210645788)

[10.3.8.7.4 dauanaapp-AppMessageNOC-LLR-005 146](#_Toc210645789)

[10.3.9 AppMessageNSC 147](#_Toc210645790)

[10.3.9.1 Brief Description 147](#_Toc210645791)

[10.3.9.2 List of HLRs allocated 147](#_Toc210645792)

[10.3.9.3 List of global variables accessed and modified 147](#_Toc210645793)

[10.3.9.4 Parameter list (Input/Output) 147](#_Toc210645794)

[10.3.9.5 Return Value 147](#_Toc210645795)

[10.3.9.6 Other CSUs called by this CSU 148](#_Toc210645796)

[10.3.9.7 Description of list of LLRs allocated 148](#_Toc210645797)

[10.3.9.7.1 dauanaapp-AppMessageNSC-LLR-001 148](#_Toc210645798)

[10.3.9.7.2 dauanaapp-AppMessageNSC-LLR-002 148](#_Toc210645799)

[10.3.9.7.3 dauanaapp-AppMessageNSC-LLR-003 148](#_Toc210645800)

[10.3.9.7.4 dauanaapp-AppMessageNSC-LLR-004 148](#_Toc210645801)

[10.3.9.7.5 dauanaapp-AppMessageNSC-LLR-006 149](#_Toc210645802)

[10.3.9.7.6 dauanaapp-AppMessageNSC-LLR-007 149](#_Toc210645803)

[10.3.9.7.7 dauanaapp-AppMessageNSC-LLR-008 149](#_Toc210645804)

[10.3.9.7.8 dauanaapp-AppMessageNSC-LLR-009 149](#_Toc210645805)

[10.3.10 AppTask 150](#_Toc210645806)

[10.3.10.1 Brief Description 150](#_Toc210645807)

[10.3.10.2 List of HLRs allocated 150](#_Toc210645808)

[10.3.10.3 List of global variables accessed and modified 150](#_Toc210645809)

[10.3.10.4 Parameter list (Input/Output) 150](#_Toc210645810)

[10.3.10.5 Return Value 150](#_Toc210645811)

[10.3.10.6 Other CSUs called by this CSU 151](#_Toc210645812)

[10.3.10.7 Description of list of LLRs allocated 151](#_Toc210645813)

[10.3.10.7.1 dauanaapp-AppTask-LLR-001 151](#_Toc210645814)

[10.3.11 APPInit 151](#_Toc210645815)

[10.3.11.1 Brief Description 151](#_Toc210645816)

[10.3.11.2 List of HLRs allocated 151](#_Toc210645817)

[10.3.11.3 List of global variables accessed and modified 152](#_Toc210645818)

[10.3.11.4 Parameter list (Input/Output) 152](#_Toc210645819)

[10.3.11.5 Return Value 152](#_Toc210645820)

[10.3.11.6 Other CSUs called by this CSU 152](#_Toc210645821)

[10.3.11.7 Description of list of LLRs allocated 152](#_Toc210645822)

[10.3.11.7.1 dauanaapp-APPInit-LLR-001 152](#_Toc210645823)

[10.3.11.7.2 dauanaapp-APPInit-LLR-002 153](#_Toc210645824)

[10.3.11.7.3 dauanaapp-APPInit-LLR-003 153](#_Toc210645825)

[10.3.12 ScaleTach 153](#_Toc210645826)

[10.3.12.1 Brief Description 153](#_Toc210645827)

[10.3.12.2 List of HLRs allocated 153](#_Toc210645828)

[10.3.12.3 List of global variables accessed and modified 154](#_Toc210645829)

[10.3.12.4 Parameter list (Input/Output) 154](#_Toc210645830)

[10.3.12.5 Return Value 154](#_Toc210645831)

[10.3.12.6 Other CSUs called by this CSU 154](#_Toc210645832)

[10.3.12.7 Description of list of LLRs allocated 154](#_Toc210645833)

[10.3.12.7.1 dauanaapp-ScaleTach-LLR-001 155](#_Toc210645834)

[10.3.12.7.2 dauanaapp-ScaleTach-LLR-002 155](#_Toc210645835)

[10.3.12.7.3 dauanaapp-ScaleTach-LLR-003 155](#_Toc210645836)

[10.3.12.7.4 dauanaapp-ScaleTach-LLR-004 155](#_Toc210645837)

[10.3.12.7.5 dauanaapp-ScaleTach-LLR-005 155](#_Toc210645838)

[10.3.12.7.6 dauanaapp-ScaleTach-LLR-006 156](#_Toc210645839)

[10.3.12.7.7 dauanaapp-ScaleTach-LLR-007 156](#_Toc210645840)

[10.3.12.7.8 dauanaapp-ScaleTach-LLR-008 156](#_Toc210645841)

[10.3.12.7.9 dauanaapp-ScaleTach-LLR-009 157](#_Toc210645842)

[10.3.12.7.10 dauanaapp-ScaleTach-LLR-010 157](#_Toc210645843)

[10.3.13 Scalepwmc 157](#_Toc210645844)

[10.3.13.1 Brief Description 157](#_Toc210645845)

[10.3.13.2 List of HLRs allocated 158](#_Toc210645846)

[10.3.13.3 List of global variables accessed and modified 158](#_Toc210645847)

[10.3.13.4 Parameter list (Input/Output) 158](#_Toc210645848)

[10.3.13.5 Return Value 158](#_Toc210645849)

[10.3.13.6 Other CSUs called by this CSU 158](#_Toc210645850)

[10.3.13.7 Description of list of LLRs allocated 158](#_Toc210645851)

[10.3.13.7.1 dauanaapp-ScalePwmc-LLR-001 159](#_Toc210645852)

[10.3.13.7.2 dauanaapp-ScalePwmc-LLR-002 159](#_Toc210645853)

[10.3.13.7.3 dauanaapp-ScalePwmc-LLR-003 159](#_Toc210645854)

[10.3.13.7.4 dauanaapp-ScalePwmc-LLR-004 159](#_Toc210645855)

[10.3.13.7.5 dauanaapp-ScalePwmc-LLR-005 160](#_Toc210645856)

[10.3.13.7.6 dauanaapp-ScalePwmc-LLR-006 160](#_Toc210645857)

[10.4 dauanacbit 160](#_Toc210645858)

[10.4.1 StackTest 160](#_Toc210645859)

[10.4.1.1 Brief Description 160](#_Toc210645860)

[10.4.1.2 List of HLRs allocated 160](#_Toc210645861)

[10.4.1.3 List of global variables accessed and modified 161](#_Toc210645862)

[10.4.1.4 Parameter list (Input/Output) 161](#_Toc210645863)

[10.4.1.5 Return Value 161](#_Toc210645864)

[10.4.1.6 Other CSUs called by this CSU 161](#_Toc210645865)

[10.4.1.7 Description of list of LLRs allocated 161](#_Toc210645866)

[10.4.1.7.1 dauanacbit-StackTest-LLR-001 161](#_Toc210645867)

[10.4.2 ErrorHandler 162](#_Toc210645868)

[10.4.2.1 Brief Description 162](#_Toc210645869)

[10.4.2.2 List of HLRs allocated 162](#_Toc210645870)

[10.4.2.3 List of global variables accessed and modified 162](#_Toc210645871)

[10.4.2.4 Parameter list (Input/Output) 162](#_Toc210645872)

[10.4.2.5 Return Value 163](#_Toc210645873)

[10.4.2.6 Other CSUs called by this CSU 163](#_Toc210645874)

[10.4.2.7 Description of list of LLRs allocated 163](#_Toc210645875)

[10.4.2.7.1 dauanacbit-ErrorHandler-LLR-001 163](#_Toc210645876)

[10.4.2.7.2 dauanacbit-ErrorHandler-LLR-002 163](#_Toc210645877)

[10.4.3 CbitTask 163](#_Toc210645878)

[10.4.3.1 Brief Description 163](#_Toc210645879)

[10.4.3.2 List of HLRs allocated 164](#_Toc210645880)

[10.4.3.3 List of global variables accessed and modified 164](#_Toc210645881)

[10.4.3.4 Parameter list (Input/Output) 164](#_Toc210645882)

[10.4.3.5 Return Value 164](#_Toc210645883)

[10.4.3.6 Other CSUs called by this CSU 164](#_Toc210645884)

[10.4.3.7 Description of list of LLRs allocated 164](#_Toc210645885)

[10.4.3.7.1 dauanacbit-CbitTask-LLR-001 165](#_Toc210645886)

[10.4.4 CbitInit 165](#_Toc210645887)

[10.4.4.1 Brief Description 165](#_Toc210645888)

[10.4.4.2 List of HLRs allocated 165](#_Toc210645889)

[10.4.4.3 List of global variables accessed and modified 165](#_Toc210645890)

[10.4.4.4 Parameter list (Input/Output) 166](#_Toc210645891)

[10.4.4.5 Return Value 166](#_Toc210645892)

[10.4.4.6 Other CSUs called by this CSU 166](#_Toc210645893)

[10.4.4.7 Description of list of LLRs allocated 166](#_Toc210645894)

[10.4.4.7.1 dauanacbit-CbitInit-LLR-001 166](#_Toc210645895)

[10.4.4.7.2 dauanacbit-CbitInit-LLR-002 167](#_Toc210645896)

[10.4.4.7.3 dauanacbit-CbitInit-LLR-003 167](#_Toc210645897)

[10.5 dauanacputest 167](#_Toc210645898)

[10.5.1 AluTest 167](#_Toc210645899)

[10.5.1.1 Brief Description 167](#_Toc210645900)

[10.5.1.2 List of HLRs allocated 167](#_Toc210645901)

[10.5.1.3 List of global variables accessed and modified 168](#_Toc210645902)

[10.5.1.4 Parameter list (Input/Output) 168](#_Toc210645903)

[10.5.1.5 Return Value 168](#_Toc210645904)

[10.5.1.6 Other CSUs called by this CSU 168](#_Toc210645905)

[10.5.1.7 Description of list of LLRs allocated 168](#_Toc210645906)

[10.5.1.7.1 dauanacputest-AluTest-LLR-001 168](#_Toc210645907)

[10.5.1.7.2 dauanacputest-AluTest-LLR-002 169](#_Toc210645908)

[10.5.1.7.3 dauanacputest-AluTest-LLR-003 169](#_Toc210645909)

[10.5.1.7.4 dauanacputest-AluTest-LLR-004 169](#_Toc210645910)

[10.5.1.7.5 dauanacputest-AluTest-LLR-005 169](#_Toc210645911)

[10.5.1.7.6 dauanacputest-AluTest-LLR-006 169](#_Toc210645912)

[10.5.1.7.7 dauanacputest-AluTest-LLR-007 170](#_Toc210645913)

[10.5.1.7.8 dauanacputest-AluTest-LLR-008 170](#_Toc210645914)

[10.5.1.7.9 dauanacputest-AluTest-LLR-009 170](#_Toc210645915)

[10.5.1.7.10 dauanacputest-AluTest-LLR-010 170](#_Toc210645916)

[10.5.1.7.11 dauanacputest-AluTest-LLR-011 170](#_Toc210645917)

[10.6 dauanacrt0 171](#_Toc210645918)

[10.6.1 Crt0TransferData 171](#_Toc210645919)

[10.6.1.1 Brief Description 171](#_Toc210645920)

[10.6.1.2 List of HLRs allocated 171](#_Toc210645921)

[10.6.1.3 List of global variables accessed and modified 171](#_Toc210645922)

[10.6.1.4 Parameter list (Input/Output) 171](#_Toc210645923)

[10.6.1.5 Return Value 172](#_Toc210645924)

[10.6.1.6 Other CSUs called by this CSU 172](#_Toc210645925)

[10.6.1.7 Description of list of LLRs allocated 172](#_Toc210645926)

[10.6.1.7.1 dauanacrt0-Crt0TransferData-LLR-001 172](#_Toc210645927)

[10.6.1.7.2 dauanacrt0-Crt0TransferData-LLR-002 172](#_Toc210645928)

[10.6.1.7.3 dauanacrt0-Crt0TransferData-LLR-003 172](#_Toc210645929)

[10.7 dauanahw 173](#_Toc210645930)

[10.7.1 InitIO 173](#_Toc210645931)

[10.7.1.1 Brief Description 173](#_Toc210645932)

[10.7.1.2 List of HLRs allocated 173](#_Toc210645933)

[10.7.1.3 List of global variables accessed and modified 173](#_Toc210645934)

[10.7.1.4 Parameter list (Input/Output) 173](#_Toc210645935)

[10.7.1.5 Return Value 174](#_Toc210645936)

[10.7.1.6 Other CSUs called by this CSU 174](#_Toc210645937)

[10.7.1.7 Description of list of LLRs allocated 174](#_Toc210645938)

[10.7.1.7.1 dauanahw-InitIO-LLR-001 174](#_Toc210645939)

[10.7.1.7.2 dauanahw-InitIO-LLR-002 174](#_Toc210645940)

[10.7.1.7.3 dauanahw-InitIO-LLR-003 174](#_Toc210645941)

[10.7.1.7.4 dauanahw-InitIO-LLR-004 175](#_Toc210645942)

[10.7.1.7.5 dauanahw-InitIO-LLR-005 175](#_Toc210645943)

[10.7.1.7.6 dauanahw-InitIO-LLR-006 175](#_Toc210645944)

[10.7.1.7.7 dauanahw-InitIO-LLR-007 175](#_Toc210645945)

[10.7.1.7.8 dauanahw-InitIO-LLR-008 176](#_Toc210645946)

[10.7.1.7.9 dauanahw-InitIO-LLR-009 176](#_Toc210645947)

[10.7.1.7.10 dauanahw-InitIO-LLR-010 176](#_Toc210645948)

[10.7.1.7.11 dauanahw-InitIO-LLR-013 176](#_Toc210645949)

[10.7.1.7.12 dauanahw-InitIO-LLR-014 177](#_Toc210645950)

[10.7.1.7.13 dauanahw-InitIO-LLR-015 177](#_Toc210645951)

[10.7.1.7.14 dauanahw-InitIO-LLR-016 177](#_Toc210645952)

[10.7.1.7.15 dauanahw-InitIO-LLR-017 178](#_Toc210645953)

[10.7.1.7.16 dauanahw-InitIO-LLR-018 178](#_Toc210645954)

[10.7.1.7.17 dauanahw-InitIO-LLR-019 178](#_Toc210645955)

[10.7.1.7.18 dauanahw-InitIO-LLR-020 179](#_Toc210645956)

[10.7.2 HwInit 179](#_Toc210645957)

[10.7.2.1 Brief Description 179](#_Toc210645958)

[10.7.2.2 List of HLRs allocated 179](#_Toc210645959)

[10.7.2.3 List of global variables accessed and modified 179](#_Toc210645960)

[10.7.2.4 Parameter list (Input/Output) 179](#_Toc210645961)

[10.7.2.5 Return Value 180](#_Toc210645962)

[10.7.2.6 Other CSUs called by this CSU 180](#_Toc210645963)

[10.7.2.7 Description of list of LLRs allocated 180](#_Toc210645964)

[10.7.2.7.1 dauanahw-HwInit-LLR-001 181](#_Toc210645965)

[10.7.2.7.2 dauanahw-HwInit-LLR-002 181](#_Toc210645966)

[10.7.2.7.3 dauanahw-HwInit-LLR-003 181](#_Toc210645967)

[10.7.2.7.4 dauanahw-HwInit-LLR-004 181](#_Toc210645968)

[10.7.2.7.5 dauanahw-HwInit-LLR-005 181](#_Toc210645969)

[10.7.2.7.6 dauanahw-HwInit-LLR-006 181](#_Toc210645970)

[10.7.2.7.7 dauanahw-HwInit-LLR-007 182](#_Toc210645971)

[10.7.2.7.8 dauanahw-HwInit-LLR-008 182](#_Toc210645972)

[10.7.2.7.9 dauanahw-HwInit-LLR-009 182](#_Toc210645973)

[10.7.2.7.10 dauanahw-HwInit-LLR-010 182](#_Toc210645974)

[10.7.2.7.11 dauanahw-HwInit-LLR-011 182](#_Toc210645975)

[10.7.2.7.12 dauanahw-HwInit-LLR-012 182](#_Toc210645976)

[10.7.2.7.13 dauanahw-HwInit-LLR-013 183](#_Toc210645977)

[10.7.2.7.14 dauanahw-HwInit-LLR-014 183](#_Toc210645978)

[10.7.2.7.15 dauanahw-HwInit-LLR-015 183](#_Toc210645979)

[10.7.2.7.16 dauanahw-HwInit-LLR-016 183](#_Toc210645980)

[10.7.2.7.17 dauanahw-HwInit-LLR-017 183](#_Toc210645981)

[10.7.2.7.18 dauanahw-HwInit-LLR-018 184](#_Toc210645982)

[10.7.2.7.19 dauanahw-HwInit-LLR-019 184](#_Toc210645983)

[10.7.2.7.20 dauanahw-HwInit-LLR-020 184](#_Toc210645984)

[10.7.2.7.21 dauanahw-HwInit-LLR-021 184](#_Toc210645985)

[10.7.2.7.22 dauanahw-HwInit-LLR-022 184](#_Toc210645986)

[10.7.2.7.23 dauanahw-HwInit-LLR-023 184](#_Toc210645987)

[10.7.2.7.24 dauanahw-HwInit-LLR-024 185](#_Toc210645988)

[10.7.2.7.25 dauanahw-HwInit-LLR-025 185](#_Toc210645989)

[10.7.2.7.26 dauanahw-HwInit-LLR-026 185](#_Toc210645990)

[10.7.2.7.27 dauanahw-HwInit-LLR-027 185](#_Toc210645991)

[10.7.2.7.28 dauanahw-HwInit-LLR-029 185](#_Toc210645992)

[10.7.2.7.29 dauanahw-HwInit-LLR-030 185](#_Toc210645993)

[10.7.2.7.30 dauanahw-HwInit-LLR-028 186](#_Toc210645994)

[10.8 dauanaiadc 186](#_Toc210645995)

[10.8.1 IadcRead 186](#_Toc210645996)

[10.8.1.1 Brief Description 186](#_Toc210645997)

[10.8.1.2 List of HLRs allocated 186](#_Toc210645998)

[10.8.1.3 List of global variables accessed and modified 186](#_Toc210645999)

[10.8.1.4 Parameter list (Input/Output) 187](#_Toc210646000)

[10.8.1.5 Return Value 187](#_Toc210646001)

[10.8.1.6 Other CSUs called by this CSU 187](#_Toc210646002)

[10.8.1.7 Description of list of LLRs allocated 187](#_Toc210646003)

[10.8.1.7.1 dauanaiadc-IadcRead-LLR-001 187](#_Toc210646004)

[10.8.2 TransferCompleteIntr 187](#_Toc210646005)

[10.8.2.1 Brief Description 187](#_Toc210646006)

[10.8.2.2 List of HLRs allocated 188](#_Toc210646007)

[10.8.2.3 List of global variables accessed and modified 188](#_Toc210646008)

[10.8.2.4 Parameter list (Input/Output) 188](#_Toc210646009)

[10.8.2.5 Return Value 188](#_Toc210646010)

[10.8.2.6 Other CSUs called by this CSU 188](#_Toc210646011)

[10.8.2.7 Description of list of LLRs allocated 188](#_Toc210646012)

[10.8.2.7.1 dauanaiadc-TransferCompleteIntr-LLR-001 189](#_Toc210646013)

[10.8.2.7.2 dauanaiadc-TransferCompleteIntr-LLR-002 189](#_Toc210646014)

[10.8.3 ConfigureDma 189](#_Toc210646015)

[10.8.3.1 Brief Description 189](#_Toc210646016)

[10.8.3.2 List of HLRs allocated 189](#_Toc210646017)

[10.8.3.3 List of global variables accessed and modified 189](#_Toc210646018)

[10.8.3.4 Parameter list (Input/Output) 190](#_Toc210646019)

[10.8.3.5 Return Value 190](#_Toc210646020)

[10.8.3.6 Other CSUs called by this CSU 190](#_Toc210646021)

[10.8.3.7 Description of list of LLRs allocated 190](#_Toc210646022)

[10.8.3.7.2 dauanaiadc-ConfigureDma-LLR-002 190](#_Toc210646023)

[10.8.3.7.3 dauanaiadc-ConfigureDma-LLR-003 191](#_Toc210646024)

[10.8.3.7.4 dauanaiadc-ConfigureDma-LLR-004 191](#_Toc210646025)

[10.8.3.7.5 dauanaiadc-ConfigureDma-LLR-005 191](#_Toc210646026)

[10.8.3.7.6 dauanaiadc-ConfigureDma-LLR-006 192](#_Toc210646027)

[10.8.4 TriggerInternalAdc 192](#_Toc210646028)

[10.8.4.1 Brief Description 192](#_Toc210646029)

[10.8.4.2 List of HLRs allocated 192](#_Toc210646030)

[10.8.4.3 List of global variables accessed and modified 192](#_Toc210646031)

[10.8.4.4 Parameter list (Input/Output) 192](#_Toc210646032)

[10.8.4.5 Return Value 193](#_Toc210646033)

[10.8.4.6 Other CSUs called by this CSU 193](#_Toc210646034)

[10.8.4.7 Description of list of LLRs allocated 193](#_Toc210646035)

[10.8.4.7.1 dauanaiadc-TriggerInternalAdc-LLR-001 193](#_Toc210646036)

[10.8.4.7.2 dauanaiadc-TriggerInternalAdc-LLR-002 193](#_Toc210646037)

[10.8.5 IadcInit 193](#_Toc210646038)

[10.8.5.1 Brief Description 193](#_Toc210646039)

[10.8.5.2 List of HLRs allocated 194](#_Toc210646040)

[10.8.5.3 List of global variables accessed and modified 194](#_Toc210646041)

[10.8.5.4 Parameter list (Input/Output) 194](#_Toc210646042)

[10.8.5.5 Return Value 194](#_Toc210646043)

[10.8.5.6 Other CSUs called by this CSU 194](#_Toc210646044)

[10.8.5.7 Description of list of LLRs allocated 195](#_Toc210646045)

[10.8.5.7.1 dauanaiadc-IadcInit-LLR-001 195](#_Toc210646046)

[10.8.5.7.2 dauanaiadc-IadcInit-LLR-002 195](#_Toc210646047)

[10.8.5.7.3 dauanaiadc-IadcInit-LLR-003 195](#_Toc210646048)

[10.8.5.7.4 dauanaiadc-IadcInit-LLR-004 195](#_Toc210646049)

[10.8.5.7.5 dauanaiadc-IadcInit-LLR-005 196](#_Toc210646050)

[10.8.5.7.6 dauanaiadc-IadcInit-LLR-006 196](#_Toc210646051)

[10.8.5.7.7 dauanaiadc-IadcInit-LLR-007 196](#_Toc210646052)

[10.8.5.7.8 dauanaiadc-IadcInit-LLR-008 196](#_Toc210646053)

[10.8.5.7.9 dauanaiadc-IadcInit-LLR-009 196](#_Toc210646054)

[10.8.5.7.10 dauanaiadc-IadcInit-LLR-010 196](#_Toc210646055)

[10.8.5.7.11 dauanaiadc-IadcInit-LLR-011 197](#_Toc210646056)

[10.8.5.7.12 dauanaiadc-IadcInit-LLR-012 197](#_Toc210646057)

[10.8.5.7.13 dauanaiadc-IadcInit-LLR-013 197](#_Toc210646058)

[10.8.5.7.14 dauanaiadc-IadcInit-LLR-014 197](#_Toc210646059)

[10.9 dauanainit 197](#_Toc210646060)

[10.9.1 InitTask 197](#_Toc210646061)

[10.9.1.1 Brief Description 198](#_Toc210646062)

[10.9.1.2 List of HLRs allocated 198](#_Toc210646063)

[10.9.1.3 List of global variables accessed and modified 198](#_Toc210646064)

[10.9.1.4 Parameter list (Input/Output) 198](#_Toc210646065)

[10.9.1.5 Return Value 198](#_Toc210646066)

[10.9.1.6 Other CSUs called by this CSU 199](#_Toc210646067)

[10.9.1.7 Description of list of LLRs allocated 199](#_Toc210646068)

[10.9.1.7.1 dauanainit-InitTask-LLR-001 199](#_Toc210646069)

[10.9.1.7.2 dauanainit-InitTask-LLR-002 199](#_Toc210646070)

[10.9.1.7.3 dauanainit-InitTask-LLR-003 199](#_Toc210646071)

[10.9.1.7.4 dauanainit-InitTask-LLR-004 200](#_Toc210646072)

[10.9.1.7.5 dauanainit-InitTask-LLR-005 200](#_Toc210646073)

[10.9.1.7.6 dauanainit-InitTask-LLR-006 200](#_Toc210646074)

[10.9.1.7.7 dauanainit-InitTask-LLR-007 200](#_Toc210646075)

[10.9.1.7.8 dauanainit-InitTask-LLR-008 201](#_Toc210646076)

[10.9.1.7.9 dauanainit-InitTask-LLR-009 201](#_Toc210646077)

[10.9.2 InitInit 201](#_Toc210646078)

[10.9.2.1 Brief Description 201](#_Toc210646079)

[10.9.2.2 List of HLRs allocated 201](#_Toc210646080)

[10.9.2.3 List of global variables accessed and modified 202](#_Toc210646081)

[10.9.2.4 Parameter list (Input/Output) 202](#_Toc210646082)

[10.9.2.5 Return Value 202](#_Toc210646083)

[10.9.2.6 Other CSUs called by this CSU 202](#_Toc210646084)

[10.9.2.7 Description of list of LLRs allocated 202](#_Toc210646085)

[10.9.2.7.1 dauanainit-InitInit-LLR-001 202](#_Toc210646086)

[10.9.2.7.2 dauanainit-InitInit-LLR-002 203](#_Toc210646087)

[10.9.2.7.3 dauanainit-InitInit-LLR-003 203](#_Toc210646088)

[10.10 dauanaintr 203](#_Toc210646089)

[10.10.1 ResetIsr 203](#_Toc210646090)

[10.10.1.1 Brief Description 203](#_Toc210646091)

[10.10.1.2 List of HLRs allocated 204](#_Toc210646092)

[10.10.1.3 List of global variables accessed and modified 204](#_Toc210646093)

[10.10.1.4 Parameter list (Input/Output) 204](#_Toc210646094)

[10.10.1.5 Return Value 204](#_Toc210646095)

[10.10.1.6 Other CSUs called by this CSU 204](#_Toc210646096)

[10.10.1.7 Description of list of LLRs allocated 204](#_Toc210646097)

[10.10.1.7.1 dauanaintr-ResetIsr-LLR-001 204](#_Toc210646098)

[10.10.2 NonMaskable 205](#_Toc210646099)

[10.10.2.1 Brief Description 205](#_Toc210646100)

[10.10.2.2 List of HLRs allocated 205](#_Toc210646101)

[10.10.2.3 List of global variables accessed and modified 205](#_Toc210646102)

[10.10.2.4 Parameter list (Input/Output) 205](#_Toc210646103)

[10.10.2.5 Return Value 205](#_Toc210646104)

[10.10.2.6 Other CSUs called by this CSU 206](#_Toc210646105)

[10.10.2.7 Description of list of LLRs allocated 206](#_Toc210646106)

[10.10.2.7.1 dauanaintr-NonMaskable-LLR-001 206](#_Toc210646107)

[10.10.3 HardFault 206](#_Toc210646108)

[10.10.3.1 Brief Description 206](#_Toc210646109)

[10.10.3.2 List of HLRs allocated 206](#_Toc210646110)

[10.10.3.3 List of global variables accessed and modified 206](#_Toc210646111)

[10.10.3.4 Parameter list (Input/Output) 207](#_Toc210646112)

[10.10.3.5 Return Value 207](#_Toc210646113)

[10.10.3.6 Other CSUs called by this CSU 207](#_Toc210646114)

[10.10.3.7 Description of list of LLRs allocated 207](#_Toc210646115)

[10.10.3.7.1 dauanaintr-HardFault-LLR-001 207](#_Toc210646116)

[10.10.4 MemManage 207](#_Toc210646117)

[10.10.4.1 Brief Description 208](#_Toc210646118)

[10.10.4.2 List of HLRs allocated 208](#_Toc210646119)

[10.10.4.3 List of global variables accessed and modified 208](#_Toc210646120)

[10.10.4.4 Parameter list (Input/Output) 208](#_Toc210646121)

[10.10.4.5 Return Value 208](#_Toc210646122)

[10.10.4.6 Other CSUs called by this CSU 208](#_Toc210646123)

[10.10.4.7 Description of list of LLRs allocated 208](#_Toc210646124)

[10.10.4.7.1 dauanaintr-MemManage-LLR-001 209](#_Toc210646125)

[10.10.4.7.2 dauanaintr-MemManage-LLR-002 209](#_Toc210646126)

[10.10.5 BusFault 209](#_Toc210646127)

[10.10.5.1 Brief Description 209](#_Toc210646128)

[10.10.5.2 List of HLRs allocated 210](#_Toc210646129)

[10.10.5.3 List of global variables accessed and modified 210](#_Toc210646130)

[10.10.5.4 Parameter list (Input/Output) 210](#_Toc210646131)

[10.10.5.5 Return Value 210](#_Toc210646132)

[10.10.5.6 Other CSUs called by this CSU 210](#_Toc210646133)

[10.10.5.7 Description of list of LLRs allocated 210](#_Toc210646134)

[10.10.5.7.1 dauanaintr-BusFault-LLR-001 210](#_Toc210646135)

[10.10.6 UsageFault 211](#_Toc210646136)

[10.10.6.1 Brief Description 211](#_Toc210646137)

[10.10.6.2 List of HLRs allocated 211](#_Toc210646138)

[10.10.6.3 List of global variables accessed and modified 211](#_Toc210646139)

[10.10.6.4 Parameter list (Input/Output) 211](#_Toc210646140)

[10.10.6.5 Return Value 211](#_Toc210646141)

[10.10.6.6 Other CSUs called by this CSU 212](#_Toc210646142)

[10.10.6.7 Description of list of LLRs allocated 212](#_Toc210646143)

[10.10.6.7.1 dauanaintr-UsageFault-LLR-001 212](#_Toc210646144)

[10.10.7 SpuriousInterrupt 212](#_Toc210646145)

[10.10.7.1 Brief Description 212](#_Toc210646146)

[10.10.7.2 List of HLRs allocated 212](#_Toc210646147)

[10.10.7.3 List of global variables accessed and modified 212](#_Toc210646148)

[10.10.7.4 Parameter list (Input/Output) 213](#_Toc210646149)

[10.10.7.5 Return Value 213](#_Toc210646150)

[10.10.7.6 Other CSUs called by this CSU 213](#_Toc210646151)

[10.10.7.7 Description of list of LLRs allocated 213](#_Toc210646152)

[10.10.7.7.1 dauanaintr-SpuriousInterrupt-LLR-001 213](#_Toc210646153)

[10.10.8 IntrInit 213](#_Toc210646154)

[10.10.8.1 Brief Description 214](#_Toc210646155)

[10.10.8.2 List of HLRs allocated 214](#_Toc210646156)

[10.10.8.3 List of global variables accessed and modified 214](#_Toc210646157)

[10.10.8.4 Parameter list (Input/Output) 214](#_Toc210646158)

[10.10.8.5 Return Value 214](#_Toc210646159)

[10.10.8.6 Other CSUs called by this CSU 214](#_Toc210646160)

[10.10.8.7 Description of list of LLRs allocated 214](#_Toc210646161)

[10.10.8.7.1 dauanaintr-IntrInit-LLR-001 215](#_Toc210646162)

[10.10.9 IntrInstall 215](#_Toc210646163)

[10.10.9.1 Brief Description 215](#_Toc210646164)

[10.10.9.2 List of HLRs allocated 215](#_Toc210646165)

[10.10.9.3 List of global variables accessed and modified 215](#_Toc210646166)

[10.10.9.4 Parameter list (Input/Output) 215](#_Toc210646167)

[10.10.9.5 Return Value 216](#_Toc210646168)

[10.10.9.6 Other CSUs called by this CSU 216](#_Toc210646169)

[10.10.9.7 Description of list of LLRs allocated 216](#_Toc210646170)

[10.10.9.7.1 dauanaintr-IntrInstall-LLR-001 216](#_Toc210646171)

[10.10.9.7.2 dauanaintr-IntrInstall-LLR-002 216](#_Toc210646172)

[10.10.9.7.3 dauanaintr-IntrInstall-LLR-003 216](#_Toc210646173)

[10.10.10 IntrGetCurrentIntrFn 217](#_Toc210646174)

[10.10.10.1 Brief Description 217](#_Toc210646175)

[10.10.10.2 List of HLRs allocated 217](#_Toc210646176)

[10.10.10.3 List of global variables accessed and modified 217](#_Toc210646177)

[10.10.10.4 Parameter list (Input/Output) 217](#_Toc210646178)

[10.10.10.5 Return Value 217](#_Toc210646179)

[10.10.10.6 Other CSUs called by this CSU 218](#_Toc210646180)

[10.10.10.7 Description of list of LLRs allocated 218](#_Toc210646181)

[10.10.10.7.1 dauanaintr-IntrGetCurrentIntrFn-LLR-001 218](#_Toc210646182)

[10.11 dauanalookup 218](#_Toc210646183)

[10.11.1 LookupTableLookup 218](#_Toc210646184)

[10.11.1.1 Brief Description 218](#_Toc210646185)

[10.11.1.2 List of HLRs allocated 218](#_Toc210646186)

[10.11.1.3 List of global variables accessed and modified 219](#_Toc210646187)

[10.11.1.4 Parameter list (Input/Output) 219](#_Toc210646188)

[10.11.1.5 Return Value 219](#_Toc210646189)

[10.11.1.6 Other CSUs called by this CSU 219](#_Toc210646190)

[10.11.1.7 Description of list of LLRs allocated 219](#_Toc210646191)

[10.11.1.7.1 dauanalookup-LookupTableLookup-LLR-001 219](#_Toc210646192)

[10.11.1.7.2 dauanalookup-LookupTableLookup-LLR-002 220](#_Toc210646193)

[10.11.1.7.3 dauanalookup-LookupTableLookup-LLR-003 220](#_Toc210646194)

[10.11.1.7.4 dauanalookup-LookupTableLookup-LLR-004 220](#_Toc210646195)

[10.11.1.7.5 dauanalookup-LookupTableLookup-LLR-005 221](#_Toc210646196)

[10.11.1.7.6 dauanalookup-LookupTableLookup-LLR-006 221](#_Toc210646197)

[10.11.1.7.7 dauanalookup-LookupTableLookup-LLR-007 222](#_Toc210646198)

[10.11.2 LookupTableInverse 222](#_Toc210646199)

[10.11.2.1 Brief Description 222](#_Toc210646200)

[10.11.2.2 List of HLRs allocated 222](#_Toc210646201)

[10.11.2.3 List of global variables accessed and modified 222](#_Toc210646202)

[10.11.2.4 Parameter list (Input/Output) 222](#_Toc210646203)

[10.11.2.5 Return Value 223](#_Toc210646204)

[10.11.2.6 Other CSUs called by this CSU 223](#_Toc210646205)

[10.11.2.7 Description of list of LLRs allocated 223](#_Toc210646206)

[10.11.2.7.1 dauanalookup-LookupTableInverse-LLR-001 223](#_Toc210646207)

[10.11.2.7.2 dauanalookup-LookupTableInverse-LLR-002 223](#_Toc210646208)

[10.11.2.7.3 dauanalookup-LookupTableInverse-LLR-003 223](#_Toc210646209)

[10.11.3 LookupTableLookup32 224](#_Toc210646210)

[10.11.3.1 Brief Description 224](#_Toc210646211)

[10.11.3.2 List of HLRs allocated 224](#_Toc210646212)

[10.11.3.3 List of global variables accessed and modified 224](#_Toc210646213)

[10.11.3.4 Parameter list (Input/Output) 224](#_Toc210646214)

[10.11.3.5 Return Value 225](#_Toc210646215)

[10.11.3.6 Other CSUs called by this CSU 225](#_Toc210646216)

[10.11.3.7 Description of list of LLRs allocated 225](#_Toc210646217)

[10.11.3.7.1 dauanalookup-LookupTableLookup32-LLR-001 225](#_Toc210646218)

[10.11.3.7.2 dauanalookup-LookupTableLookup32-LLR-002 225](#_Toc210646219)

[10.11.3.7.3 dauanalookup-LookupTableLookup32-LLR-003 226](#_Toc210646220)

[10.11.3.7.4 dauanalookup-LookupTableLookup32-LLR-004 226](#_Toc210646221)

[10.11.3.7.5 dauanalookup-LookupTableLookup32-LLR-005 227](#_Toc210646222)

[10.11.3.7.6 dauanalookup-LookupTableLookup32-LLR-006 227](#_Toc210646223)

[10.11.3.7.7 dauanalookup-LookupTableLookup32-LLR-007 228](#_Toc210646224)

[10.11.3.7.8 dauanalookup-LookupTableLookup32-LLR-008 228](#_Toc210646225)

[10.11.3.7.9 dauanalookup-LookupTableLookup32-LLR-009 229](#_Toc210646226)

[10.11.3.7.10 dauanalookup-LookupTableLookup32-LLR-010 229](#_Toc210646227)

[10.11.3.7.11 dauanalookup-LookupTableLookup32-LLR-011 229](#_Toc210646228)

[10.11.3.7.12 dauanalookup-LookupTableLookup32-LLR-012 230](#_Toc210646229)

[10.11.3.7.13 dauanalookup-LookupTableLookup32-LLR-013 230](#_Toc210646230)

[10.11.3.7.14 dauanalookup-LookupTableLookup32-LLR-014 230](#_Toc210646231)

[10.11.3.7.15 dauanalookup-LookupTableLookup32-LLR-015 231](#_Toc210646232)

[10.11.3.7.16 dauanalookup-LookupTableLookup32-LLR-016 231](#_Toc210646233)

[10.11.4 LookupTableInverse32 231](#_Toc210646234)

[10.11.4.1 Brief Description 231](#_Toc210646235)

[10.11.4.2 List of HLRs allocated 231](#_Toc210646236)

[10.11.4.3 List of global variables accessed and modified 231](#_Toc210646237)

[10.11.4.4 Parameter list (Input/Output) 232](#_Toc210646238)

[10.11.4.5 Return Value 232](#_Toc210646239)

[10.11.4.6 Other CSUs called by this CSU 232](#_Toc210646240)

[10.11.4.7 Description of list of LLRs allocated 232](#_Toc210646241)

[10.11.4.7.1 dauanalookup-LookupTableInverse32-LLR-001 232](#_Toc210646242)

[10.11.4.7.2 dauanalookup-LookupTableInverse32-LLR-002 232](#_Toc210646243)

[10.11.4.7.3 dauanalookup-LookupTableInverse32-LLR-003 233](#_Toc210646244)

[10.12 dauanamain 233](#_Toc210646245)

[10.12.1 MainFunc 233](#_Toc210646246)

[10.12.1.1 Brief Description 233](#_Toc210646247)

[10.12.1.2 List of HLRs allocated 233](#_Toc210646248)

[10.12.1.3 List of global variables accessed and modified 234](#_Toc210646249)

[10.12.1.4 Parameter list (Input/Output) 234](#_Toc210646250)

[10.12.1.5 Return Value 234](#_Toc210646251)

[10.12.1.6 Other CSUs called by this CSU 234](#_Toc210646252)

[10.12.1.7 Description of list of LLRs allocated 234](#_Toc210646253)

[10.12.1.7.1 dauanamain-MainFunc-LLR-001 234](#_Toc210646254)

[10.12.1.7.2 dauanamain-MainFunc-LLR-002 235](#_Toc210646255)

[10.12.1.7.3 dauanamain-MainFunc-LLR-003 235](#_Toc210646256)

[10.12.1.7.4 dauanamain-MainFunc-LLR-004 235](#_Toc210646257)

[10.12.1.7.5 dauanamain-MainFunc-LLR-005 235](#_Toc210646258)

[10.12.1.7.6 dauanamain-MainFunc-LLR-006 235](#_Toc210646259)

[10.13 dauanaoscpu 235](#_Toc210646260)

[10.13.1 OsTaskCreate 235](#_Toc210646261)

[10.13.1.1 Brief Description 236](#_Toc210646262)

[10.13.1.2 List of HLRs allocated 236](#_Toc210646263)

[10.13.1.3 List of global variables accessed and modified 236](#_Toc210646264)

[10.13.1.4 Parameter list (Input/Output) 236](#_Toc210646265)

[10.13.1.5 Return Value 236](#_Toc210646266)

[10.13.1.6 Other CSUs called by this CSU 237](#_Toc210646267)

[10.13.1.7 Description of list of LLRs allocated 237](#_Toc210646268)

[10.13.1.7.1 dauanaoscpu-OsTaskCreate-LLR-001 237](#_Toc210646269)

[10.13.1.7.2 dauanaoscpu-OsTaskCreate-LLR-002 237](#_Toc210646270)

[10.13.1.7.3 dauanaoscpu-OsTaskCreate-LLR-003 237](#_Toc210646271)

[10.13.1.7.4 dauanaoscpu-OsTaskCreate-LLR-004 238](#_Toc210646272)

[10.14 dauanaoscpua 238](#_Toc210646273)

[10.14.1 OsCtxSw 238](#_Toc210646274)

[10.14.1.1 Brief Description 239](#_Toc210646275)

[10.14.1.2 List of HLRs allocated 239](#_Toc210646276)

[10.14.1.3 List of global variables accessed and modified 239](#_Toc210646277)

[10.14.1.4 Parameter list (Input/Output) 239](#_Toc210646278)

[10.14.1.5 Return Value 239](#_Toc210646279)

[10.14.1.6 Other CSUs called by this CSU 239](#_Toc210646280)

[10.14.1.7 Description of list of LLRs allocated 239](#_Toc210646281)

[10.14.1.7.1 dauanaoscpua-OsCtxSw-LLR-001 240](#_Toc210646282)

[10.14.2 OsIntCtxSw 240](#_Toc210646283)

[10.14.2.1 Brief Description 240](#_Toc210646284)

[10.14.2.2 List of HLRs allocated 240](#_Toc210646285)

[10.14.2.3 List of global variables accessed and modified 240](#_Toc210646286)

[10.14.2.4 Parameter list (Input/Output) 240](#_Toc210646287)

[10.14.2.5 Return Value 241](#_Toc210646288)

[10.14.2.6 Other CSUs called by this CSU 241](#_Toc210646289)

[10.14.2.7 Description of list of LLRs allocated 241](#_Toc210646290)

[10.14.2.7.1 dauanaoscpua-OsIntCtxSw-LLR-001 241](#_Toc210646291)

[10.14.3 SaveStatusReg 241](#_Toc210646292)

[10.14.3.1 Brief Description 241](#_Toc210646293)

[10.14.3.2 List of HLRs allocated 242](#_Toc210646294)

[10.14.3.3 List of global variables accessed and modified 242](#_Toc210646295)

[10.14.3.4 Parameter list (Input/Output) 242](#_Toc210646296)

[10.14.3.5 Return Value 242](#_Toc210646297)

[10.14.3.6 Other CSUs called by this CSU 242](#_Toc210646298)

[10.14.3.7 Description of list of LLRs allocated 242](#_Toc210646299)

[10.14.3.7.1 dauanaoscpua-SaveStatusReg-LLR-001 242](#_Toc210646300)

[10.14.4 RestoreStatusReg 243](#_Toc210646301)

[10.14.4.1 Brief Description 243](#_Toc210646302)

[10.14.4.2 List of HLRs allocated 243](#_Toc210646303)

[10.14.4.3 List of global variables accessed and modified 243](#_Toc210646304)

[10.14.4.4 Parameter list (Input/Output) 243](#_Toc210646305)

[10.14.4.5 Return Value 243](#_Toc210646306)

[10.14.4.6 Other CSUs called by this CSU 244](#_Toc210646307)

[10.14.4.7 Description of list of LLRs allocated 244](#_Toc210646308)

[10.14.4.7.1 dauanaoscpua-RestoreStatusReg-LLR-001 244](#_Toc210646309)

[10.14.5 OsStartHighRdy 244](#_Toc210646310)

[10.14.5.1 Brief Description 244](#_Toc210646311)

[10.14.5.2 List of HLRs allocated 244](#_Toc210646312)

[10.14.5.3 List of global variables accessed and modified 244](#_Toc210646313)

[10.14.5.4 Parameter list (Input/Output) 245](#_Toc210646314)

[10.14.5.5 Return Value 245](#_Toc210646315)

[10.14.5.6 Other CSUs called by this CSU 245](#_Toc210646316)

[10.14.5.7 Description of list of LLRs allocated 245](#_Toc210646317)

[10.14.5.7.1 dauanaoscpua-OsStartHighRdy-LLR-001 245](#_Toc210646318)

[10.14.5.7.2 dauanaoscpua-OsStartHighRdy-LLR-002 245](#_Toc210646319)

[10.14.5.7.3 dauanaoscpua-OsStartHighRdy-LLR-003 246](#_Toc210646320)

[10.14.5.7.4 dauanaoscpua-OsStartHighRdy-LLR-004 246](#_Toc210646321)

[10.14.5.7.5 dauanaoscpua-OsStartHighRdy-LLR-005 246](#_Toc210646322)

[10.14.5.7.6 dauanaoscpua-OsStartHighRdy-LLR-006 246](#_Toc210646323)

[10.14.6 PendSvHandler 246](#_Toc210646324)

[10.14.6.1 Brief Description 247](#_Toc210646325)

[10.14.6.2 List of HLRs allocated 247](#_Toc210646326)

[10.14.6.3 List of global variables accessed and modified 247](#_Toc210646327)

[10.14.6.4 Parameter list (Input/Output) 247](#_Toc210646328)

[10.14.6.5 Return Value 247](#_Toc210646329)

[10.14.6.6 Other CSUs called by this CSU 247](#_Toc210646330)

[10.14.6.7 Description of list of LLRs allocated 247](#_Toc210646331)

[10.14.6.7.1 dauanaoscpua-PendSvHandler-LLR-001 248](#_Toc210646332)

[10.14.6.7.2 dauanaoscpua-PendSvHandler-LLR-002 248](#_Toc210646333)

[10.14.6.7.3 dauanaoscpua-PendSvHandler-LLR-003 248](#_Toc210646334)

[10.14.6.7.4 dauanaoscpua-PendSvHandler-LLR-004 248](#_Toc210646335)

[10.14.6.7.5 dauanaoscpua-PendSvHandler-LLR-005 248](#_Toc210646336)

[10.14.6.7.6 dauanaoscpua-PendSvHandler-LLR-006 248](#_Toc210646337)

[10.14.6.7.7 dauanaoscpua-PendSvHandler-LLR-007 249](#_Toc210646338)

[10.15 dauanapbit 249](#_Toc210646339)

[10.15.1 RamTest 249](#_Toc210646340)

[10.15.1.1 Brief Description 249](#_Toc210646341)

[10.15.1.2 List of HLRs allocated 250](#_Toc210646342)

[10.15.1.3 List of global variables accessed and modified 250](#_Toc210646343)

[10.15.1.4 Parameter list (Input/Output) 250](#_Toc210646344)

[10.15.1.5 Return Value 250](#_Toc210646345)

[10.15.1.6 Other CSUs called by this CSU 250](#_Toc210646346)

[10.15.1.7 Description of list of LLRs allocated 250](#_Toc210646347)

[10.15.1.7.1 dauanapbit-RamTest-LLR-001 251](#_Toc210646348)

[10.15.2 Cputest 251](#_Toc210646349)

[10.15.2.1 Brief Description 251](#_Toc210646350)

[10.15.2.2 List of HLRs allocated 251](#_Toc210646351)

[10.15.2.3 List of global variables accessed and modified 251](#_Toc210646352)

[10.15.2.4 Parameter list (Input/Output) 251](#_Toc210646353)

[10.15.2.5 Return Value 252](#_Toc210646354)

[10.15.2.6 Other CSUs called by this CSU 252](#_Toc210646355)

[10.15.2.7 Description of list of LLRs allocated 252](#_Toc210646356)

[10.15.2.7.1 dauanapbit-CpuTest-LLR-001 252](#_Toc210646357)

[10.15.2.7.2 dauanapbit-CpuTest-LLR-002 252](#_Toc210646358)

[10.15.2.7.3 dauanapbit-CpuTest-LLR-003 252](#_Toc210646359)

[10.15.3 Mcdcheck 253](#_Toc210646360)

[10.15.3.1 Brief Description 253](#_Toc210646361)

[10.15.3.2 List of HLRs allocated 253](#_Toc210646362)

[10.15.3.3 List of global variables accessed and modified 253](#_Toc210646363)

[10.15.3.4 Parameter list (Input/Output) 253](#_Toc210646364)

[10.15.3.5 Return Value 253](#_Toc210646365)

[10.15.3.6 Other CSUs called by this CSU 253](#_Toc210646366)

[10.15.3.7 Description of list of LLRs allocated 254](#_Toc210646367)

[10.15.3.7.1 dauanapbit-McdCheck-LLR-001 254](#_Toc210646368)

[10.15.3.7.2 dauanapbit-McdCheck-LLR-002 254](#_Toc210646369)

[10.15.3.7.3 dauanapbit-McdCheck-LLR-003 254](#_Toc210646370)

[10.15.3.7.4 dauanapbit-McdCheck-LLR-004 255](#_Toc210646371)

[10.15.3.7.5 dauanapbit-McdCheck-LLR-005 255](#_Toc210646372)

[10.15.3.7.6 dauanapbit-McdCheck-LLR-006 255](#_Toc210646373)

[10.15.3.7.7 dauanapbit-McdCheck-LLR-007 256](#_Toc210646374)

[10.15.3.7.8 dauanapbit-McdCheck-LLR-008 256](#_Toc210646375)

[10.15.3.7.9 dauanapbit-McdCheck-LLR-009 256](#_Toc210646376)

[10.15.3.7.10 dauanapbit-McdCheck-LLR-010 257](#_Toc210646377)

[10.15.3.7.11 dauanapbit-McdCheck-LLR-011 257](#_Toc210646378)

[10.15.3.7.12 dauanapbit-McdCheck-LLR-012 257](#_Toc210646379)

[10.15.3.7.13 dauanapbit-McdCheck-LLR-013 258](#_Toc210646380)

[10.15.3.7.14 dauanapbit-McdCheck-LLR-014 258](#_Toc210646381)

[10.15.3.7.15 dauanapbit-McdCheck-LLR-015 259](#_Toc210646382)

[10.15.3.7.16 dauanapbit-McdCheck-LLR-016 259](#_Toc210646383)

[10.15.3.7.17 dauanapbit-McdCheck-LLR-017 259](#_Toc210646384)

[10.15.3.7.18 dauanapbit-McdCheck-LLR-018 259](#_Toc210646385)

[10.15.3.7.19 dauanapbit-McdCheck-LLR-019 260](#_Toc210646386)

[10.15.3.7.20 dauanapbit-McdCheck-LLR-020 260](#_Toc210646387)

[10.15.3.7.21 dauanapbit-McdCheck-LLR-021 260](#_Toc210646388)

[10.15.3.7.22 dauanapbit-McdCheck-LLR-022 260](#_Toc210646389)

[10.15.3.7.23 dauanapbit-McdCheck-LLR-023 261](#_Toc210646390)

[10.15.3.7.24 dauanapbit-McdCheck-LLR-024 261](#_Toc210646391)

[10.15.3.7.25 dauanapbit-McdCheck-LLR-025 262](#_Toc210646392)

[10.15.3.7.26 dauanapbit-McdCheck-LLR-026 262](#_Toc210646393)

[10.15.3.7.27 dauanapbit-McdCheck-LLR-027 262](#_Toc210646394)

[10.15.3.7.28 dauanapbit-McdCheck-LLR-028 263](#_Toc210646395)

[10.15.3.7.29 dauanapbit-McdCheck-LLR-029 263](#_Toc210646396)

[10.15.3.7.30 dauanapbit-McdCheck-LLR-030 263](#_Toc210646397)

[10.15.4 Crccheck 264](#_Toc210646398)

[10.15.4.1 Brief Description 264](#_Toc210646399)

[10.15.4.2 List of HLRs allocated 264](#_Toc210646400)

[10.15.4.3 List of global variables accessed and modified 264](#_Toc210646401)

[10.15.4.4 Parameter list (Input/Output) 264](#_Toc210646402)

[10.15.4.5 Return Value 264](#_Toc210646403)

[10.15.4.6 Other CSUs called by this CSU 264](#_Toc210646404)

[10.15.4.7 Description of list of LLRs allocated 265](#_Toc210646405)

[10.15.4.7.1 dauanapbit-CrcCheck-LLR-001 265](#_Toc210646406)

[10.15.4.7.2 dauanapbit-CrcCheck-LLR-002 265](#_Toc210646407)

[10.15.4.7.3 dauanapbit-CrcCheck-LLR-003 265](#_Toc210646408)

[10.15.4.7.4 dauanapbit-CrcCheck-LLR-004 265](#_Toc210646409)

[10.15.4.7.5 dauanapbit-CrcCheck-LLR-005 266](#_Toc210646410)

[10.15.4.7.6 dauanapbit-CrcCheck-LLR-006 266](#_Toc210646411)

[10.15.5 BitErrorHandler 266](#_Toc210646412)

[10.15.5.1 Brief Description 266](#_Toc210646413)

[10.15.5.2 List of HLRs allocated 266](#_Toc210646414)

[10.15.5.3 List of global variables accessed and modified 267](#_Toc210646415)

[10.15.5.4 Parameter list (Input/Output) 267](#_Toc210646416)

[10.15.5.5 Return Value 267](#_Toc210646417)

[10.15.5.6 Other CSUs called by this CSU 267](#_Toc210646418)

[10.15.5.7 Description of list of LLRs allocated 267](#_Toc210646419)

[10.15.5.7.1 dauanapbit-BitErrorHandler-LLR-001 267](#_Toc210646420)

[10.15.5.7.2 dauanapbit-BitErrorHandler-LLR-002 268](#_Toc210646421)

[10.15.6 Pbitcheck 268](#_Toc210646422)

[10.15.6.1 Brief Description 268](#_Toc210646423)

[10.15.6.2 List of HLRs allocated 268](#_Toc210646424)

[10.15.6.3 List of global variables accessed and modified 268](#_Toc210646425)

[10.15.6.4 Parameter list (Input/Output) 269](#_Toc210646426)

[10.15.6.5 Return Value 269](#_Toc210646427)

[10.15.6.6 Other CSUs called by this CSU 269](#_Toc210646428)

[10.15.6.7 Description of list of LLRs allocated 269](#_Toc210646429)

[10.15.6.7.1 dauanapbit-PbitCheck-LLR-001 269](#_Toc210646430)

[10.15.6.7.2 dauanapbit-PbitCheck-LLR-002 269](#_Toc210646431)

[10.15.6.7.3 dauanapbit-PbitCheck-LLR-003 269](#_Toc210646432)

[10.15.6.7.4 dauanapbit-PbitCheck-LLR-004 270](#_Toc210646433)

[10.16 dauanarevno 270](#_Toc210646434)

[10.16.1 Brief Description 270](#_Toc210646435)

[10.16.2 List of HLRs allocated 270](#_Toc210646436)

[10.16.3 List of global variables accessed and modified 270](#_Toc210646437)

[10.16.4 Parameter list (Input/Output) 270](#_Toc210646438)

[10.16.5 Return Value 271](#_Toc210646439)

[10.16.6 Other CSUs called by this CSU 271](#_Toc210646440)

[10.16.7 Description of list of LLRs allocated 271](#_Toc210646441)

[10.16.7.1 dauanarevno-LLR-001 271](#_Toc210646442)

[10.17 dauanatbase 271](#_Toc210646443)

[10.17.1 TbaseTaskSignaling 271](#_Toc210646444)

[10.17.1.1 Brief Description 272](#_Toc210646445)

[10.17.1.2 List of HLRs allocated 272](#_Toc210646446)

[10.17.1.3 List of global variables accessed and modified 272](#_Toc210646447)

[10.17.1.4 Parameter list (Input/Output) 272](#_Toc210646448)

[10.17.1.5 Return Value 272](#_Toc210646449)

[10.17.1.6 Other CSUs called by this CSU 272](#_Toc210646450)

[10.17.1.7 Description of list of LLRs allocated 273](#_Toc210646451)

[10.17.1.7.1 dauanatbase-TbaseTaskSignaling-LLR-001 273](#_Toc210646452)

[10.17.1.7.2 dauanatbase-TbaseTaskSignaling-LLR-002 273](#_Toc210646453)

[10.17.1.7.3 dauanatbase-TbaseTaskSignaling-LLR-003 273](#_Toc210646454)

[10.17.1.7.4 dauanatbase-TbaseTaskSignaling-LLR-004 273](#_Toc210646455)

[10.17.1.7.5 dauanatbase-TbaseTaskSignaling-LLR-005 273](#_Toc210646456)

[10.17.2 TbaseIntrHandler 274](#_Toc210646457)

[10.17.2.1 Brief Description 274](#_Toc210646458)

[10.17.2.2 List of HLRs allocated 274](#_Toc210646459)

[10.17.2.3 List of global variables accessed and modified 274](#_Toc210646460)

[10.17.2.4 Parameter list (Input/Output) 274](#_Toc210646461)

[10.17.2.5 Return Value 274](#_Toc210646462)

[10.17.2.6 Other CSUs called by this CSU 274](#_Toc210646463)

[10.17.2.7 Description of list of LLRs allocated 275](#_Toc210646464)

[10.17.2.7.1 dauanatbase-TbaseIntrHandler-LLR-001 275](#_Toc210646465)

[10.17.2.7.2 dauanatbase-TbaseIntrHandler-LLR-002 275](#_Toc210646466)

[10.17.2.7.3 dauanatbase-TbaseIntrHandler-LLR-003 275](#_Toc210646467)

[10.17.2.7.4 dauanatbase-TbaseIntrHandler-LLR-004 276](#_Toc210646468)

[10.17.3 TbaseInit 276](#_Toc210646469)

[10.17.3.1 Brief Description 276](#_Toc210646470)

[10.17.3.2 List of HLRs allocated 276](#_Toc210646471)

[10.17.3.3 List of global variables accessed and modified 276](#_Toc210646472)

[10.17.3.4 Parameter list (Input/Output) 276](#_Toc210646473)

[10.17.3.5 Return Value 276](#_Toc210646474)

[10.17.3.6 Other CSUs called by this CSU 277](#_Toc210646475)

[10.17.3.7 Description of list of LLRs allocated 277](#_Toc210646476)

[10.17.3.7.1 dauanatbase-TbaseInit-LLR-001 277](#_Toc210646477)

[10.18 dauanatmr 277](#_Toc210646478)

[10.18.1 TMRInitTIM6 277](#_Toc210646479)

[10.18.1.1 Brief Description 277](#_Toc210646480)

[10.18.1.2 List of HLRs allocated 277](#_Toc210646481)

[10.18.1.3 List of global variables accessed and modified 278](#_Toc210646482)

[10.18.1.4 Parameter list (Input/Output) 278](#_Toc210646483)

[10.18.1.5 Return Value 278](#_Toc210646484)

[10.18.1.6 Other CSUs called by this CSU 278](#_Toc210646485)

[10.18.1.7 Description of list of LLRs allocated 278](#_Toc210646486)

[10.18.1.7.1 dauanatmr-TMRInitTIM6-LLR-001 279](#_Toc210646487)

[10.18.1.7.2 dauanatmr-TMRInitTIM6-LLR-002 279](#_Toc210646488)

[10.18.1.7.3 dauanatmr-TMRInitTIM6-LLR-003 279](#_Toc210646489)

[10.18.1.7.4 dauanatmr-TMRInitTIM6-LLR-004 279](#_Toc210646490)

[10.18.1.7.5 dauanatmr-TMRInitTIM6-LLR-005 279](#_Toc210646491)

[10.18.1.7.6 dauanatmr-TMRInitTIM6-LLR-006 279](#_Toc210646492)

[10.18.1.7.7 dauanatmr-TMRInitTIM6-LLR-007 280](#_Toc210646493)

[10.18.2 TMRInitTIM7 280](#_Toc210646494)

[10.18.2.1 Brief Description 280](#_Toc210646495)

[10.18.2.2 List of HLRs allocated 280](#_Toc210646496)

[10.18.2.3 List of global variables accessed and modified 280](#_Toc210646497)

[10.18.2.4 Parameter list (Input/Output) 280](#_Toc210646498)

[10.18.2.5 Return Value 281](#_Toc210646499)

[10.18.2.6 Other CSUs called by this CSU 281](#_Toc210646500)

[10.18.2.7 Description of list of LLRs allocated 281](#_Toc210646501)

[10.18.2.7.1 dauanatmr-TMRInitTIM7-LLR-001 281](#_Toc210646502)

[10.18.2.7.2 dauanatmr-TMRInitTIM7-LLR-002 281](#_Toc210646503)

[10.18.2.7.3 dauanatmr-TMRInitTIM7-LLR-003 282](#_Toc210646504)

[10.18.2.7.4 dauanatmr-TMRInitTIM7-LLR-004 282](#_Toc210646505)

[10.18.2.7.5 dauanatmr-TMRInitTIM7-LLR-005 282](#_Toc210646506)

[10.18.2.7.6 dauanatmr-TMRInitTIM7-LLR-006 282](#_Toc210646507)

[10.18.2.7.7 dauanatmr-TMRInitTIM7-LLR-007 282](#_Toc210646508)

[10.18.3 TMRInitTIM12 283](#_Toc210646509)

[10.18.3.1 Brief Description 283](#_Toc210646510)

[10.18.3.2 List of HLRs allocated 283](#_Toc210646511)

[10.18.3.3 List of global variables accessed and modified 283](#_Toc210646512)

[10.18.3.4 Parameter list (Input/Output) 283](#_Toc210646513)

[10.18.3.5 Return Value 283](#_Toc210646514)

[10.18.3.6 Other CSUs called by this CSU 284](#_Toc210646515)

[10.18.3.7 Description of list of LLRs allocated 284](#_Toc210646516)

[10.18.3.7.1 dauanatmr-TMRInitTIM12-LLR-001 284](#_Toc210646517)

[10.18.3.7.2 dauanatmr-TMRInitTIM12-LLR-002 284](#_Toc210646518)

[10.18.3.7.3 dauanatmr-TMRInitTIM12-LLR-003 284](#_Toc210646519)

[10.18.3.7.4 dauanatmr-TMRInitTIM12-LLR-004 285](#_Toc210646520)

[10.18.3.7.5 dauanatmr-TMRInitTIM12-LLR-005 285](#_Toc210646521)

[10.18.3.7.6 dauanatmr-TMRInitTIM12-LLR-006 285](#_Toc210646522)

[10.18.3.7.7 dauanatmr-TMRInitTIM12-LLR-007 285](#_Toc210646523)

[10.18.4 GetDivConfig 285](#_Toc210646524)

[10.18.4.1 Brief Description 286](#_Toc210646525)

[10.18.4.2 List of HLRs allocated 286](#_Toc210646526)

[10.18.4.3 List of global variables accessed and modified. 286](#_Toc210646527)

[10.18.4.4 Parameter list (Input/Output) 286](#_Toc210646528)

[10.18.4.5 Return Value 286](#_Toc210646529)

[10.18.4.6 Other CSUs called by this CSU 287](#_Toc210646530)

[10.18.4.7 Description of list of LLRs allocated. 287](#_Toc210646531)

[10.18.4.7.1 dauanatmr-GetDivConfig-LLR-001 287](#_Toc210646532)

[10.18.4.7.2 dauanatmr-GetDivConfig-LLR-002 287](#_Toc210646533)

[10.18.4.7.3 dauanatmr-GetDivConfig-LLR-003 287](#_Toc210646534)

[10.18.4.7.4 dauanatmr-GetDivConfig-LLR-004 287](#_Toc210646535)

[10.18.5 TMRInitTach1 288](#_Toc210646536)

[10.18.5.1 Brief Description 288](#_Toc210646537)

[10.18.5.2 List of HLRs allocated 288](#_Toc210646538)

[10.18.5.3 List of global variables accessed and modified. 288](#_Toc210646539)

[10.18.5.4 Parameter list (Input/Output) 288](#_Toc210646540)

[10.18.5.5 Return Value 289](#_Toc210646541)

[10.18.5.6 Other CSUs called by this CSU 289](#_Toc210646542)

[10.18.5.7 Description of list of LLRs allocated. 289](#_Toc210646543)

[10.18.5.7.1 dauanatmr-TMRInitTach1-LLR-001 289](#_Toc210646544)

[10.18.5.7.2 dauanatmr-TMRInitTach1-LLR-002 290](#_Toc210646545)

[10.18.5.7.3 dauanatmr-TMRInitTach1-LLR-003 290](#_Toc210646546)

[10.18.5.7.4 dauanatmr-TMRInitTach1-LLR-004 290](#_Toc210646547)

[10.18.5.7.5 dauanatmr-TMRInitTach1-LLR-005 290](#_Toc210646548)

[10.18.5.7.6 dauanatmr-TMRInitTach1-LLR-006 290](#_Toc210646549)

[10.18.5.7.7 dauanatmr-TMRInitTach1-LLR-007 291](#_Toc210646550)

[10.18.5.7.8 dauanatmr-TMRInitTach1-LLR-008 291](#_Toc210646551)

[10.18.5.7.9 dauanatmr-TMRInitTach1-LLR-009 291](#_Toc210646552)

[10.18.5.7.10 dauanatmr-TMRInitTach1-LLR-010 292](#_Toc210646553)

[10.18.5.7.11 dauanatmr-TMRInitTach1-LLR-011 292](#_Toc210646554)

[10.18.5.7.12 dauanatmr-TMRInitTach1-LLR-012 292](#_Toc210646555)

[10.18.6 TMRInitTach2 292](#_Toc210646556)

[10.18.6.1 Brief Description 292](#_Toc210646557)

[10.18.6.2 List of HLRs allocated 293](#_Toc210646558)

[10.18.6.3 List of global variables accessed and modified 293](#_Toc210646559)

[10.18.6.4 Parameter list (Input/Output) 293](#_Toc210646560)

[10.18.6.5 Return Value 293](#_Toc210646561)

[10.18.6.6 Other CSUs called by this CSU 293](#_Toc210646562)

[10.18.6.7 Description of list of LLRs allocated 294](#_Toc210646563)

[10.18.6.7.2 dauanatmr-TMRInitTach2-LLR-002 294](#_Toc210646564)

[10.18.6.7.3 dauanatmr-TMRInitTach2-LLR-003 294](#_Toc210646565)

[10.18.6.7.4 dauanatmr-TMRInitTach2-LLR-004 294](#_Toc210646566)

[10.18.6.7.5 dauanatmr-TMRInitTach2-LLR-005 295](#_Toc210646567)

[10.18.6.7.6 dauanatmr-TMRInitTach2-LLR-006 295](#_Toc210646568)

[10.18.6.7.7 dauanatmr-TMRInitTach2-LLR-007 295](#_Toc210646569)

[10.18.6.7.8 dauanatmr-TMRInitTach2-LLR-008 296](#_Toc210646570)

[10.18.6.7.9 dauanatmr-TMRInitTach2-LLR-009 296](#_Toc210646571)

[10.18.6.7.10 dauanatmr-TMRInitTach2-LLR-010 296](#_Toc210646572)

[10.18.6.7.11 dauanatmr-TMRInitTach2-LLR-011 296](#_Toc210646573)

[10.18.6.7.12 dauanatmr-TMRInitTach2-LLR-012 297](#_Toc210646574)

[10.18.7 TMRInitTach3 297](#_Toc210646575)

[10.18.7.1 Brief Description 297](#_Toc210646576)

[10.18.7.2 List of HLRs allocated 297](#_Toc210646577)

[10.18.7.3 List of global variables accessed and modified 297](#_Toc210646578)

[10.18.7.4 Parameter list (Input/Output) 297](#_Toc210646579)

[10.18.7.5 Return Value 298](#_Toc210646580)

[10.18.7.6 Other CSUs called by this CSU 298](#_Toc210646581)

[10.18.7.7 Description of list of LLRs allocated 298](#_Toc210646582)

[10.18.7.7.1 dauanatmr-TMRInitTach3-LLR-001 298](#_Toc210646583)

[10.18.7.7.2 dauanatmr-TMRInitTach3-LLR-002 299](#_Toc210646584)

[10.18.7.7.3 dauanatmr-TMRInitTach3-LLR-003 299](#_Toc210646585)

[10.18.7.7.4 dauanatmr-TMRInitTach3-LLR-004 299](#_Toc210646586)

[10.18.7.7.5 dauanatmr-TMRInitTach3-LLR-005 299](#_Toc210646587)

[10.18.7.7.6 dauanatmr-TMRInitTach3-LLR-006 299](#_Toc210646588)

[10.18.7.7.7 dauanatmr-TMRInitTach3-LLR-007 300](#_Toc210646589)

[10.18.7.7.8 dauanatmr-TMRInitTach3-LLR-008 300](#_Toc210646590)

[10.18.7.7.9 dauanatmr-TMRInitTach3-LLR-009 300](#_Toc210646591)

[10.18.7.7.10 dauanatmr-TMRInitTach3-LLR-010 301](#_Toc210646592)

[10.18.7.7.11 dauanatmr-TMRInitTach3-LLR-011 301](#_Toc210646593)

[10.18.7.7.12 dauanatmr-TMRInitTach3-LLR-012 301](#_Toc210646594)

[10.18.8 TMRInitTach4 301](#_Toc210646595)

[10.18.8.1 Brief Description 301](#_Toc210646596)

[10.18.8.2 List of HLRs allocated 302](#_Toc210646597)

[10.18.8.3 List of global variables accessed and modified 302](#_Toc210646598)

[10.18.8.4 Parameter list (Input/Output) 302](#_Toc210646599)

[10.18.8.5 Return Value 302](#_Toc210646600)

[10.18.8.6 Other CSUs called by this CSU 302](#_Toc210646601)

[10.18.8.7 Description of list of LLRs allocated 303](#_Toc210646602)

[10.18.8.7.1 dauanatmr-TMRInitTach4-LLR-001 303](#_Toc210646603)

[10.18.8.7.2 dauanatmr-TMRInitTach4-LLR-002 303](#_Toc210646604)

[10.18.8.7.3 dauanatmr-TMRInitTach4-LLR-003 303](#_Toc210646605)

[10.18.8.7.4 dauanatmr-TMRInitTach4-LLR-004 303](#_Toc210646606)

[10.18.8.7.5 dauanatmr-TMRInitTach4-LLR-005 304](#_Toc210646607)

[10.18.8.7.6 dauanatmr-TMRInitTach4-LLR-006 304](#_Toc210646608)

[10.18.8.7.7 dauanatmr-TMRInitTach4-LLR-007 304](#_Toc210646609)

[10.18.8.7.8 dauanatmr-TMRInitTach4-LLR-008 305](#_Toc210646610)

[10.18.8.7.9 dauanatmr-TMRInitTach4-LLR-009 305](#_Toc210646611)

[10.18.8.7.10 dauanatmr-TMRInitTach4-LLR-010 305](#_Toc210646612)

[10.18.8.7.11 dauanatmr-TMRInitTach4-LLR-011 305](#_Toc210646613)

[10.18.8.7.12 dauanatmr-TMRInitTach4-LLR-012 306](#_Toc210646614)

[10.18.9 TMRInitTach5 306](#_Toc210646615)

[10.18.9.1 Brief Description 306](#_Toc210646616)

[10.18.9.2 List of HLRs allocated 306](#_Toc210646617)

[10.18.9.3 List of global variables accessed and modified 306](#_Toc210646618)

[10.18.9.4 Parameter list (Input/Output) 307](#_Toc210646619)

[10.18.9.5 Return Value 307](#_Toc210646620)

[10.18.9.6 Other CSUs called by this CSU 307](#_Toc210646621)

[10.18.9.7 Description of list of LLRs allocated 307](#_Toc210646622)

[10.18.9.7.1 dauanatmr-TMRInitTach5-LLR-001 307](#_Toc210646623)

[10.18.9.7.2 dauanatmr-TMRInitTach5-LLR-002 308](#_Toc210646624)

[10.18.9.7.3 dauanatmr-TMRInitTach5-LLR-003 308](#_Toc210646625)

[10.18.9.7.4 dauanatmr-TMRInitTach5-LLR-004 308](#_Toc210646626)

[10.18.9.7.5 dauanatmr-TMRInitTach5-LLR-005 308](#_Toc210646627)

[10.18.9.7.6 dauanatmr-TMRInitTach5-LLR-006 309](#_Toc210646628)

[10.18.9.7.7 dauanatmr-TMRInitTach5-LLR-007 309](#_Toc210646629)

[10.18.9.7.8 dauanatmr-TMRInitTach5-LLR-008 309](#_Toc210646630)

[10.18.9.7.9 dauanatmr-TMRInitTach5-LLR-009 309](#_Toc210646631)

[10.18.9.7.10 dauanatmr-TMRInitTach5-LLR-010 310](#_Toc210646632)

[10.18.9.7.11 dauanatmr-TMRInitTach5-LLR-011 310](#_Toc210646633)

[10.18.9.7.12 dauanatmr-TMRInitTach5-LLR-012 310](#_Toc210646634)

[10.18.10 TMRInitTach6 310](#_Toc210646635)

[10.18.10.1 Brief Description 311](#_Toc210646636)

[10.18.10.2 List of HLRs allocated 311](#_Toc210646637)

[10.18.10.3 3 List of global variables accessed and modified 311](#_Toc210646638)

[10.18.10.4 Parameter list (Input/Output) 311](#_Toc210646639)

[10.18.10.5 Return Value 311](#_Toc210646640)

[10.18.10.6 Other CSUs called by this CSU 311](#_Toc210646641)

[10.18.10.7 Description of list of LLRs allocated 312](#_Toc210646642)

[10.18.10.7.1 dauanatmr-TMRInitTach6-LLR-001 312](#_Toc210646643)

[10.18.10.7.2 dauanatmr-TMRInitTach6-LLR-002 312](#_Toc210646644)

[10.18.10.7.3 dauanatmr-TMRInitTach6-LLR-003 312](#_Toc210646645)

[10.18.10.7.4 dauanatmr-TMRInitTach6-LLR-004 313](#_Toc210646646)

[10.18.10.7.5 dauanatmr-TMRInitTach6-LLR-005 313](#_Toc210646647)

[10.18.10.7.6 dauanatmr-TMRInitTach6-LLR-006 313](#_Toc210646648)

[10.18.10.7.7 dauanatmr-TMRInitTach6-LLR-007 313](#_Toc210646649)

[10.18.10.7.8 dauanatmr-TMRInitTach6-LLR-008 314](#_Toc210646650)

[10.18.10.7.9 dauanatmr-TMRInitTach6-LLR-009 314](#_Toc210646651)

[10.18.10.7.10 dauanatmr-TMRInitTach6-LLR-010 314](#_Toc210646652)

[10.18.10.7.11 dauanatmr-TMRInitTach6-LLR-011 314](#_Toc210646653)

[10.18.10.7.12 dauanatmr-TMRInitTach6-LLR-012 315](#_Toc210646654)

[10.18.11 TMRInitPwmc1 315](#_Toc210646655)

[10.18.11.1 Brief Description 315](#_Toc210646656)

[10.18.11.2 List of HLRs allocated 315](#_Toc210646657)

[10.18.11.3 List of global variables accessed and modified 315](#_Toc210646658)

[10.18.11.4 Parameter list (Input/Output) 316](#_Toc210646659)

[10.18.11.5 Return Value 316](#_Toc210646660)

[10.18.11.6 Other CSUs called by this CSU 316](#_Toc210646661)

[10.18.11.7 Description of list of LLRs allocated 316](#_Toc210646662)

[10.18.11.7.1 dauanatmr-TMRInitPwmc1-LLR-001 316](#_Toc210646663)

[10.18.11.7.2 dauanatmr-TMRInitPwmc1-LLR-002 317](#_Toc210646664)

[10.18.11.7.3 dauanatmr-TMRInitPwmc1-LLR-003 317](#_Toc210646665)

[10.18.11.7.4 dauanatmr-TMRInitPwmc1-LLR-004 317](#_Toc210646666)

[10.18.11.7.5 dauanatmr-TMRInitPwmc1-LLR-005 317](#_Toc210646667)

[10.18.11.7.6 dauanatmr-TMRInitPwmc1-LLR-006 318](#_Toc210646668)

[10.18.11.7.7 dauanatmr-TMRInitPwmc1-LLR-007 318](#_Toc210646669)

[10.18.11.7.8 dauanatmr-TMRInitPwmc1-LLR-008 318](#_Toc210646670)

[10.18.11.7.9 dauanatmr-TMRInitPwmc1-LLR-009 318](#_Toc210646671)

[10.18.11.7.10 dauanatmr-TMRInitPwmc1-LLR-010 319](#_Toc210646672)

[10.18.11.7.11 dauanatmr-TMRInitPwmc1-LLR-011 319](#_Toc210646673)

[10.18.11.7.12 dauanatmr-TMRInitPwmc1-LLR-012 319](#_Toc210646674)

[10.18.12 TMRInitPwmc2 319](#_Toc210646675)

[10.18.12.1 Brief Description 320](#_Toc210646676)

[10.18.12.2 List of HLRs allocated 320](#_Toc210646677)

[10.18.12.3 List of global variables accessed and modified 320](#_Toc210646678)

[10.18.12.4 Parameter list (Input/Output) 320](#_Toc210646679)

[10.18.12.5 Return Value 320](#_Toc210646680)

[10.18.12.6 Other CSUs called by this CSU 320](#_Toc210646681)

[10.18.12.7 Description of list of LLRs allocated 321](#_Toc210646682)

[10.18.12.7.1 dauanatmr-TMRInitPwmc2-LLR-001 321](#_Toc210646683)

[10.18.12.7.2 dauanatmr-TMRInitPwmc2-LLR-002 321](#_Toc210646684)

[10.18.12.7.3 dauanatmr-TMRInitPwmc2-LLR-003 321](#_Toc210646685)

[10.18.12.7.4 dauanatmr-TMRInitPwmc2-LLR-004 322](#_Toc210646686)

[10.18.12.7.5 dauanatmr-TMRInitPwmc2-LLR-005 322](#_Toc210646687)

[10.18.12.7.6 dauanatmr-TMRInitPwmc2-LLR-006 322](#_Toc210646688)

[10.18.12.7.7 dauanatmr-TMRInitPwmc2-LLR-007 322](#_Toc210646689)

[10.18.12.7.8 dauanatmr-TMRInitPwmc2-LLR-008 323](#_Toc210646690)

[10.18.12.7.9 dauanatmr-TMRInitPwmc2-LLR-009 323](#_Toc210646691)

[10.18.12.7.10 dauanatmr-TMRInitPwmc2-LLR-010 323](#_Toc210646692)

[10.18.12.7.11 dauanatmr-TMRInitPwmc2-LLR-011 323](#_Toc210646693)

[10.18.12.7.12 dauanatmr-TMRInitPwmc2-LLR-012 324](#_Toc210646694)

[10.18.13 TMRInitPwmc3 324](#_Toc210646695)

[10.18.13.1 Brief Description 324](#_Toc210646696)

[10.18.13.2 List of HLRs allocated 324](#_Toc210646697)

[10.18.13.3 List of global variables accessed and modified 324](#_Toc210646698)

[10.18.13.4 Parameter list (Input/Output) 325](#_Toc210646699)

[10.18.13.5 Return Value 325](#_Toc210646700)

[10.18.13.6 Other CSUs called by this CSU 325](#_Toc210646701)

[10.18.13.7 Description of list of LLRs allocated 325](#_Toc210646702)

[10.18.13.7.1 dauanatmr-TMRInitPwmc3-LLR-001 325](#_Toc210646703)

[10.18.13.7.2 dauanatmr-TMRInitPwmc3-LLR-002 326](#_Toc210646704)

[10.18.13.7.3 dauanatmr-TMRInitPwmc3-LLR-003 326](#_Toc210646705)

[10.18.13.7.4 dauanatmr-TMRInitPwmc3-LLR-004 326](#_Toc210646706)

[10.18.13.7.5 dauanatmr-TMRInitPwmc3-LLR-005 326](#_Toc210646707)

[10.18.13.7.6 dauanatmr-TMRInitPwmc3-LLR-006 327](#_Toc210646708)

[10.18.13.7.7 dauanatmr-TMRInitPwmc3-LLR-007 327](#_Toc210646709)

[10.18.13.7.8 dauanatmr-TMRInitPwmc3-LLR-008 327](#_Toc210646710)

[10.18.13.7.9 dauanatmr-TMRInitPwmc3-LLR-009 327](#_Toc210646711)

[10.18.13.7.10 dauanatmr-TMRInitPwmc3-LLR-010 328](#_Toc210646712)

[10.18.13.7.11 dauanatmr-TMRInitPwmc3-LLR-011 328](#_Toc210646713)

[10.18.13.7.12 dauanatmr-TMRInitPwmc3-LLR-012 328](#_Toc210646714)

[10.18.14 TMRInitPwmc4 328](#_Toc210646715)

[10.18.14.1 Brief Description 329](#_Toc210646716)

[10.18.14.2 List of HLRs allocated 329](#_Toc210646717)

[10.18.14.3 List of global variables accessed and modified 329](#_Toc210646718)

[10.18.14.4 Parameter list (Input/Output) 329](#_Toc210646719)

[10.18.14.5 Return Value 329](#_Toc210646720)

[10.18.14.6 Other CSUs called by this CSU 329](#_Toc210646721)

[10.18.14.7 Description of list of LLRs allocated 330](#_Toc210646722)

[10.18.14.7.1 dauanatmr-TMRInitPwmc4-LLR-001 330](#_Toc210646723)

[10.18.14.7.2 dauanatmr-TMRInitPwmc4-LLR-002 330](#_Toc210646724)

[10.18.14.7.3 dauanatmr-TMRInitPwmc4-LLR-003 330](#_Toc210646725)

[10.18.14.7.4 dauanatmr-TMRInitPwmc4-LLR-004 331](#_Toc210646726)

[10.18.14.7.5 dauanatmr-TMRInitPwmc4-LLR-005 331](#_Toc210646727)

[10.18.14.7.6 dauanatmr-TMRInitPwmc4-LLR-006 331](#_Toc210646728)

[10.18.14.7.7 dauanatmr-TMRInitPwmc4-LLR-007 331](#_Toc210646729)

[10.18.14.7.8 dauanatmr-TMRInitPwmc4-LLR-008 332](#_Toc210646730)

[10.18.14.7.9 dauanatmr-TMRInitPwmc4-LLR-009 332](#_Toc210646731)

[10.18.14.7.10 dauanatmr-TMRInitPwmc4-LLR-010 332](#_Toc210646732)

[10.18.14.7.11 dauanatmr-TMRInitPwmc4-LLR-011 332](#_Toc210646733)

[10.18.14.7.12 dauanatmr-TMRInitPwmc4-LLR-012 333](#_Toc210646734)

[10.19 dauanaucos 333](#_Toc210646735)

[10.19.1 OsInit 333](#_Toc210646736)

[10.19.1.1 Brief Description 333](#_Toc210646737)

[10.19.1.2 List of HLRs allocated 333](#_Toc210646738)

[10.19.1.3 List of global variables accessed and modified 333](#_Toc210646739)

[10.19.1.4 Parameter list (Input/Output) 334](#_Toc210646740)

[10.19.1.5 Return Value 334](#_Toc210646741)

[10.19.1.6 Other CSUs called by this CSU 334](#_Toc210646742)

[10.19.1.7 Description of list of LLRs allocated 334](#_Toc210646743)

[10.19.1.7.1 dauanaucos-OsInit-LLR-001 334](#_Toc210646744)

[10.19.1.7.2 dauanaucos-OsInit-LLR-002 335](#_Toc210646745)

[10.19.2 OsTaskIdle 335](#_Toc210646746)

[10.19.2.1 Brief Description 335](#_Toc210646747)

[10.19.2.2 List of HLRs allocated 335](#_Toc210646748)

[10.19.2.3 List of global variables accessed and modified 336](#_Toc210646749)

[10.19.2.4 Parameter list (Input/Output) 336](#_Toc210646750)

[10.19.2.5 Return Value 336](#_Toc210646751)

[10.19.2.6 Other CSUs called by this CSU 336](#_Toc210646752)

[10.19.2.7 Description of list of LLRs allocated 336](#_Toc210646753)

[10.19.2.7.1 dauanaucos-OsTaskIdle-LLR-001 336](#_Toc210646754)

[10.19.3 OsStart 337](#_Toc210646755)

[10.19.3.1 Brief Description 337](#_Toc210646756)

[10.19.3.2 List of HLRs allocated 337](#_Toc210646757)

[10.19.3.3 List of global variables accessed and modified 337](#_Toc210646758)

[10.19.3.4 Parameter list (Input/Output) 337](#_Toc210646759)

[10.19.3.5 Return Value 338](#_Toc210646760)

[10.19.3.6 Other CSUs called by this CSU 338](#_Toc210646761)

[10.19.3.7 Description of list of LLRs allocated 338](#_Toc210646762)

[10.19.3.7.1 dauanaucos-OsStart-LLR-001 338](#_Toc210646763)

[10.19.3.7.2 dauanaucos-OsStart-LLR-002 338](#_Toc210646764)

[10.19.3.7.3 dauanaucos-OsStart-LLR-003 339](#_Toc210646765)

[10.19.3.7.4 dauanaucos-OsStart-LLR-004 339](#_Toc210646766)

[10.19.4 OsSched 339](#_Toc210646767)

[10.19.4.1 Brief Description 339](#_Toc210646768)

[10.19.4.2 List of HLRs allocated 339](#_Toc210646769)

[10.19.4.3 List of global variables accessed and modified 339](#_Toc210646770)

[10.19.4.4 Parameter list (Input/Output) 339](#_Toc210646771)

[10.19.4.5 Return Value 340](#_Toc210646772)

[10.19.4.6 Other CSUs called by this CSU 340](#_Toc210646773)

[10.19.4.7 Description of list of LLRs allocated 340](#_Toc210646774)

[10.19.4.7.1 dauanaucos-OsSched-LLR-001 340](#_Toc210646775)

[10.19.4.7.2 dauanaucos-OsSched-LLR-002 340](#_Toc210646776)

[10.19.4.7.3 dauanaucos-OsSched-LLR-003 341](#_Toc210646777)

[10.19.4.7.4 dauanaucos-OsSched-LLR-004 341](#_Toc210646778)

[10.19.4.7.5 dauanaucos-OsSched-LLR-005 341](#_Toc210646779)

[10.19.5 OsTcbInit 341](#_Toc210646780)

[10.19.5.1 Brief Description 341](#_Toc210646781)

[10.19.5.2 List of HLRs allocated 341](#_Toc210646782)

[10.19.5.3 List of global variables accessed and modified 342](#_Toc210646783)

[10.19.5.4 Parameter list (Input/Output) 342](#_Toc210646784)

[10.19.5.5 Return Value 342](#_Toc210646785)

[10.19.5.6 Other CSUs called by this CSU 342](#_Toc210646786)

[10.19.5.7 Description of list of LLRs allocated 342](#_Toc210646787)

[10.19.5.7.1 dauanaucos-OsTcbInit-LLR-001 342](#_Toc210646788)

[10.19.5.7.2 dauanaucos-OsTcbInit-LLR-002 343](#_Toc210646789)

[10.19.5.7.3 dauanaucos-OsTcbInit-LLR-003 343](#_Toc210646790)

[10.19.5.7.4 dauanaucos-OsTcbInit-LLR-004 344](#_Toc210646791)

[10.19.5.7.5 dauanaucos-OsTcbInit-LLR-005 344](#_Toc210646792)

[10.19.5.7.6 dauanaucos-OsTcbInit-LLR-006 344](#_Toc210646793)

[10.19.5.7.7 dauanaucos-OsTcbInit-LLR-007 344](#_Toc210646794)

[10.19.6 OsIntEnter 344](#_Toc210646795)

[10.19.6.1 Brief Description 344](#_Toc210646796)

[10.19.6.2 List of HLRs allocated 344](#_Toc210646797)

[10.19.6.3 List of global variables accessed and modified 345](#_Toc210646798)

[10.19.6.4 Parameter list (Input/Output) 345](#_Toc210646799)

[10.19.6.5 Return Value 345](#_Toc210646800)

[10.19.6.6 Other CSUs called by this CSU 345](#_Toc210646801)

[10.19.6.7 Description of list of LLRs allocated 345](#_Toc210646802)

[10.19.6.7.1 dauanaucos-OsIntEnter-LLR-001 345](#_Toc210646803)

[10.19.7 OsIntExit 346](#_Toc210646804)

[10.19.7.1 Brief Description 346](#_Toc210646805)

[10.19.7.2 List of HLRs allocated 346](#_Toc210646806)

[10.19.7.3 List of global variables accessed and modified 346](#_Toc210646807)

[10.19.7.4 Parameter list (Input/Output) 346](#_Toc210646808)

[10.19.7.5 Return Value 346](#_Toc210646809)

[10.19.7.6 Other CSUs called by this CSU 347](#_Toc210646810)

[10.19.7.7 Description of list of LLRs allocated 347](#_Toc210646811)

[10.19.7.7.1 dauanaucos-OsIntExit-LLR-001 347](#_Toc210646812)

[10.19.7.7.2 dauanaucos-OsIntExit-LLR-002 347](#_Toc210646813)

[10.19.7.7.3 dauanaucos-OsIntExit-LLR-003 347](#_Toc210646814)

[10.19.7.7.4 dauanaucos-OsIntExit-LLR-004 348](#_Toc210646815)

[10.19.7.7.5 dauanaucos-OsIntExit-LLR-005 348](#_Toc210646816)

[10.19.8 OsTimeTick 348](#_Toc210646817)

[10.19.8.1 Brief Description 348](#_Toc210646818)

[10.19.8.2 List of HLRs allocated 348](#_Toc210646819)

[10.19.8.3 List of global variables accessed and modified 349](#_Toc210646820)

[10.19.8.4 Parameter list (Input/Output) 349](#_Toc210646821)

[10.19.8.5 Return Value 349](#_Toc210646822)

[10.19.8.6 Other CSUs called by this CSU 349](#_Toc210646823)

[10.19.8.7 Description of list of LLRs allocated 349](#_Toc210646824)

[10.19.8.7.1 dauanaucos-OsTimeTick-LLR-001 349](#_Toc210646825)

[10.19.8.7.2 dauanaucos-OsTimeTick-LLR-002 350](#_Toc210646826)

[10.19.8.7.3 dauanaucos-OsTimeTick-LLR-003 350](#_Toc210646827)

[10.19.8.7.4 dauanaucos-OsTimeTick-LLR-004 350](#_Toc210646828)

[10.19.9 OsSemCreate 350](#_Toc210646829)

[10.19.9.1 Brief Description 351](#_Toc210646830)

[10.19.9.2 List of HLRs allocated 351](#_Toc210646831)

[10.19.9.4 Parameter list (Input/Output) 351](#_Toc210646832)

[10.19.9.5 Return Value 351](#_Toc210646833)

[10.19.9.6 Other CSUs called by this CSU 351](#_Toc210646834)

[10.19.9.7 Description of list of LLRs allocated 351](#_Toc210646835)

[10.19.9.7.1 dauanaucos-OsSemCreate-LLR-001 352](#_Toc210646836)

[10.19.9.7.2 dauanaucos-OsSemCreate-LLR-002 352](#_Toc210646837)

[10.19.9.7.3 dauanaucos-OsSemCreate-LLR-003 352](#_Toc210646838)

[10.19.9.7.4 dauanaucos-OsSemCreate-LLR-004 352](#_Toc210646839)

[10.19.9.7.5 dauanaucos-OsSemCreate-LLR-005 352](#_Toc210646840)

[10.19.10 OsSemPend 352](#_Toc210646841)

[10.19.10.1 Brief Description 353](#_Toc210646842)

[10.19.10.2 List of HLRs allocated 353](#_Toc210646843)

[10.19.10.3 List of global variables accessed and modified 353](#_Toc210646844)

[10.19.10.4 Parameter list (Input/Output) 353](#_Toc210646845)

[10.19.10.5 Return Value 353](#_Toc210646846)

[10.19.10.6 Other CSUs called by this CSU 354](#_Toc210646847)

[10.19.10.7 Description of list of LLRs allocated 354](#_Toc210646848)

[10.19.10.7.1 dauanaucos-OsSemPend-LLR-001 354](#_Toc210646849)

[10.19.10.7.2 dauanaucos-OsSemPend-LLR-002 354](#_Toc210646850)

[10.19.10.7.3 dauanaucos-OsSemPend-LLR-003 354](#_Toc210646851)

[10.19.11 OsSemPost 355](#_Toc210646852)

[10.19.11.1 Brief Description 355](#_Toc210646853)

[10.19.11.2 List of HLRs allocated 356](#_Toc210646854)

[10.19.11.3 List of global variables accessed and modified 356](#_Toc210646855)

[10.19.11.4 Parameter list (Input/Output) 356](#_Toc210646856)

[10.19.11.5 Return Value 356](#_Toc210646857)

[10.19.11.6 Other CSUs called by this CSU 356](#_Toc210646858)

[10.19.11.7 Description of list of LLRs allocated 356](#_Toc210646859)

[10.19.11.7.1 dauanaucos-OsSemPost-LLR-001 357](#_Toc210646860)

[10.19.11.7.2 dauanaucos-OsSemPost-LLR-002 357](#_Toc210646861)

[10.19.11.7.3 dauanaucos-OsSemPost-LLR-003 357](#_Toc210646862)

[10.19.11.7.4 dauanaucos-OsSemPost-LLR-004 358](#_Toc210646863)

[10.20 dauanautils 358](#_Toc210646864)

[10.20.1 UtilsCopy 358](#_Toc210646865)

[10.20.1.1 Brief Description 358](#_Toc210646866)

[10.20.1.2 List of HLRs allocated 358](#_Toc210646867)

[10.20.1.3 List of global variables accessed and modified 359](#_Toc210646868)

[10.20.1.4 Parameter list (Input/Output) 359](#_Toc210646869)

[10.20.1.5 Return Value 359](#_Toc210646870)

[10.20.1.6 Other CSUs called by this CSU 359](#_Toc210646871)

[10.20.1.7 Description of list of LLRs allocated 359](#_Toc210646872)

[10.20.1.7.1 dauanautils-UtilsCopy-LLR-001 359](#_Toc210646873)

[10.20.1.7.2 dauanautils-UtilsCopy-LLR-002 360](#_Toc210646874)

[10.21 dauanawdog 360](#_Toc210646875)

[10.21.1 WdogInit 360](#_Toc210646876)

[10.21.1.1 Brief Description 360](#_Toc210646877)

[10.21.1.2 List of HLRs allocated 360](#_Toc210646878)

[10.21.1.3 List of global variables accessed and modified 360](#_Toc210646879)

[10.21.1.4 Parameter list (Input/Output) 361](#_Toc210646880)

[10.21.1.5 Return Value 361](#_Toc210646881)

[10.21.1.6 Other CSUs called by this CSU 361](#_Toc210646882)

[10.21.1.7 Description of list of LLRs allocated 361](#_Toc210646883)

[10.21.1.7.1 dauanawdog-WdogInit-LLR-001 361](#_Toc210646884)

[10.21.1.7.2 dauanawdog-WdogInit-LLR-002 362](#_Toc210646885)

[10.21.1.7.3 dauanawdog-WdogInit-LLR-003 362](#_Toc210646886)

[10.21.1.7.4 dauanawdog-WdogInit-LLR-004 362](#_Toc210646887)

[10.21.1.7.5 dauanawdog-WdogInit-LLR-005 362](#_Toc210646888)

[10.21.1.7.6 dauanawdog-WdogInit-LLR-006 362](#_Toc210646889)

[10.21.1.7.7 dauanawdog-WdogInit-LLR-007 362](#_Toc210646890)

[10.21.1.7.8 dauanawdog-WdogInit-LLR-008 363](#_Toc210646891)

[10.21.2 WdogKickWatchDog 363](#_Toc210646892)

[10.21.2.1 Brief Description 363](#_Toc210646893)

[10.21.2.2 List of HLRs allocated 363](#_Toc210646894)

[10.21.2.3 List of global variables accessed and modified 363](#_Toc210646895)

[10.21.2.4 Parameter list (Input/Output) 363](#_Toc210646896)

[10.21.2.5 Return Value 364](#_Toc210646897)

[10.21.2.6 Other CSUs called by this CSU 364](#_Toc210646898)

[10.21.2.7 Description of list of LLRs allocated 364](#_Toc210646899)

[10.21.2.7.1 dauanawdog-WdogKickWatchDog-LLR-001 364](#_Toc210646900)

[10.22 dauanaxadc 364](#_Toc210646901)

[10.22.1 XADCRead 364](#_Toc210646902)

[10.22.1.1 Brief Description 364](#_Toc210646903)

[10.22.1.2 List of HLRs allocated 365](#_Toc210646904)

[10.22.1.3 List of global variables accessed and modified 365](#_Toc210646905)

[10.22.1.4 Parameter list (Input/Output) 365](#_Toc210646906)

[10.22.1.5 Return Value 365](#_Toc210646907)

[10.22.1.6 Other CSUs called by this CSU 365](#_Toc210646908)

[10.22.1.7 Description of list of LLRs allocated 365](#_Toc210646909)

[10.22.1.7.1 dauanaxadc-XADCRead-LLR-001 366](#_Toc210646910)

[10.22.2 SetMultiplexer 366](#_Toc210646911)

[10.22.2.1 Brief Description 366](#_Toc210646912)

[10.22.2.2 List of HLRs allocated 366](#_Toc210646913)

[10.22.2.3 List of global variables accessed and modified 366](#_Toc210646914)

[10.22.2.4 Parameter list (Input/Output) 366](#_Toc210646915)

[10.22.2.5 Return Value 366](#_Toc210646916)

[10.22.2.6 Other CSUs called by this CSU 367](#_Toc210646917)

[10.22.2.7 Description of list of LLRs allocated 367](#_Toc210646918)

[10.22.2.7.1 dauanaxadc-SetMultiplexer-LLR-001 367](#_Toc210646919)

[10.22.3 ReadExternalADC 367](#_Toc210646920)

[10.22.3.1 Brief Description 367](#_Toc210646921)

[10.22.3.2 List of HLRs allocated 368](#_Toc210646922)

[10.22.3.3 List of global variables accessed and modified 368](#_Toc210646923)

[10.22.3.4 Parameter list (Input/Output) 368](#_Toc210646924)

[10.22.3.5 Return Value 368](#_Toc210646925)

[10.22.3.6 Other CSUs called by this CSU 368](#_Toc210646926)

[10.22.3.7 Description of list of LLRs allocated 368](#_Toc210646927)

[10.22.3.7.1 dauanaxadc-ReadExternalADC-LLR-001 368](#_Toc210646928)

[10.22.3.7.2 dauanaxadc-ReadExternalADC-LLR-002 369](#_Toc210646929)

[10.22.3.7.3 dauanaxadc-ReadExternalADC-LLR-003 369](#_Toc210646930)

[10.22.3.7.4 dauanaxadc-ReadExternalADC-LLR-004 369](#_Toc210646931)

[10.22.3.7.5 dauanaxadc-ReadExternalADC-LLR-005 369](#_Toc210646932)

[10.22.4 XADCInit 370](#_Toc210646933)

[10.22.4.1 Brief Description 370](#_Toc210646934)

[10.22.4.2 List of HLRs allocated 370](#_Toc210646935)

[10.22.4.3 List of global variables accessed and modified 370](#_Toc210646936)

[10.22.4.4 Parameter list (Input/Output) 370](#_Toc210646937)

[10.22.4.5 Return Value 370](#_Toc210646938)

[10.22.4.6 Other CSUs called by this CSU 371](#_Toc210646939)

[10.22.4.7 Description of list of LLRs allocated 371](#_Toc210646940)

[10.22.4.7.1 dauanaxadc-XADCInit-LLR-001 371](#_Toc210646941)

[10.22.4.7.2 dauanaxadc-XADCInit-LLR-002 372](#_Toc210646942)

[10.22.4.7.3 dauanaxadc-XADCInit-LLR-003 372](#_Toc210646943)

[10.22.4.7.4 dauanaxadc-XADCInit-LLR-004 372](#_Toc210646944)

[10.23 dauanasspi 372](#_Toc210646945)

[10.23.1 SSPIclaimBusSPI2 372](#_Toc210646946)

[10.23.1.1 Brief Description 372](#_Toc210646947)

[10.23.1.2 List of HLRs allocated 373](#_Toc210646948)

[10.23.1.3 List of global variables accessed and modified 373](#_Toc210646949)

[10.23.1.4 Parameter list (Input/Output) 373](#_Toc210646950)

[10.23.1.5 Return Value 373](#_Toc210646951)

[10.23.1.6 Other CSUs called by this CSU 373](#_Toc210646952)

[10.23.1.7 Description of list of LLRs allocated 373](#_Toc210646953)

[10.23.1.7.1 dauanasspi-SSPIclaimBusSPI2-LLR-001 374](#_Toc210646954)

[10.23.2 SSPIReleaseBusSPI2 374](#_Toc210646955)

[10.23.2.1 Brief Description 374](#_Toc210646956)

[10.23.2.2 List of HLRs allocated 374](#_Toc210646957)

[10.23.2.3 List of global variables accessed and modified 374](#_Toc210646958)

[10.23.2.4 Parameter list (Input/Output) 374](#_Toc210646959)

[10.23.2.5 Return Value 375](#_Toc210646960)

[10.23.2.6 Other CSUs called by this CSU 375](#_Toc210646961)

[10.23.2.7 Description of list of LLRs allocated 375](#_Toc210646962)

[10.23.2.7.1 dauanasspi-SSPIReleaseBusSPI2-LLR-001 375](#_Toc210646963)

[10.23.3 SSPIWaitSPI2 375](#_Toc210646964)

[10.23.3.1 Brief Description 375](#_Toc210646965)

[10.23.3.2 List of HLRs allocated 376](#_Toc210646966)

[10.23.3.3 List of global variables accessed and modified 376](#_Toc210646967)

[10.23.3.4 Parameter list (Input/Output) 376](#_Toc210646968)

[10.23.3.5 Return Value 376](#_Toc210646969)

[10.23.3.6 Other CSUs called by this CSU 376](#_Toc210646970)

[10.23.3.7 Description of list of LLRs allocated 376](#_Toc210646971)

[10.23.3.7.1 dauanasspi-SSPIWaitSPI2-LLR-001 376](#_Toc210646972)

[10.23.4 SSPIFinishSPI2 377](#_Toc210646973)

[10.23.4.1 Brief Description 377](#_Toc210646974)

[10.23.4.2 List of HLRs allocated 377](#_Toc210646975)

[10.23.4.3 List of global variables accessed and modified 377](#_Toc210646976)

[10.23.4.4 Parameter list (Input/Output) 377](#_Toc210646977)

[10.23.4.5 Return Value 377](#_Toc210646978)

[10.23.4.6 Other CSUs called by this CSU 378](#_Toc210646979)

[10.23.4.7 Description of list of LLRs allocated 378](#_Toc210646980)

[10.23.4.7.1 dauanasspi-SSPIFinishSPI2-LLR-001 378](#_Toc210646981)

[10.23.5 SSPIInitSPI2 378](#_Toc210646982)

[10.23.5.1 Brief Description 378](#_Toc210646983)

[10.23.5.2 List of HLRs allocated 378](#_Toc210646984)

[10.23.5.3 List of global variables accessed and modified 379](#_Toc210646985)

[10.23.5.4 Parameter list (Input/Output) 379](#_Toc210646986)

[10.23.5.5 Return Value 379](#_Toc210646987)

[10.23.5.6 Other CSUs called by this CSU 379](#_Toc210646988)

[10.23.5.7 Description of list of LLRs allocated 379](#_Toc210646989)

[10.23.5.7.1 dauanasspi-SSPIInitSPI2-LLR-001 380](#_Toc210646990)

[10.23.5.7.2 dauanasspi-SSPIInitSPI2-LLR-002 380](#_Toc210646991)

[10.24 dauanapwmc 381](#_Toc210646992)

[10.24.1 PWMCRead 381](#_Toc210646993)

[10.24.1.1 Brief Description 381](#_Toc210646994)

[10.24.1.2 List of HLRs allocated 381](#_Toc210646995)

[10.24.1.3 List of global variables accessed and modified 381](#_Toc210646996)

[10.24.1.4 Parameter list (Input/Output) 382](#_Toc210646997)

[10.24.1.5 Return Value 382](#_Toc210646998)

[10.24.1.6 Other CSUs called by this CSU 382](#_Toc210646999)

[10.24.1.7 Description of list of LLRs allocated 382](#_Toc210647000)

[10.24.1.7.1 dauanapwmc-PWMCRead-LLR-001 382](#_Toc210647001)

[10.24.1.7.2 dauanapwmc-PWMCRead-LLR-002 382](#_Toc210647002)

[10.24.1.7.3 dauanapwmc-PWMCRead-LLR-004 383](#_Toc210647003)

[10.24.1.7.4 dauanapwmc-PWMCRead-LLR-003 383](#_Toc210647004)

[10.25 dauanatach 383](#_Toc210647005)

[10.25.1 TACHRead 383](#_Toc210647006)

[10.25.1.1 Brief Description 383](#_Toc210647007)

[10.25.1.2 List of HLRs allocated 384](#_Toc210647008)

[10.25.1.3 List of global variables accessed and modified 384](#_Toc210647009)

[10.25.1.4 Parameter list (Input/Output) 384](#_Toc210647010)

[10.25.1.5 Return Value 384](#_Toc210647011)

[10.25.1.6 Other CSUs called by this CSU 384](#_Toc210647012)

[10.25.1.7 Description of list of LLRs allocated 384](#_Toc210647013)

[10.25.1.7.1 dauanatach-TACHRead-LLR-001 385](#_Toc210647014)

[10.25.1.7.2 dauanatach-TACHRead-LLR-002 385](#_Toc210647015)

[10.25.1.7.3 dauanatach-TACHRead-LLR-003 385](#_Toc210647016)

[10.25.2 TachDriver 385](#_Toc210647017)

[10.25.2.1 Brief Description 386](#_Toc210647018)

[10.25.2.2 List of HLRs allocated 386](#_Toc210647019)

[10.25.2.3 List of global variables accessed and modified 386](#_Toc210647020)

[10.25.2.4 Parameter list (Input/Output) 386](#_Toc210647021)

[10.25.2.5 Return Value 386](#_Toc210647022)

[10.25.2.6 Other CSUs called by this CSU 386](#_Toc210647023)

[10.25.2.7 Description of list of LLRs allocated 387](#_Toc210647024)

[10.25.2.7.1 dauanatach-TachDriver-LLR-001 387](#_Toc210647025)

[10.25.2.7.2 dauanatach-TachDriver-LLR-002 387](#_Toc210647026)

[10.25.2.7.3 dauanatach-TachDriver-LLR-003 387](#_Toc210647027)

[10.25.2.7.4 dauanatach-TachDriver-LLR-004 387](#_Toc210647028)

[10.25.2.7.5 dauanatach-TachDriver-LLR-005 388](#_Toc210647029)

[10.25.2.7.6 dauanatach-TachDriver-LLR-006 388](#_Toc210647030)

[10.25.2.7.7 dauanatach-TachDriver-LLR-007 389](#_Toc210647031)

[10.25.2.7.8 dauanatach-TachDriver-LLR-008 389](#_Toc210647032)

[10.25.2.7.9 dauanatach-TachDriver-LLR-009 389](#_Toc210647033)

[10.25.2.7.10 dauanatach-TachDriver-LLR-010 390](#_Toc210647034)

[10.25.2.7.11 dauanatach-TachDriver-LLR-011 390](#_Toc210647035)

[10.25.2.7.12 dauanatach-TachDriver-LLR-012 390](#_Toc210647036)

[10.25.2.7.13 dauanatach-TachDriver-LLR-013 391](#_Toc210647037)

[10.25.3 TACHInit 391](#_Toc210647038)

[10.25.3.1 Brief Description 391](#_Toc210647039)

[10.25.3.2 List of HLRs allocated 391](#_Toc210647040)

[10.25.3.3 List of global variables accessed and modified 391](#_Toc210647041)

[10.25.3.4 Parameter list (Input/Output) 392](#_Toc210647042)

[10.25.3.5 Return Value 392](#_Toc210647043)

[10.25.3.6 Other CSUs called by this CSU 392](#_Toc210647044)

[10.25.3.7 Description of list of LLRs allocated 392](#_Toc210647045)

[10.25.3.7.1 dauanatach-TACHInit-LLR-001 392](#_Toc210647046)

[10.25.3.7.2 dauanatach-TACHInit-LLR-002 392](#_Toc210647047)

[10.25.3.7.3 dauanatach-TACHInit-LLR-003 393](#_Toc210647048)

[10.25.3.7.4 dauanatach-TACHInit-LLR-004 393](#_Toc210647049)

[10.25.3.7.5 dauanatach-TACHInit-LLR-005 393](#_Toc210647050)

[10.26 dauanaxdac 394](#_Toc210647051)

[10.26.1 XDACinit 394](#_Toc210647052)

[10.26.1.1 Brief description 394](#_Toc210647053)

[10.26.1.2 List of HLR's allocated 394](#_Toc210647054)

[10.26.1.3 List of global variables accessed and modified 394](#_Toc210647055)

[10.26.1.4 Parameter List (Input/Output) 394](#_Toc210647056)

[10.26.1.5 Return Value 395](#_Toc210647057)

[10.26.1.6 Other CSUs called by this CSU 395](#_Toc210647058)

[10.26.1.7 Description of list of LLRs allocated 395](#_Toc210647059)

[10.26.1.7.1 dauanaxdac-XDACinit-LLR-001 395](#_Toc210647060)

[10.26.2 XDACWrite 395](#_Toc210647061)

[10.26.2.1 Brief description 396](#_Toc210647062)

[10.26.2.2 List of HLR's allocated 396](#_Toc210647063)

[10.26.2.3 List of global variables accessed and modified 396](#_Toc210647064)

[10.26.2.4 Parameter List (Input/Output) 396](#_Toc210647065)

[10.26.2.5 Return value 396](#_Toc210647066)

[10.26.2.6 Other CSUS called by this CSU 396](#_Toc210647067)

[10.26.2.7 Description of list of LLRS allocated 397](#_Toc210647068)

[10.26.2.7.1 dauanaxdac-XDACWrite-LLR-001 397](#_Toc210647069)

[10.26.3 SerialIntrSPI2 397](#_Toc210647070)

[10.26.3.1 Brief Description 397](#_Toc210647071)

[10.26.3.2 List of HLR's allocated 397](#_Toc210647072)

[10.26.3.3 List of global variables accessed and modified 397](#_Toc210647073)

[10.26.3.4 Parameter List (Input/Output) 398](#_Toc210647074)

[10.26.3.5 Return Value 398](#_Toc210647075)

[10.26.3.6 Other CSUs called by this CSU 398](#_Toc210647076)

[10.26.3.7 Description of list of LLRs allocated 398](#_Toc210647077)

[10.26.3.7.1 dauanaxdac-SerialIntrSPI2-LLR-001 398](#_Toc210647078)

[10.26.3.7.2 dauanaxdac-SerialIntrSPI2-LLR-002 399](#_Toc210647079)

[10.26.3.7.3 dauanaxdac-SerialIntrSPI2-LLR-003 399](#_Toc210647080)

[10.26.3.7.4 dauanaxdac-SerialIntrSPI2-LLR-004 399](#_Toc210647081)

[10.26.4 WriteSPI2 399](#_Toc210647082)

[10.26.4.1 Brief description 400](#_Toc210647083)

[10.26.4.2 List of HLR's allocated 400](#_Toc210647084)

[10.26.4.3 List of global variables accessed and modified. 400](#_Toc210647085)

[10.26.4.4 Parameter List (Input/Output) 400](#_Toc210647086)

[10.26.4.5 Return values 400](#_Toc210647087)

[10.26.4.6 Other CSUs called by this CSU 400](#_Toc210647088)

[10.26.4.7 Description of the List of LLRs allocated 401](#_Toc210647089)

[10.26.4.7.1 dauanaxdac-WriteSPI2-LLR-001 401](#_Toc210647090)

[10.26.5 DoDacOutputs 401](#_Toc210647091)

[10.26.5.1 Brief description 401](#_Toc210647092)

[10.26.5.2 List of HLR's allocated 402](#_Toc210647093)

[10.26.5.3 List of global variables accessed and modified 402](#_Toc210647094)

[10.26.5.4 Parameter List (Input/Output) 402](#_Toc210647095)

[10.26.5.5 Return values 402](#_Toc210647096)

[10.26.5.6 Other CSUs called by this CSU 402](#_Toc210647097)

[10.26.5.7 Description of the List of LLRs allocated 402](#_Toc210647098)

[10.26.5.7.1 dauanaxdac-DoDacOutputs-LLR-001 403](#_Toc210647099)

[10.26.5.7.2 dauanaxdac-DoDacOutputs-LLR-002 403](#_Toc210647100)

[10.26.5.7.3 dauanaxdac-DoDacOutputs-LLR-003 403](#_Toc210647101)

[10.26.5.7.4 dauanaxdac-DoDacOutputs-LLR-004 404](#_Toc210647102)

[10.26.5.7.5 dauanaxdac-DoDacOutputs-LLR-005 404](#_Toc210647103)

[10.26.5.7.6 dauanaxdac-DoDacOutputs-LLR-007 404](#_Toc210647104)

[10.26.5.7.7 dauanaxdac-DoDacOutputs-LLR-006 405](#_Toc210647105)

[10.26.6 XdacTask 405](#_Toc210647106)

[10.26.6.1 Brief description 405](#_Toc210647107)

[10.26.6.2 List of HLR's allocated 405](#_Toc210647108)

[10.26.6.3 List of global variables accessed and modified 405](#_Toc210647109)

[10.26.6.4 Parameter List (Input/Output) 406](#_Toc210647110)

[10.26.6.5 Return values 406](#_Toc210647111)

[10.26.6.6 Other CSUs called by this CSU 406](#_Toc210647112)

[10.26.6.7 Description of the List of LLRs allocated 406](#_Toc210647113)

[10.26.6.7.1 dauanaxdac-XdacTask-LLR-001 406](#_Toc210647114)

[11 Software Low Level Requirements-Analog Module Library 407](#_Toc210647115)

[11.1 daulibmisc 407](#_Toc210647116)

[11.1.1 NvicPriorityGroupConfig 407](#_Toc210647117)

[11.1.1.1 Brief Description 407](#_Toc210647118)

[11.1.1.2 List of HLRs allocated 407](#_Toc210647119)

[11.1.1.3 List of global variables accessed and modified 407](#_Toc210647120)

[11.1.1.4 Parameter list (Input/Output) 407](#_Toc210647121)

[11.1.1.5 Return Value 408](#_Toc210647122)

[11.1.1.6 Other CSUs called by this CSU 408](#_Toc210647123)

[11.1.1.7 Description of list of LLRs allocated 408](#_Toc210647124)

[11.1.1.7.1 daulibmisc-NvicPriorityGroupConfig-LLR-001 408](#_Toc210647125)

[11.1.2 NvicInit 408](#_Toc210647126)

[11.1.2.1 Brief Description 408](#_Toc210647127)

[11.1.2.2 List of HLRs allocated 408](#_Toc210647128)

[11.1.2.3 List of global variables accessed and modified 409](#_Toc210647129)

[11.1.2.4 Parameter list (Input/Output) 409](#_Toc210647130)

[11.1.2.5 Return Value 409](#_Toc210647131)

[11.1.2.6 Other CSUs called by this CSU 409](#_Toc210647132)

[11.1.2.7 Description of list of LLRs allocated 409](#_Toc210647133)

[11.1.2.7.1 daulibmisc-NvicInit-LLR-001 409](#_Toc210647134)

[11.1.2.7.2 daulibmisc-NvicInit-LLR-002 410](#_Toc210647135)

[11.1.2.7.3 daulibmisc-NvicInit-LLR-003 410](#_Toc210647136)

[11.1.3 NvicSetVectorTable 410](#_Toc210647137)

[11.1.3.1 Brief Description 411](#_Toc210647138)

[11.1.3.2 List of HLRs allocated 411](#_Toc210647139)

[11.1.3.3 List of global variables accessed and modified 411](#_Toc210647140)

[11.1.3.4 Parameter list (Input/Output) 411](#_Toc210647141)

[11.1.3.5 Return Value 411](#_Toc210647142)

[11.1.3.6 Other CSUs called by this CSU 411](#_Toc210647143)

[11.1.3.7 Description of list of LLRs allocated 412](#_Toc210647144)

[11.1.3.7.1 daulibmisc-NvicSetVectorTable-LLR-001 412](#_Toc210647145)

[11.2 daulibstm32f4xxadc 412](#_Toc210647146)

[11.2.1 AdcInit 412](#_Toc210647147)

[11.2.1.1 Brief Description 412](#_Toc210647148)

[11.2.1.2 List of HLRs allocated 412](#_Toc210647149)

[11.2.1.3 List of global variables accessed and modified 413](#_Toc210647150)

[11.2.1.4 Parameter list (Input/Output) 413](#_Toc210647151)

[11.2.1.5 Return Value 413](#_Toc210647152)

[11.2.1.6 Other CSUs called by this CSU 413](#_Toc210647153)

[11.2.1.7 Description of list of LLRs allocated 413](#_Toc210647154)

[11.2.1.7.1 daulibstm32f4xxadc-AdcInit-LLR-001 413](#_Toc210647155)

[11.2.1.7.2 daulibstm32f4xxadc-AdcInit-LLR-002 414](#_Toc210647156)

[11.2.1.7.3 daulibstm32f4xxadc-AdcInit-LLR-003 414](#_Toc210647157)

[11.2.2 AdcCommonInit 415](#_Toc210647158)

[11.2.2.1 Brief Description 415](#_Toc210647159)

[11.2.2.2 List of HLRs allocated 415](#_Toc210647160)

[11.2.2.3 List of global variables accessed and modified 415](#_Toc210647161)

[11.2.2.4 Parameter list (Input/Output) 415](#_Toc210647162)

[11.2.2.5 Return Value 416](#_Toc210647163)

[11.2.2.6 Other CSUs called by this CSU 416](#_Toc210647164)

[11.2.2.7 Description of list of LLRs allocated 416](#_Toc210647165)

[11.2.2.7.1 daulibstm32f4xxadc-AdcCommonInit-LLR-001 416](#_Toc210647166)

[11.2.3 AdcCmd 416](#_Toc210647167)

[11.2.3.1 Brief Description 416](#_Toc210647168)

[11.2.3.2 List of HLRs allocated 417](#_Toc210647169)

[11.2.3.3 List of global variables accessed and modified 417](#_Toc210647170)

[11.2.3.4 Parameter list (Input/Output) 417](#_Toc210647171)

[11.2.3.5 Return Value 417](#_Toc210647172)

[11.2.3.6 Other CSUs called by this CSU 417](#_Toc210647173)

[11.2.3.7 Description of list of LLRs allocated 417](#_Toc210647174)

[11.2.3.7.1 daulibstm32f4xxadc-AdcCmd-LLR-001 418](#_Toc210647175)

[11.2.3.7.2 daulibstm32f4xxadc-AdcCmd-LLR-002 418](#_Toc210647176)

[11.2.4 AdcRegularChannelConfig 418](#_Toc210647177)

[11.2.4.1 Brief Description 418](#_Toc210647178)

[11.2.4.2 List of HLRs allocated 418](#_Toc210647179)

[11.2.4.3 List of global variables accessed and modified 418](#_Toc210647180)

[11.2.4.4 Parameter list (Input/Output) 419](#_Toc210647181)

[11.2.4.5 Return Value 419](#_Toc210647182)

[11.2.4.6 Other CSUs called by this CSU 419](#_Toc210647183)

[11.2.4.7 Description of list of LLRs allocated 419](#_Toc210647184)

[11.2.4.7.1 daulibstm32f4xxadc-AdcRegularChannelConfig-LLR-001 419](#_Toc210647185)

[11.2.4.7.2 daulibstm32f4xxadc-AdcRegularChannelConfig-LLR-002 420](#_Toc210647186)

[11.2.4.7.3 daulibstm32f4xxadc-AdcRegularChannelConfig-LLR-003 420](#_Toc210647187)

[11.2.4.7.4 daulibstm32f4xxadc-AdcRegularChannelConfig-LLR-004 421](#_Toc210647188)

[11.2.4.7.5 daulibstm32f4xxadc-AdcRegularChannelConfig-LLR-005 421](#_Toc210647189)

[11.2.5 AdcSoftwareStartConv 421](#_Toc210647190)

[11.2.5.1 Brief Description 422](#_Toc210647191)

[11.2.5.2 List of HLRs allocated 422](#_Toc210647192)

[11.2.5.3 List of global variables accessed and modified 422](#_Toc210647193)

[11.2.5.4 Parameter list (Input/Output) 422](#_Toc210647194)

[11.2.5.5 Return Value 422](#_Toc210647195)

[11.2.5.6 Other CSUs called by this CSU 422](#_Toc210647196)

[11.2.5.7 Description of list of LLRs allocated 423](#_Toc210647197)

[11.2.5.7.1 daulibstm32f4xxadc-AdcSoftwareStartConv-LLR-001 423](#_Toc210647198)

[11.2.6 AdcDmaCmd 423](#_Toc210647199)

[11.2.6.1 Brief Description 423](#_Toc210647200)

[11.2.6.2 List of HLRs allocated 423](#_Toc210647201)

[11.2.6.3 List of global variables accessed and modified 423](#_Toc210647202)

[11.2.6.4 Parameter list (Input/Output) 423](#_Toc210647203)

[11.2.6.5 Return Value 424](#_Toc210647204)

[11.2.6.6 Other CSUs called by this CSU 424](#_Toc210647205)

[11.2.6.7 Description of list of LLRs allocated 424](#_Toc210647206)

[11.2.6.7.1 daulibstm32f4xxadc-AdcDmaCmd-LLR-001 424](#_Toc210647207)

[11.2.6.7.2 daulibstm32f4xxadc-AdcDmaCmd-LLR-002 424](#_Toc210647208)

[11.2.7 AdcDmaReqAfterLastTransferCmd 424](#_Toc210647209)

[11.2.7.1 Brief Description 425](#_Toc210647210)

[11.2.7.2 List of HLRs allocated 425](#_Toc210647211)

[11.2.7.3 List of global variables accessed and modified 425](#_Toc210647212)

[11.2.7.4 Parameter list (Input/Output) 425](#_Toc210647213)

[11.2.7.5 Return Value 425](#_Toc210647214)

[11.2.7.6 Other CSUs called by this CSU 425](#_Toc210647215)

[11.2.7.7 Description of list of LLRs allocated 426](#_Toc210647216)

[11.2.7.7.1 daulibstm32f4xxadc-AdcDmaReqAfterLastTransferCmd-LLR-001 426](#_Toc210647217)

[11.2.7.7.2 daulibstm32f4xxadc-AdcDmaReqAfterLastTransferCmd-LLR-002 426](#_Toc210647218)

[11.3 daulibstm32f4xxcan 426](#_Toc210647219)

[11.3.1 CanDeInit 426](#_Toc210647220)

[11.3.1.1 Brief Description 426](#_Toc210647221)

[11.3.1.2 List of HLRs allocated 427](#_Toc210647222)

[11.3.1.3 List of global variables accessed and modified 427](#_Toc210647223)

[11.3.1.4 Parameter list (Input/Output) 427](#_Toc210647224)

[11.3.1.5 Return Value 427](#_Toc210647225)

[11.3.1.6 Other CSUs called by this CSU 427](#_Toc210647226)

[11.3.1.7 Description of list of LLRs allocated 427](#_Toc210647227)

[11.3.1.7.1 daulibstm32f4xxcan-CanDeInit-LLR-001 428](#_Toc210647228)

[11.3.1.7.2 daulibstm32f4xxcan-CanDeInit-LLR-002 428](#_Toc210647229)

[11.3.2 CanInit 428](#_Toc210647230)

[11.3.2.1 Brief Description 428](#_Toc210647231)

[11.3.2.2 List of HLRs allocated 428](#_Toc210647232)

[11.3.2.3 List of global variables accessed and modified 428](#_Toc210647233)

[11.3.2.4 Parameter list (Input/Output) 429](#_Toc210647234)

[11.3.2.5 Return Value 429](#_Toc210647235)

[11.3.2.6 Other CSUs called by this CSU 429](#_Toc210647236)

[11.3.2.7 Description of list of LLRs allocated 429](#_Toc210647237)

[11.3.2.7.1 daulibstm32f4xxcan-CanInit-LLR-001 429](#_Toc210647238)

[11.3.2.7.2 daulibstm32f4xxcan-CanInit-LLR-002 429](#_Toc210647239)

[11.3.2.7.3 daulibstm32f4xxcan-CanInit-LLR-003 430](#_Toc210647240)

[11.3.2.7.4 daulibstm32f4xxcan-CanInit-LLR-004 430](#_Toc210647241)

[11.3.2.7.5 daulibstm32f4xxcan-CanInit-LLR-005 430](#_Toc210647242)

[11.3.2.7.6 daulibstm32f4xxcan-CanInit-LLR-006 430](#_Toc210647243)

[11.3.2.7.7 daulibstm32f4xxcan-CanInit-LLR-007 431](#_Toc210647244)

[11.3.2.7.8 daulibstm32f4xxcan-CanInit-LLR-008 431](#_Toc210647245)

[11.3.2.7.9 daulibstm32f4xxcan-CanInit-LLR-009 431](#_Toc210647246)

[11.3.2.7.10 daulibstm32f4xxcan-CanInit-LLR-010 431](#_Toc210647247)

[11.3.2.7.11 daulibstm32f4xxcan-CanInit-LLR-011 431](#_Toc210647248)

[11.3.2.7.12 daulibstm32f4xxcan-CanInit-LLR-012 432](#_Toc210647249)

[11.3.2.7.13 daulibstm32f4xxcan-CanInit-LLR-013 432](#_Toc210647250)

[11.3.2.7.14 daulibstm32f4xxcan-CanInit-LLR-014 432](#_Toc210647251)

[11.3.2.7.15 daulibstm32f4xxcan-CanInit-LLR-015 432](#_Toc210647252)

[11.3.2.7.16 daulibstm32f4xxcan-CanInit-LLR-016 432](#_Toc210647253)

[11.3.2.7.17 daulibstm32f4xxcan-CanInit-LLR-017 433](#_Toc210647254)

[11.3.2.7.18 daulibstm32f4xxcan-CanInit-LLR-018 433](#_Toc210647255)

[11.3.2.7.19 daulibstm32f4xxcan-CanInit-LLR-019 433](#_Toc210647256)

[11.3.2.7.20 daulibstm32f4xxcan-CanInit-LLR-020 434](#_Toc210647257)

[11.3.3 CanFilterInit 434](#_Toc210647258)

[11.3.3.1 Brief Description 434](#_Toc210647259)

[11.3.3.2 List of HLRs allocated 434](#_Toc210647260)

[11.3.3.3 List of global variables accessed and modified 434](#_Toc210647261)

[11.3.3.4 Parameter list (Input/Output) 435](#_Toc210647262)

[11.3.3.5 Return Value 435](#_Toc210647263)

[11.3.3.6 Other CSUs called by this CSU 435](#_Toc210647264)

[11.3.3.7 Description of list of LLRs allocated 435](#_Toc210647265)

[11.3.3.7.1 daulibstm32f4xxcan-CanFilterInit-LLR-001 435](#_Toc210647266)

[11.3.3.7.2 daulibstm32f4xxcan-CanFilterInit-LLR-002 435](#_Toc210647267)

[11.3.3.7.3 daulibstm32f4xxcan-CanFilterInit-LLR-003 436](#_Toc210647268)

[11.3.3.7.4 daulibstm32f4xxcan-CanFilterInit-LLR-004 436](#_Toc210647269)

[11.3.3.7.5 daulibstm32f4xxcan-CanFilterInit-LLR-005 436](#_Toc210647270)

[11.3.3.7.6 daulibstm32f4xxcan-CanFilterInit-LLR-006 437](#_Toc210647271)

[11.3.3.7.7 daulibstm32f4xxcan-CanFilterInit-LLR-007 437](#_Toc210647272)

[11.3.3.7.8 daulibstm32f4xxcan-CanFilterInit-LLR-008 437](#_Toc210647273)

[11.3.3.7.9 daulibstm32f4xxcan-CanFilterInit-LLR-009 437](#_Toc210647274)

[11.3.3.7.10 daulibstm32f4xxcan-CanFilterInit-LLR-010 437](#_Toc210647275)

[11.3.4 CanTransmit 438](#_Toc210647276)

[11.3.4.1 Brief Description 438](#_Toc210647277)

[11.3.4.2 List of HLRs allocated 438](#_Toc210647278)

[11.3.4.3 List of global variables accessed and modified 438](#_Toc210647279)

[11.3.4.4 Parameter list (Input/Output) 438](#_Toc210647280)

[11.3.4.5 Return Value 438](#_Toc210647281)

[11.3.4.6 Other CSUs called by this CSU 439](#_Toc210647282)

[11.3.4.7 Description of list of LLRs allocated 439](#_Toc210647283)

[11.3.4.7.1 daulibstm32f4xxcan-CanTransmit-LLR-001 439](#_Toc210647284)

[11.3.4.7.2 daulibstm32f4xxcan-CanTransmit-LLR-002 439](#_Toc210647285)

[11.3.4.7.3 daulibstm32f4xxcan-CanTransmit-LLR-003 439](#_Toc210647286)

[11.3.4.7.4 daulibstm32f4xxcan-CanTransmit-LLR-004 440](#_Toc210647287)

[11.3.4.7.5 daulibstm32f4xxcan-CanTransmit-LLR-005 440](#_Toc210647288)

[11.3.4.7.6 daulibstm32f4xxcan-CanTransmit-LLR-006 440](#_Toc210647289)

[11.3.4.7.7 daulibstm32f4xxcan-CanTransmit-LLR-007 440](#_Toc210647290)

[11.3.4.7.8 daulibstm32f4xxcan-CanTransmit-LLR-008 441](#_Toc210647291)

[11.3.4.7.9 daulibstm32f4xxcan-CanTransmit-LLR-009 441](#_Toc210647292)

[11.3.4.7.10 daulibstm32f4xxcan-CanTransmit-LLR-010 441](#_Toc210647293)

[11.3.4.7.11 daulibstm32f4xxcan-CanTransmit-LLR-011 442](#_Toc210647294)

[11.3.5 CanReceive 442](#_Toc210647295)

[11.3.5.1 Brief Description 442](#_Toc210647296)

[11.3.5.2 List of HLRs allocated 442](#_Toc210647297)

[11.3.5.3 List of global variables accessed and modified 442](#_Toc210647298)

[11.3.5.4 Parameter list (Input/Output) 442](#_Toc210647299)

[11.3.5.5 Return Value 443](#_Toc210647300)

[11.3.5.6 Other CSUs called by this CSU 443](#_Toc210647301)

[11.3.5.7 Description of list of LLRs allocated 443](#_Toc210647302)

[11.3.5.7.1 daulibstm32f4xxcan-CanReceive-LLR-001 443](#_Toc210647303)

[11.3.5.7.2 daulibstm32f4xxcan-CanReceive-LLR-002 443](#_Toc210647304)

[11.3.5.7.3 daulibstm32f4xxcan-CanReceive-LLR-003 443](#_Toc210647305)

[11.3.5.7.4 daulibstm32f4xxcan-CanReceive-LLR-004 444](#_Toc210647306)

[11.3.5.7.5 daulibstm32f4xxcan-CanReceive-LLR-005 444](#_Toc210647307)

[11.3.5.7.6 daulibstm32f4xxcan-CanReceive-LLR-006 444](#_Toc210647308)

[11.3.5.7.7 daulibstm32f4xxcan-CanReceive-LLR-007 444](#_Toc210647309)

[11.3.5.7.8 daulibstm32f4xxcan-CanReceive-LLR-008 445](#_Toc210647310)

[11.3.5.7.9 daulibstm32f4xxcan-CanReceive-LLR-009 445](#_Toc210647311)

[11.3.6 CanItConfig 445](#_Toc210647312)

[11.3.6.1 Brief Description 445](#_Toc210647313)

[11.3.6.2 List of HLRs allocated 446](#_Toc210647314)

[11.3.6.3 List of global variables accessed and modified 446](#_Toc210647315)

[11.3.6.4 Parameter list (Input/Output) 446](#_Toc210647316)

[11.3.6.5 Return Value 446](#_Toc210647317)

[11.3.6.6 Other CSUs called by this CSU 446](#_Toc210647318)

[11.3.6.7 Description of list of LLRs allocated 446](#_Toc210647319)

[11.3.6.7.1 daulibstm32f4xxcan-CanItConfig-LLR-001 446](#_Toc210647320)

[11.3.6.7.2 daulibstm32f4xxcan-CanItConfig-LLR-002 447](#_Toc210647321)

[11.4 daulibstm32f4xxcrc 447](#_Toc210647322)

[11.4.1 CrcResetDr 447](#_Toc210647323)

[11.4.1.1 Brief Description 447](#_Toc210647324)

[11.4.1.2 List of HLRs allocated 447](#_Toc210647325)

[11.4.1.3 List of global variables accessed and modified 447](#_Toc210647326)

[11.4.1.4 Parameter list (Input/Output) 448](#_Toc210647327)

[11.4.1.5 Return Value 448](#_Toc210647328)

[11.4.1.6 Other CSUs called by this CSU 448](#_Toc210647329)

[11.4.1.7 Description of list of LLRs allocated 448](#_Toc210647330)

[11.4.1.7.1 daulibstm32f4xxcrc-CrcResetDr-LLR-001 448](#_Toc210647331)

[11.4.2 CrcCalcBlockCrc 448](#_Toc210647332)

[11.4.2.1 Brief Description 448](#_Toc210647333)

[11.4.2.2 List of HLRs allocated 449](#_Toc210647334)

[11.4.2.3 List of global variables accessed and modified 449](#_Toc210647335)

[11.4.2.4 Parameter list (Input/Output) 449](#_Toc210647336)

[11.4.2.5 Return Value 449](#_Toc210647337)

[11.4.2.6 Other CSUs called by this CSU 449](#_Toc210647338)

[11.4.2.7 Description of list of LLRs allocated 449](#_Toc210647339)

[11.4.2.7.1 daulibstm32f4xxcrc-CrcCalcBlockCrc-LLR-001 449](#_Toc210647340)

[11.5 daulibstm32f4xxdma 450](#_Toc210647341)

[11.5.1 DmaInit 450](#_Toc210647342)

[11.5.1.1 Brief Description 450](#_Toc210647343)

[11.5.1.2 List of HLRs allocated 450](#_Toc210647344)

[11.5.1.3 List of global variables accessed and modified 450](#_Toc210647345)

[11.5.1.4 Parameter list (Input/Output) 450](#_Toc210647346)

[11.5.1.5 Return Value 451](#_Toc210647347)

[11.5.1.6 Other CSUs called by this CSU 451](#_Toc210647348)

[11.5.1.7 Description of list of LLRs allocated 451](#_Toc210647349)

[11.5.1.7.1 daulibstm32f4xxdma-DmaInit-LLR-001 451](#_Toc210647350)

[11.5.1.7.2 daulibstm32f4xxdma-DmaInit-LLR-002 451](#_Toc210647351)

[11.5.1.7.3 daulibstm32f4xxdma-DmaInit-LLR-003 452](#_Toc210647352)

[11.5.2 DmaCmd 452](#_Toc210647353)

[11.5.2.1 Brief Description 452](#_Toc210647354)

[11.5.2.2 List of HLRs allocated 452](#_Toc210647355)

[11.5.2.3 List of global variables accessed and modified 453](#_Toc210647356)

[11.5.2.4 Parameter list (Input/Output) 453](#_Toc210647357)

[11.5.2.5 Return Value 453](#_Toc210647358)

[11.5.2.6 Other CSUs called by this CSU 453](#_Toc210647359)

[11.5.2.7 Description of list of LLRs allocated 453](#_Toc210647360)

[11.5.2.7.1 daulibstm32f4xxdma-DmaCmd-LLR-001 453](#_Toc210647361)

[11.5.2.7.2 daulibstm32f4xxdma-DmaCmd-LLR-002 454](#_Toc210647362)

[11.5.3 DmaClearItPendingBit 454](#_Toc210647363)

[11.5.3.1 Brief Description 454](#_Toc210647364)

[11.5.3.2 List of HLRs allocated 454](#_Toc210647365)

[11.5.3.3 List of global variables accessed and modified 454](#_Toc210647366)

[11.5.3.4 Parameter list (Input/Output) 454](#_Toc210647367)

[11.5.3.5 Return Value 455](#_Toc210647368)

[11.5.3.6 Other CSUs called by this CSU 455](#_Toc210647369)

[11.5.3.7 Description of list of LLRs allocated 455](#_Toc210647370)

[11.5.3.7.1 daulibstm32f4xxdma-DmaClearItPendingBit-LLR-001 455](#_Toc210647371)

[11.5.3.7.2 daulibstm32f4xxdma-DmaClearItPendingBit-LLR-002 455](#_Toc210647372)

[11.5.3.7.3 daulibstm32f4xxdma-DmaClearItPendingBit-LLR-003 455](#_Toc210647373)

[11.5.3.7.4 daulibstm32f4xxdma-DmaClearItPendingBit-LLR-004 455](#_Toc210647374)

[11.5.4 DmaItConfig 456](#_Toc210647375)

[11.5.4.1 Brief Description 456](#_Toc210647376)

[11.5.4.2 List of HLRs allocated 456](#_Toc210647377)

[11.5.4.3 List of global variables accessed and modified 456](#_Toc210647378)

[11.5.4.4 Parameter list (Input/Output) 456](#_Toc210647379)

[11.5.4.5 Return Value 457](#_Toc210647380)

[11.5.4.6 Other CSUs called by this CSU 457](#_Toc210647381)

[11.5.4.7 Description of list of LLRs allocated 457](#_Toc210647382)

[11.5.4.7.1 daulibstm32f4xxdma-DmaItConfig-LLR-001 457](#_Toc210647383)

[11.5.4.7.2 daulibstm32f4xxdma-DmaItConfig-LLR-002 457](#_Toc210647384)

[11.5.4.7.3 daulibstm32f4xxdma-DmaItConfig-LLR-003 457](#_Toc210647385)

[11.5.4.7.4 daulibstm32f4xxdma-DmaItConfig-LLR-004 458](#_Toc210647386)

[11.5.4.7.5 daulibstm32f4xxdma-DmaItConfig-LLR-005 458](#_Toc210647387)

[11.5.4.7.6 daulibstm32f4xxdma-DmaItConfig-LLR-006 458](#_Toc210647388)

[11.5.5 DmaDeInit 458](#_Toc210647389)

[11.5.5.1 Brief Description 458](#_Toc210647390)

[11.5.5.2 List of HLRs allocated 459](#_Toc210647391)

[11.5.5.3 List of global variables accessed and modified 459](#_Toc210647392)

[11.5.5.4 Parameter list (Input/Output) 459](#_Toc210647393)

[11.5.5.5 Return Value 459](#_Toc210647394)

[11.5.5.6 Other CSUs called by this CSU 459](#_Toc210647395)

[11.5.5.7 Description of list of LLRs allocated 459](#_Toc210647396)

[11.5.5.7.1 daulibstm32f4xxdma-DmaDeInit-LLR-001 460](#_Toc210647397)

[11.5.5.7.2 daulibstm32f4xxdma-DmaDeInit-LLR-002 460](#_Toc210647398)

[11.5.5.7.3 daulibstm32f4xxdma-DmaDeInit-LLR-003 460](#_Toc210647399)

[11.5.5.7.4 daulibstm32f4xxdma-DmaDeInit-LLR-004 460](#_Toc210647400)

[11.5.5.7.5 daulibstm32f4xxdma-DmaDeInit-LLR-005 461](#_Toc210647401)

[11.5.5.7.6 daulibstm32f4xxdma-DmaDeInit-LLR-006 461](#_Toc210647402)

[11.5.5.7.7 daulibstm32f4xxdma-DmaDeInit-LLR-007 461](#_Toc210647403)

[11.5.5.7.8 daulibstm32f4xxdma-DmaDeInit-LLR-008 461](#_Toc210647404)

[11.5.5.7.9 daulibstm32f4xxdma-DmaDeInit-LLR-009 461](#_Toc210647405)

[11.5.5.7.10 daulibstm32f4xxdma-DmaDeInit-LLR-010 462](#_Toc210647406)

[11.5.5.7.11 daulibstm32f4xxdma-DmaDeInit-LLR-011 462](#_Toc210647407)

[11.5.5.7.12 daulibstm32f4xxdma-DmaDeInit-LLR-012 462](#_Toc210647408)

[11.5.5.7.13 daulibstm32f4xxdma-DmaDeInit-LLR-013 462](#_Toc210647409)

[11.5.5.7.14 daulibstm32f4xxdma-DmaDeInit-LLR-014 462](#_Toc210647410)

[11.5.5.7.15 daulibstm32f4xxdma-DmaDeInit-LLR-015 463](#_Toc210647411)

[11.5.5.7.16 daulibstm32f4xxdma-DmaDeInit-LLR-016 463](#_Toc210647412)

[11.5.5.7.17 daulibstm32f4xxdma-DmaDeInit-LLR-017 463](#_Toc210647413)

[11.5.5.7.18 daulibstm32f4xxdma-DmaDeInit-LLR-018 463](#_Toc210647414)

[11.6 daulibstm32f4xxflash 463](#_Toc210647415)

[11.6.1 FlashSetLatency 464](#_Toc210647416)

[11.6.1.1 Brief Description 464](#_Toc210647417)

[11.6.1.2 List of HLRs allocated 464](#_Toc210647418)

[11.6.1.3 List of global variables accessed and modified 464](#_Toc210647419)

[11.6.1.4 Parameter list (Input/Output) 464](#_Toc210647420)

[11.6.1.5 Return Value 464](#_Toc210647421)

[11.6.1.6 Other CSUs called by this CSU 465](#_Toc210647422)

[11.6.1.7 Description of list of LLRs allocated 465](#_Toc210647423)

[11.6.1.7.1 daulibstm32f4xxflash-FlashSetLatency-LLR-001 465](#_Toc210647424)

[11.6.2 FlashPrefetchBufferCmd 465](#_Toc210647425)

[11.6.2.1 Brief Description 465](#_Toc210647426)

[11.6.2.2 List of HLRs allocated 465](#_Toc210647427)

[11.6.2.3 List of global variables accessed and modified 465](#_Toc210647428)

[11.6.2.4 Parameter list (Input/Output) 466](#_Toc210647429)

[11.6.2.5 Return Value 466](#_Toc210647430)

[11.6.2.6 Other CSUs called by this CSU 466](#_Toc210647431)

[11.6.2.7 Description of list of LLRs allocated 466](#_Toc210647432)

[11.6.2.7.1 daulibstm32f4xxflash-FlashPrefetchBufferCmd-LLR-001 466](#_Toc210647433)

[11.6.2.7.2 daulibstm32f4xxflash-FlashPrefetchBufferCmd-LLR-002 466](#_Toc210647434)

[11.6.3 FlashInstructionCacheCmd 467](#_Toc210647435)

[11.6.3.1 Brief Description 467](#_Toc210647436)

[11.6.3.2 List of HLRs allocated 467](#_Toc210647437)

[11.6.3.3 List of global variables accessed and modified 467](#_Toc210647438)

[11.6.3.4 Parameter list (Input/Output) 467](#_Toc210647439)

[11.6.3.5 Return Value 467](#_Toc210647440)

[11.6.3.6 Other CSUs called by this CSU 468](#_Toc210647441)

[11.6.3.7 Description of list of LLRs allocated 468](#_Toc210647442)

[11.6.3.7.1 daulibstm32f4xxflash-FlashInstructionCacheCmd-LLR-001 468](#_Toc210647443)

[11.6.3.7.2 daulibstm32f4xxflash-FlashInstructionCacheCmd-LLR-002 468](#_Toc210647444)

[11.6.4 FlashDataCacheCmd 468](#_Toc210647445)

[11.6.4.1 Brief Description 468](#_Toc210647446)

[11.6.4.2 List of HLRs allocated 468](#_Toc210647447)

[11.6.4.3 List of global variables accessed and modified 469](#_Toc210647448)

[11.6.4.4 Parameter list (Input/Output) 469](#_Toc210647449)

[11.6.4.5 Return Value 469](#_Toc210647450)

[11.6.4.6 Other CSUs called by this CSU 469](#_Toc210647451)

[11.6.4.7 Description of list of LLRs allocated 469](#_Toc210647452)

[11.6.4.7.1 daulibstm32f4xxflash-FlashDataCacheCmd-LLR-001 469](#_Toc210647453)

[11.6.4.7.2 daulibstm32f4xxflash-FlashDataCacheCmd-LLR-002 470](#_Toc210647454)

[11.7 daulibstm32f4xxfsmc 470](#_Toc210647455)

[11.7.1 FsmcNorSramInit 470](#_Toc210647456)

[11.7.1.1 Brief Description 470](#_Toc210647457)

[11.7.1.2 List of HLRs allocated 470](#_Toc210647458)

[11.7.1.3 List of global variables accessed and modified 470](#_Toc210647459)

[11.7.1.4 Parameter list (Input/Output) 471](#_Toc210647460)

[11.7.1.5 Return Value 471](#_Toc210647461)

[11.7.1.6 Other CSUs called by this CSU 471](#_Toc210647462)

[11.7.1.7 Description of list of LLRs allocated 471](#_Toc210647463)

[11.7.1.7.1 daulibstm32f4xxfsmc-FsmcNorSramInit-LLR-001 471](#_Toc210647464)

[11.7.1.7.2 daulibstm32f4xxfsmc-FsmcNorSramInit-LLR-002 472](#_Toc210647465)

[11.7.1.7.3 daulibstm32f4xxfsmc-FsmcNorSramInit-LLR-003 472](#_Toc210647466)

[11.7.1.7.4 daulibstm32f4xxfsmc-FsmcNorSramInit-LLR-004 473](#_Toc210647467)

[11.7.1.7.5 daulibstm32f4xxfsmc-FsmcNorSramInit-LLR-005 473](#_Toc210647468)

[11.7.2 FsmcNorSramCmd 473](#_Toc210647469)

[11.7.2.1 Brief Description 473](#_Toc210647470)

[11.7.2.2 List of HLRs allocated 474](#_Toc210647471)

[11.7.2.3 List of global variables accessed and modified 474](#_Toc210647472)

[11.7.2.4 Parameter list (Input/Output) 474](#_Toc210647473)

[11.7.2.5 Return Value 474](#_Toc210647474)

[11.7.2.6 Other CSUs called by this CSU 474](#_Toc210647475)

[11.7.2.7 Description of list of LLRs allocated 474](#_Toc210647476)

[11.7.2.7.1 daulibstm32f4xxfsmc-FsmcNorSramCmd-LLR-001 474](#_Toc210647477)

[11.7.2.7.2 daulibstm32f4xxfsmc-FsmcNorSramCmd-LLR-002 475](#_Toc210647478)

[11.8 daulibstm32f4xxgpio 475](#_Toc210647479)

[11.8.1 GpioInit 475](#_Toc210647480)

[11.8.1.1 Brief Description 475](#_Toc210647481)

[11.8.1.2 List of HLRs allocated 475](#_Toc210647482)

[11.8.1.3 List of global variables accessed and modified 476](#_Toc210647483)

[11.8.1.4 Parameter list (Input/Output) 476](#_Toc210647484)

[11.8.1.5 Return Value 476](#_Toc210647485)

[11.8.1.6 Other CSUs called by this CSU 476](#_Toc210647486)

[11.8.1.7 Description of list of LLRs allocated 476](#_Toc210647487)

[11.8.1.7.1 daulibstm32f4xxgpio-GpioInit-LLR-001 476](#_Toc210647488)

[11.8.2 GpioSetBits 477](#_Toc210647489)

[11.8.2.1 Brief Description 477](#_Toc210647490)

[11.8.2.2 List of HLRs allocated 477](#_Toc210647491)

[11.8.2.3 List of global variables accessed and modified 478](#_Toc210647492)

[11.8.2.4 Parameter list (Input/Output) 478](#_Toc210647493)

[11.8.2.5 Return Value 478](#_Toc210647494)

[11.8.2.6 Other CSUs called by this CSU 478](#_Toc210647495)

[11.8.2.7 Description of list of LLRs allocated 478](#_Toc210647496)

[11.8.2.7.1 daulibstm32f4xxgpio-GpioSetBits-LLR-001 478](#_Toc210647497)

[11.8.3 GpioResetBits 479](#_Toc210647498)

[11.8.3.1 Brief Description 479](#_Toc210647499)

[11.8.3.2 List of HLRs allocated 479](#_Toc210647500)

[11.8.3.3 List of global variables accessed and modified 479](#_Toc210647501)

[11.8.3.4 Parameter list (Input/Output) 479](#_Toc210647502)

[11.8.3.5 Return Value 479](#_Toc210647503)

[11.8.3.6 Other CSUs called by this CSU 479](#_Toc210647504)

[11.8.3.7 Description of list of LLRs allocated 480](#_Toc210647505)

[11.8.3.7.1 daulibstm32f4xxgpio-GpioResetBits-LLR-001 480](#_Toc210647506)

[11.8.4 GpioToggleBits 480](#_Toc210647507)

[11.8.4.1 Brief Description 480](#_Toc210647508)

[11.8.4.2 List of HLRs allocated 480](#_Toc210647509)

[11.8.4.3 List of global variables accessed and modified 480](#_Toc210647510)

[11.8.4.4 Parameter list (Input/Output) 480](#_Toc210647511)

[11.8.4.5 Return Value 481](#_Toc210647512)

[11.8.4.6 Other CSUs called by this CSU 481](#_Toc210647513)

[11.8.4.7 Description of list of LLRs allocated 481](#_Toc210647514)

[11.8.4.7.1 daulibstm32f4xxgpio-GpioToggleBits-LLR-001 481](#_Toc210647515)

[11.8.5 GpioPinAFConfig 481](#_Toc210647516)

[11.8.5.1 Brief Description 481](#_Toc210647517)

[11.8.5.2 List of HLRs allocated 482](#_Toc210647518)

[11.8.5.3 List of global variables accessed and modified 482](#_Toc210647519)

[11.8.5.4 Parameter list (Input/Output) 482](#_Toc210647520)

[11.8.5.5 Return Value 482](#_Toc210647521)

[11.8.5.6 Other CSUs called by this CSU 482](#_Toc210647522)

[11.8.5.7 Description of list of LLRs allocated 482](#_Toc210647523)

[11.8.5.7.1 daulibstm32f4xxgpio-GpioPinAFConfig-LLR-001 482](#_Toc210647524)

[11.9 daulibstm32f4xxiwdg 483](#_Toc210647525)

[11.9.1 IwdgWriteAccessCmd 483](#_Toc210647526)

[11.9.1.1 Brief Description 483](#_Toc210647527)

[11.9.1.2 List of HLRs allocated 483](#_Toc210647528)

[11.9.1.3 List of global variables accessed and modified 484](#_Toc210647529)

[11.9.1.4 Parameter list (Input/Output) 484](#_Toc210647530)

[11.9.1.5 Return Value 484](#_Toc210647531)

[11.9.1.6 Other CSUs called by this CSU 484](#_Toc210647532)

[11.9.1.7 Description of list of LLRs allocated 484](#_Toc210647533)

[11.9.1.7.1 daulibstm32f4xxiwdg-IwdgWriteAccessCmd-LLR-001 484](#_Toc210647534)

[11.9.2 IwdgSetPrescaler 485](#_Toc210647535)

[11.9.2.1 Brief Description 485](#_Toc210647536)

[11.9.2.2 List of HLRs allocated 485](#_Toc210647537)

[11.9.2.3 List of global variables accessed and modified 485](#_Toc210647538)

[11.9.2.4 Parameter list (Input/Output) 485](#_Toc210647539)

[11.9.2.5 Return Value 485](#_Toc210647540)

[11.9.2.6 Other CSUs called by this CSU 485](#_Toc210647541)

[11.9.2.7 Description of list of LLRs allocated 486](#_Toc210647542)

[11.9.2.7.1 daulibstm32f4xxiwdg-IwdgSetPrescaler-LLR-001 486](#_Toc210647543)

[11.9.3 IwdgSetReload 486](#_Toc210647544)

[11.9.3.1 Brief Description 486](#_Toc210647545)

[11.9.3.2 List of HLRs allocated 486](#_Toc210647546)

[11.9.3.3 List of global variables accessed and modified 486](#_Toc210647547)

[11.9.3.4 Parameter list (Input/Output) 487](#_Toc210647548)

[11.9.3.5 Return Value 487](#_Toc210647549)

[11.9.3.6 Other CSUs called by this CSU 487](#_Toc210647550)

[11.9.3.7 Description of list of LLRs allocated 487](#_Toc210647551)

[11.9.3.7.1 daulibstm32f4xxiwdg-IwdgSetReload-LLR-001 487](#_Toc210647552)

[11.9.4 IwdgReloadCounter 487](#_Toc210647553)

[11.9.4.1 Brief Description 487](#_Toc210647554)

[11.9.4.2 List of HLRs allocated 488](#_Toc210647555)

[11.9.4.3 List of global variables accessed and modified 488](#_Toc210647556)

[11.9.4.4 Parameter list (Input/Output) 488](#_Toc210647557)

[11.9.4.5 Return Value 488](#_Toc210647558)

[11.9.4.6 Other CSUs called by this CSU 488](#_Toc210647559)

[11.9.4.7 Description of list of LLRs allocated 488](#_Toc210647560)

[11.9.4.7.1 daulibstm32f4xxiwdg-IwdgReloadCounter-LLR-001 488](#_Toc210647561)

[11.9.5 IwdgEnable 489](#_Toc210647562)

[11.9.5.1 Brief Description 489](#_Toc210647563)

[11.9.5.2 List of HLRs allocated 489](#_Toc210647564)

[11.9.5.3 List of global variables accessed and modified 489](#_Toc210647565)

[11.9.5.4 Parameter list (Input/Output) 489](#_Toc210647566)

[11.9.5.5 Return Value 490](#_Toc210647567)

[11.9.5.6 Other CSUs called by this CSU 490](#_Toc210647568)

[11.9.5.7 Description of list of LLRs allocated 490](#_Toc210647569)

[11.9.5.7.1 daulibstm32f4xxiwdg-IwdgEnable-LLR-001 490](#_Toc210647570)

[11.10 daulibstm32f4xxpwr 490](#_Toc210647571)

[11.10.1 PwrMainRegulatorModeConfig 490](#_Toc210647572)

[11.10.1.1 Brief Description 490](#_Toc210647573)

[11.10.1.2 List of HLRs allocated 491](#_Toc210647574)

[11.10.1.3 List of global variables accessed and modified 491](#_Toc210647575)

[11.10.1.4 Parameter list (Input/Output) 491](#_Toc210647576)

[11.10.1.5 Return Value 491](#_Toc210647577)

[11.10.1.6 Other CSUs called by this CSU 491](#_Toc210647578)

[11.10.1.7 Description of list of LLRs allocated 491](#_Toc210647579)

[11.10.1.7.1 daulibstm32f4xxpwr-PwrMainRegulatorModeConfig-LLR-001 492](#_Toc210647580)

[11.10.1.7.2 daulibstm32f4xxpwr-PwrMainRegulatorModeConfig-LLR-002 492](#_Toc210647581)

[11.11 daulibstm32f4xxrcc 492](#_Toc210647582)

[11.11.1 RccDeInit 492](#_Toc210647583)

[11.11.1.1 Brief Description 492](#_Toc210647584)

[11.11.1.2 List of HLRs allocated 492](#_Toc210647585)

[11.11.1.3 List of global variables accessed and modified 493](#_Toc210647586)

[11.11.1.4 Parameter list (Input/Output) 493](#_Toc210647587)

[11.11.1.5 Return Value 493](#_Toc210647588)

[11.11.1.6 Other CSUs called by this CSU 493](#_Toc210647589)

[11.11.1.7 Description of list of LLRs allocated 493](#_Toc210647590)

[11.11.1.7.1 daulibstm32f4xxrcc-RccDeInit-LLR-001 493](#_Toc210647591)

[11.11.1.7.2 daulibstm32f4xxrcc-RccDeInit-LLR-002 494](#_Toc210647592)

[11.11.1.7.3 daulibstm32f4xxrcc-RccDeInit-LLR-003 494](#_Toc210647593)

[11.11.1.7.4 daulibstm32f4xxrcc-RccDeInit-LLR-004 494](#_Toc210647594)

[11.11.1.7.5 daulibstm32f4xxrcc-RccDeInit-LLR-005 494](#_Toc210647595)

[11.11.1.7.6 daulibstm32f4xxrcc-RccDeInit-LLR-006 494](#_Toc210647596)

[11.11.2 RccHseConfig 494](#_Toc210647597)

[11.11.2.1 Brief Description 495](#_Toc210647598)

[11.11.2.2 List of HLRs allocated 495](#_Toc210647599)

[11.11.2.3 List of global variables accessed and modified 495](#_Toc210647600)

[11.11.2.4 Parameter list (Input/Output) 495](#_Toc210647601)

[11.11.2.5 Return Value 495](#_Toc210647602)

[11.11.2.6 Other CSUs called by this CSU 495](#_Toc210647603)

[11.11.2.7 Description of list of LLRs allocated 496](#_Toc210647604)

[11.11.2.7.1 daulibstm32f4xxrcc-RccHseConfig-LLR-001 496](#_Toc210647605)

[11.11.3 RccWaitForHseStartUp 496](#_Toc210647606)

[11.11.3.1 Brief Description 496](#_Toc210647607)

[11.11.3.2 List of HLRs allocated 496](#_Toc210647608)

[11.11.3.3 List of global variables accessed and modified 496](#_Toc210647609)

[11.11.3.4 Parameter list (Input/Output) 496](#_Toc210647610)

[11.11.3.5 Return Value 497](#_Toc210647611)

[11.11.3.6 Other CSUs called by this CSU 497](#_Toc210647612)

[11.11.3.7 Description of list of LLRs allocated 497](#_Toc210647613)

[11.11.3.7.1 daulibstm32f4xxrcc-RccWaitForHseStartUp-LLR-001 497](#_Toc210647614)

[11.11.3.7.2 daulibstm32f4xxrcc-RccWaitForHseStartUp-LLR-002 497](#_Toc210647615)

[11.11.3.7.3 daulibstm32f4xxrcc-RccWaitForHseStartUp-LLR-003 497](#_Toc210647616)

[11.11.4 RccPllConfig 498](#_Toc210647617)

[11.11.4.1 Brief Description 498](#_Toc210647618)

[11.11.4.2 List of HLRs allocated 498](#_Toc210647619)

[11.11.4.3 List of global variables accessed and modified 498](#_Toc210647620)

[11.11.4.4 Parameter list (Input/Output) 498](#_Toc210647621)

[11.11.4.5 Return Value 499](#_Toc210647622)

[11.11.4.6 Other CSUs called by this CSU 499](#_Toc210647623)

[11.11.4.7 Description of list of LLRs allocated 499](#_Toc210647624)

[11.11.4.7.1 daulibstm32f4xxrcc-RccPllConfig-LLR-001 499](#_Toc210647625)

[11.11.5 RccPllCmd 499](#_Toc210647626)

[11.11.5.1 Brief Description 499](#_Toc210647627)

[11.11.5.2 List of HLRs allocated 499](#_Toc210647628)

[11.11.5.3 List of global variables accessed and modified 500](#_Toc210647629)

[11.11.5.4 Parameter list (Input/Output) 500](#_Toc210647630)

[11.11.5.5 Return Value 500](#_Toc210647631)

[11.11.5.6 Other CSUs called by this CSU 500](#_Toc210647632)

[11.11.5.7 Description of list of LLRs allocated 500](#_Toc210647633)

[11.11.5.7.1 daulibstm32f4xxrcc-RccPllCmd-LLR-001 500](#_Toc210647634)

[11.11.6 RccSysClkConfig 501](#_Toc210647635)

[11.11.6.1 Brief Description 501](#_Toc210647636)

[11.11.6.2 List of HLRs allocated 501](#_Toc210647637)

[11.11.6.3 List of global variables accessed and modified 501](#_Toc210647638)

[11.11.6.4 Parameter list (Input/Output) 501](#_Toc210647639)

[11.11.6.5 Return Value 501](#_Toc210647640)

[11.11.6.6 Other CSUs called by this CSU 501](#_Toc210647641)

[11.11.6.7 Description of list of LLRs allocated 502](#_Toc210647642)

[11.11.6.7.1 daulibstm32f4xxrcc-RccSysClkConfig-LLR-001 502](#_Toc210647643)

[11.11.7 RccGetSysClkSource 502](#_Toc210647644)

[11.11.7.1 Brief Description 502](#_Toc210647645)

[11.11.7.2 List of HLRs allocated 502](#_Toc210647646)

[11.11.7.3 List of global variables accessed and modified 502](#_Toc210647647)

[11.11.7.4 Parameter list (Input/Output) 503](#_Toc210647648)

[11.11.7.5 Return Value 503](#_Toc210647649)

[11.11.7.6 Other CSUs called by this CSU 503](#_Toc210647650)

[11.11.7.7 Description of list of LLRs allocated 503](#_Toc210647651)

[11.11.7.7.1 daulibstm32f4xxrcc-RccGetSysClkSource-LLR-001 503](#_Toc210647652)

[11.11.8 RccHclkConfig 503](#_Toc210647653)

[11.11.8.1 Brief Description 504](#_Toc210647654)

[11.11.8.2 List of HLRs allocated 504](#_Toc210647655)

[11.11.8.3 List of global variables accessed and modified 504](#_Toc210647656)

[11.11.8.4 Parameter list (Input/Output) 504](#_Toc210647657)

[11.11.8.5 Return Value 504](#_Toc210647658)

[11.11.8.6 Other CSUs called by this CSU 504](#_Toc210647659)

[11.11.8.7 Description of list of LLRs allocated 505](#_Toc210647660)

[11.11.8.7.1 daulibstm32f4xxrcc-RccHclkConfig-LLR-001 505](#_Toc210647661)

[11.11.9 RccPclk1Config 505](#_Toc210647662)

[11.11.9.1 Brief Description 505](#_Toc210647663)

[11.11.9.2 List of HLRs allocated 505](#_Toc210647664)

[11.11.9.3 List of global variables accessed and modified 505](#_Toc210647665)

[11.11.9.4 Parameter list (Input/Output) 506](#_Toc210647666)

[11.11.9.5 Return Value 506](#_Toc210647667)

[11.11.9.6 Other CSUs called by this CSU 506](#_Toc210647668)

[11.11.9.7 Description of list of LLRs allocated 506](#_Toc210647669)

[11.11.9.7.1 daulibstm32f4xxrcc-RccPclk1Config-LLR-001 506](#_Toc210647670)

[11.11.10 RccPclk2Config 506](#_Toc210647671)

[11.11.10.1 Brief Description 507](#_Toc210647672)

[11.11.10.2 List of HLRs allocated 507](#_Toc210647673)

[11.11.10.3 List of global variables accessed and modified 507](#_Toc210647674)

[11.11.10.4 Parameter list (Input/Output) 507](#_Toc210647675)

[11.11.10.5 Return Value 507](#_Toc210647676)

[11.11.10.6 Other CSUs called by this CSU 507](#_Toc210647677)

[11.11.10.7 Description of list of LLRs allocated 507](#_Toc210647678)

[11.11.10.7.1 daulibstm32f4xxrcc-RccPclk2Config-LLR-001 508](#_Toc210647679)

[11.11.11 RccAhb1PeriphClockCmd 508](#_Toc210647680)

[11.11.11.1 Brief Description 508](#_Toc210647681)

[11.11.11.2 List of HLRs allocated 508](#_Toc210647682)

[11.11.11.3 List of global variables accessed and modified 508](#_Toc210647683)

[11.11.11.4 Parameter list (Input/Output) 508](#_Toc210647684)

[11.11.11.5 Return Value 509](#_Toc210647685)

[11.11.11.6 Other CSUs called by this CSU 509](#_Toc210647686)

[11.11.11.7 Description of list of LLRs allocated 509](#_Toc210647687)

[11.11.11.7.1 daulibstm32f4xxrcc-RccAhb1PeriphClockCmd-LLR-001 509](#_Toc210647688)

[11.11.11.7.2 daulibstm32f4xxrcc-RccAhb1PeriphClockCmd-LLR-002 509](#_Toc210647689)

[11.11.12 RccAhb3PeriphClockCmd 510](#_Toc210647690)

[11.11.12.1 Brief Description 510](#_Toc210647691)

[11.11.12.2 List of HLRs allocated 510](#_Toc210647692)

[11.11.12.3 List of global variables accessed and modified 510](#_Toc210647693)

[11.11.12.4 Parameter list (Input/Output) 510](#_Toc210647694)

[11.11.12.5 Return Value 510](#_Toc210647695)

[11.11.12.6 Other CSUs called by this CSU 510](#_Toc210647696)

[11.11.12.7 Description of list of LLRs allocated 511](#_Toc210647697)

[11.11.12.7.1 daulibstm32f4xxrcc-RccAhb3PeriphClockCmd-LLR-001 511](#_Toc210647698)

[11.11.12.7.2 daulibstm32f4xxrcc-RccAhb3PeriphClockCmd-LLR-002 511](#_Toc210647699)

[11.11.13 RccApb1PeriphClockCmd 511](#_Toc210647700)

[11.11.13.1 Brief Description 511](#_Toc210647701)

[11.11.13.2 List of HLRs allocated 511](#_Toc210647702)

[11.11.13.3 List of global variables accessed and modified 512](#_Toc210647703)

[11.11.13.4 Parameter list (Input/Output) 512](#_Toc210647704)

[11.11.13.5 Return Value 512](#_Toc210647705)

[11.11.13.6 Other CSUs called by this CSU 512](#_Toc210647706)

[11.11.13.7 Description of list of LLRs allocated 512](#_Toc210647707)

[11.11.13.7.1 daulibstm32f4xxrcc-RccApb1PeriphClockCmd-LLR-001 513](#_Toc210647708)

[11.11.13.7.2 daulibstm32f4xxrcc-RccApb1PeriphClockCmd-LLR-002 513](#_Toc210647709)

[11.11.14 RccApb2PeriphClockCmd 513](#_Toc210647710)

[11.11.14.1 Brief Description 513](#_Toc210647711)

[11.11.14.2 List of HLRs allocated 513](#_Toc210647712)

[11.11.14.3 List of global variables accessed and modified 513](#_Toc210647713)

[11.11.14.4 Parameter list (Input/Output) 514](#_Toc210647714)

[11.11.14.5 Return Value 514](#_Toc210647715)

[11.11.14.6 Other CSUs called by this CSU 514](#_Toc210647716)

[11.11.14.7 Description of list of LLRs allocated 514](#_Toc210647717)

[11.11.14.7.1 daulibstm32f4xxrcc-RccApb2PeriphClockCmd-LLR-001 514](#_Toc210647718)

[11.11.14.7.2 daulibstm32f4xxrcc-RccApb2PeriphClockCmd-LLR-002 514](#_Toc210647719)

[11.11.15 RccGetFlagStatus 515](#_Toc210647720)

[11.11.15.1 Brief Description 515](#_Toc210647721)

[11.11.15.2 List of HLRs allocated 515](#_Toc210647722)

[11.11.15.3 List of global variables accessed and modified 515](#_Toc210647723)

[11.11.15.4 Parameter list (Input/Output) 515](#_Toc210647724)

[11.11.15.5 Return Value 515](#_Toc210647725)

[11.11.15.6 Other CSUs called by this CSU 516](#_Toc210647726)

[11.11.15.7 Description of list of LLRs allocated 516](#_Toc210647727)

[11.11.15.7.1 daulibstm32f4xxrcc-RccGetFlagStatus-LLR-001 516](#_Toc210647728)

[11.11.16 RccApb1PeriphResetCmd 516](#_Toc210647729)

[11.11.16.1 Brief Description 516](#_Toc210647730)

[11.11.16.2 List of HLRs allocated 517](#_Toc210647731)

[11.11.16.3 List of global variables accessed and modified 517](#_Toc210647732)

[11.11.16.4 Parameter list (Input/Output) 517](#_Toc210647733)

[11.11.16.5 Return Value 517](#_Toc210647734)

[11.11.16.6 Other CSUs called by this CSU 517](#_Toc210647735)

[11.11.16.7 Description of list of LLRs allocated 517](#_Toc210647736)

[11.11.16.7.1 daulibstm32f4xxrcc-RccApb1PeriphResetCmd-LLR-001 518](#_Toc210647737)

[11.11.16.7.2 daulibstm32f4xxrcc-RccApb1PeriphResetCmd-LLR-002 518](#_Toc210647738)

[11.12 daulibstm32f4xxtim 518](#_Toc210647739)

[11.12.1 TimPrescalerConfig 518](#_Toc210647740)

[11.12.1.1 Brief Description 518](#_Toc210647741)

[11.12.1.2 List of HLRs allocated 518](#_Toc210647742)

[11.12.1.3 List of global variables accessed and modified 519](#_Toc210647743)

[11.12.1.4 Parameter list (Input/Output) 519](#_Toc210647744)

[11.12.1.5 Return Value 519](#_Toc210647745)

[11.12.1.6 Other CSUs called by this CSU 519](#_Toc210647746)

[11.12.1.7 Description of list of LLRs allocated 519](#_Toc210647747)

[11.12.1.7.1 daulibstm32f4xxtim-TimPrescalerConfig-LLR-001 519](#_Toc210647748)

[11.12.1.7.2 daulibstm32f4xxtim-TimPrescalerConfig-LLR-002 520](#_Toc210647749)

[11.12.2 TimSetAutoReload 520](#_Toc210647750)

[11.12.2.1 Brief Description 520](#_Toc210647751)

[11.12.2.2 List of HLRs allocated 520](#_Toc210647752)

[11.12.2.3 List of global variables accessed and modified 520](#_Toc210647753)

[11.12.2.4 Parameter list (Input/Output) 520](#_Toc210647754)

[11.12.2.5 Return Value 521](#_Toc210647755)

[11.12.2.6 Other CSUs called by this CSU 521](#_Toc210647756)

[11.12.2.7 Description of list of LLRs allocated 521](#_Toc210647757)

[11.12.2.7.1 daulibstm32f4xxtim-TimSetAutoReload-LLR-001 521](#_Toc210647758)

[11.12.3 TimCmd 521](#_Toc210647759)

[11.12.3.1 Brief Description 521](#_Toc210647760)

[11.12.3.2 List of HLRs allocated 522](#_Toc210647761)

[11.12.3.3 List of global variables accessed and modified 522](#_Toc210647762)

[11.12.3.4 Parameter list (Input/Output) 522](#_Toc210647763)

[11.12.3.5 Return Value 522](#_Toc210647764)

[11.12.3.6 Other CSUs called by this CSU 522](#_Toc210647765)

[11.12.3.7 Description of list of LLRs allocated 522](#_Toc210647766)

[11.12.3.7.1 daulibstm32f4xxtim-TimCmd-LLR-001 522](#_Toc210647767)

[11.12.3.7.2 daulibstm32f4xxtim-TimCmd-LLR-002 523](#_Toc210647768)

[11.12.4 TimItConfig 523](#_Toc210647769)

[11.12.4.1 Brief Description 523](#_Toc210647770)

[11.12.4.2 List of HLRs allocated 523](#_Toc210647771)

[11.12.4.3 List of global variables accessed and modified 523](#_Toc210647772)

[11.12.4.4 Parameter list (Input/Output) 523](#_Toc210647773)

[11.12.4.5 Return Value 524](#_Toc210647774)

[11.12.4.6 Other CSUs called by this CSU 524](#_Toc210647775)

[11.12.4.7 Description of list of LLRs allocated 524](#_Toc210647776)

[11.12.4.7.1 daulibstm32f4xxtim-TimItConfig-LLR-001 524](#_Toc210647777)

[11.12.4.7.2 daulibstm32f4xxtim-TimItConfig-LLR-002 524](#_Toc210647778)

[11.12.5 TimClearITPendingBit 524](#_Toc210647779)

[11.12.5.1 Brief Description 525](#_Toc210647780)

[11.12.5.2 List of HLRs allocated 525](#_Toc210647781)

[11.12.5.3 List of global variables accessed and modified 525](#_Toc210647782)

[11.12.5.4 Parameter list (Input/Output) 525](#_Toc210647783)

[11.12.5.5 Return Value 525](#_Toc210647784)

[11.12.5.6 Other CSUs called by this CSU 525](#_Toc210647785)

[11.12.5.7 Description of list of LLRs allocated 526](#_Toc210647786)

[11.12.5.7.1 daulibstm32f4xxtim-TimClearITPendingBit-LLR-001 526](#_Toc210647787)

[11.12.6 TimICInit 526](#_Toc210647788)

[11.12.6.1 Brief Description 526](#_Toc210647789)

[11.12.6.2 List of HLRs allocated 526](#_Toc210647790)

[11.12.6.3 List of global variables accessed and modified 526](#_Toc210647791)

[11.12.6.4 Parameter list (Input/Output) 527](#_Toc210647792)

[11.12.6.5 Return Value 527](#_Toc210647793)

[11.12.6.6 Other CSUs called by this CSU 527](#_Toc210647794)

[11.12.6.7 Description of list of LLRs allocated 527](#_Toc210647795)

[11.12.6.7.1 daulibstm32f4xxtim-TimICInit-LLR-001 527](#_Toc210647796)

[11.12.6.7.2 daulibstm32f4xxtim-TimICInit-LLR-002 528](#_Toc210647797)

[11.12.6.7.3 daulibstm32f4xxtim-TimICInit-LLR-003 528](#_Toc210647798)

[11.12.6.7.4 daulibstm32f4xxtim-TimICInit-LLR-004 528](#_Toc210647799)

[11.12.7 TimSetIC1Prescaler 528](#_Toc210647800)

[11.12.7.1 Brief Description 529](#_Toc210647801)

[11.12.7.2 List of HLRs allocated 529](#_Toc210647802)

[11.12.7.3 List of global variables accessed and modified 529](#_Toc210647803)

[11.12.7.4 Parameter list (Input/Output) 529](#_Toc210647804)

[11.12.7.5 Return Value 529](#_Toc210647805)

[11.12.7.6 Other CSUs called by this CSU 529](#_Toc210647806)

[11.12.7.7 Description of list of LLRs allocated 530](#_Toc210647807)

[11.12.7.7.1 daulibstm32f4xxtim-TimSetIC1Prescaler-LLR-001 530](#_Toc210647808)

[11.12.7.7.2 daulibstm32f4xxtim-TimSetIC1Prescaler-LLR-002 530](#_Toc210647809)

[11.12.8 TimSetIC2Prescaler 530](#_Toc210647810)

[11.12.8.1 Brief Description 530](#_Toc210647811)

[11.12.8.2 List of HLRs allocated 530](#_Toc210647812)

[11.12.8.3 List of global variables accessed and modified 531](#_Toc210647813)

[11.12.8.4 Parameter list (Input/Output) 531](#_Toc210647814)

[11.12.8.5 Return Value 531](#_Toc210647815)

[11.12.8.6 Other CSUs called by this CSU 531](#_Toc210647816)

[11.12.8.7 Description of list of LLRs allocated 531](#_Toc210647817)

[11.12.8.7.1 daulibstm32f4xxtim-TimSetIC2Prescaler-LLR-001 531](#_Toc210647818)

[11.12.8.7.2 daulibstm32f4xxtim-TimSetIC2Prescaler-LLR-002 532](#_Toc210647819)

[11.12.9 TimSetIC3Prescaler 532](#_Toc210647820)

[11.12.9.1 Brief Description 532](#_Toc210647821)

[11.12.9.2 List of HLRs allocated 532](#_Toc210647822)

[11.12.9.3 List of global variables accessed and modified 532](#_Toc210647823)

[11.12.9.4 Parameter list (Input/Output) 533](#_Toc210647824)

[11.12.9.5 Return Value 533](#_Toc210647825)

[11.12.9.6 Other CSUs called by this CSU 533](#_Toc210647826)

[11.12.9.7 Description of list of LLRs allocated 533](#_Toc210647827)

[11.12.9.7.1 daulibstm32f4xxtim-TimSetIC3Prescaler-LLR-001 533](#_Toc210647828)

[11.12.9.7.2 daulibstm32f4xxtim-TimSetIC3Prescaler-LLR-002 533](#_Toc210647829)

[11.12.10 TimSetIC4Prescaler 534](#_Toc210647830)

[11.12.10.1 Brief Description 534](#_Toc210647831)

[11.12.10.2 List of HLRs allocated 534](#_Toc210647832)

[11.12.10.3 List of global variables accessed and modified 534](#_Toc210647833)

[11.12.10.4 Parameter list (Input/Output) 534](#_Toc210647834)

[11.12.10.5 Return Value 534](#_Toc210647835)

[11.12.10.6 Other CSUs called by this CSU 535](#_Toc210647836)

[11.12.10.7 Description of list of LLRs allocated 535](#_Toc210647837)

[11.12.10.7.1 daulibstm32f4xxtim-TimSetIC4Prescaler-LLR-001 535](#_Toc210647838)

[11.12.10.7.2 daulibstm32f4xxtim-TimSetIC4Prescaler-LLR-002 535](#_Toc210647839)

[11.12.11 TimDmaCmd 535](#_Toc210647840)

[11.12.11.1 Brief Description 535](#_Toc210647841)

[11.12.11.2 List of HLRs allocated 536](#_Toc210647842)

[11.12.11.3 List of global variables accessed and modified 536](#_Toc210647843)

[11.12.11.4 Parameter list (Input/Output) 536](#_Toc210647844)

[11.12.11.5 Return Value 536](#_Toc210647845)

[11.12.11.6 Other CSUs called by this CSU 536](#_Toc210647846)

[11.12.11.7 Description of list of LLRs allocated 536](#_Toc210647847)

[11.12.11.7.1 daulibstm32f4xxtim-TimDmaCmd-LLR-001 537](#_Toc210647848)

[11.12.11.7.2 daulibstm32f4xxtim-TimDmaCmd-LLR-002 537](#_Toc210647849)

[11.12.12 Ti1Config 537](#_Toc210647850)

[11.12.12.1 Brief Description 537](#_Toc210647851)

[11.12.12.2 List of HLRs allocated 537](#_Toc210647852)

[11.12.12.3 List of global variables accessed and modified 537](#_Toc210647853)

[11.12.12.4 Parameter list (Input/Output) 538](#_Toc210647854)

[11.12.12.5 Return Value 538](#_Toc210647855)

[11.12.12.6 Other CSUs called by this CSU 538](#_Toc210647856)

[11.12.12.7 Description of list of LLRs allocated 538](#_Toc210647857)

[11.12.12.7.1 daulibstm32f4xxtim-Ti1Config-LLR-001 538](#_Toc210647858)

[11.12.13 Ti2Config 539](#_Toc210647859)

[11.12.13.1 Brief Description 539](#_Toc210647860)

[11.12.13.2 List of HLRs allocated 539](#_Toc210647861)

[11.12.13.3 List of global variables accessed and modified 539](#_Toc210647862)

[11.12.13.4 Parameter list (Input/Output) 540](#_Toc210647863)

[11.12.13.5 Return Value 540](#_Toc210647864)

[11.12.13.6 Other CSUs called by this CSU 540](#_Toc210647865)

[11.12.13.7 Description of list of LLRs allocated 540](#_Toc210647866)

[11.12.13.7.1 daulibstm32f4xxtim-Ti2Config-LLR-001 540](#_Toc210647867)

[11.12.14 Ti3Config 541](#_Toc210647868)

[11.12.14.1 Brief Description 541](#_Toc210647869)

[11.12.14.2 List of HLRs allocated 541](#_Toc210647870)

[11.12.14.3 List of global variables accessed and modified 541](#_Toc210647871)

[11.12.14.4 Parameter list (Input/Output) 541](#_Toc210647872)

[11.12.14.5 Return Value 542](#_Toc210647873)

[11.12.14.6 Other CSUs called by this CSU 542](#_Toc210647874)

[11.12.14.7 Description of list of LLRs allocated 542](#_Toc210647875)

[11.12.14.7.1 daulibstm32f4xxtim-Ti3Config-LLR-001 542](#_Toc210647876)

[11.12.15 Ti4Config 542](#_Toc210647877)

[11.12.15.1 Brief Description 543](#_Toc210647878)

[11.12.15.2 List of HLRs allocated 543](#_Toc210647879)

[11.12.15.3 List of global variables accessed and modified 543](#_Toc210647880)

[11.12.15.4 Parameter list (Input/Output) 543](#_Toc210647881)

[11.12.15.5 Return Value 543](#_Toc210647882)

[11.12.15.6 Other CSUs called by this CSU 544](#_Toc210647883)

[11.12.15.7 Description of list of LLRs allocated 544](#_Toc210647884)

[11.12.15.7.1 daulibstm32f4xxtim-Ti4Config-LLR-001 544](#_Toc210647885)

[11.12.16 TIMSelectInputTrigger 544](#_Toc210647886)

[11.12.16.1 Brief Description 544](#_Toc210647887)

[11.12.16.2 List of HLRs allocated 545](#_Toc210647888)

[11.12.16.3 List of global variables accessed and modified 545](#_Toc210647889)

[11.12.16.4 Parameter list (Input/Output) 545](#_Toc210647890)

[11.12.16.5 Return Value 545](#_Toc210647891)

[11.12.16.6 Other CSUs called by this CSU 545](#_Toc210647892)

[11.12.16.7 Description of list of LLRs allocated 546](#_Toc210647893)

[11.12.16.7.1 daulibstm32f4xxtim-TIMSelectInputTrigger-LLR-001 546](#_Toc210647894)

[11.12.17 TIMSelectSlaveMode 546](#_Toc210647895)

[11.12.17.1 Brief Description 546](#_Toc210647896)

[11.12.17.2 List of HLRs allocated 546](#_Toc210647897)

[11.12.17.3 List of global variables accessed and modified 546](#_Toc210647898)

[11.12.17.4 Parameter list (Input/Output) 547](#_Toc210647899)

[11.12.17.5 Return Value 547](#_Toc210647900)

[11.12.17.6 Other CSUs called by this CSU 547](#_Toc210647901)

[11.12.17.7 Description of list of LLRs allocated 547](#_Toc210647902)

[11.12.17.7.1 daulibstm32f4xxtim-TIMSelectSlaveMode-LLR-001 547](#_Toc210647903)

[11.12.18 TIMSelectMasterSlaveMode 547](#_Toc210647904)

[11.12.18.1 Brief Description 548](#_Toc210647905)

[11.12.18.2 List of HLRs allocated 548](#_Toc210647906)

[11.12.18.3 List of global variables accessed and modified 548](#_Toc210647907)

[11.12.18.4 Parameter list (Input/Output) 548](#_Toc210647908)

[11.12.18.5 Return Value 548](#_Toc210647909)

[11.12.18.6 Other CSUs called by this CSU 549](#_Toc210647910)

[11.12.18.7 Description of list of LLRs allocated 549](#_Toc210647911)

[11.12.18.7.1 daulibstm32f4xxtim-TIMSelectMasterSlaveMode-LLR-001 549](#_Toc210647912)

[11.12.19 TIMUpdateRequestConfig 549](#_Toc210647913)

[11.12.19.1 Brief Description 549](#_Toc210647914)

[11.12.19.2 List of HLRs allocated 549](#_Toc210647915)

[11.12.19.4 Parameter list (Input/Output) 550](#_Toc210647916)

[11.12.19.5 Return Value 550](#_Toc210647917)

[11.12.19.6 Other CSUs called by this CSU 550](#_Toc210647918)

[11.12.19.7 Description of list of LLRs allocated 550](#_Toc210647919)

[11.12.19.7.1 daulibstm32f4xxtim-TIMUpdateRequestConfig-LLR-001 550](#_Toc210647920)

[11.12.20 TIMGetCapture1 551](#_Toc210647921)

[11.12.20.1 Brief Description 551](#_Toc210647922)

[11.12.20.2 List of HLRs allocated 551](#_Toc210647923)

[11.12.20.3 List of global variables accessed and modified 551](#_Toc210647924)

[11.12.20.4 Parameter list (Input/Output) 551](#_Toc210647925)

[11.12.20.5 Return Value 551](#_Toc210647926)

[11.12.20.6 Other CSUs called by this CSU 552](#_Toc210647927)

[11.12.20.7 Description of list of LLRs allocated 552](#_Toc210647928)

[11.12.20.7.1 daulibstm32f4xxtim-TIMGetCapture1-LLR-001 552](#_Toc210647929)

[11.12.21 TIMGetCapture2 552](#_Toc210647930)

[11.12.21.1 Brief Description 552](#_Toc210647931)

[11.12.21.2 List of HLRs allocated 552](#_Toc210647932)

[11.12.21.3 List of global variables accessed and modified 553](#_Toc210647933)

[11.12.21.4 Parameter list (Input/Output) 553](#_Toc210647934)

[11.12.21.5 Return Value 553](#_Toc210647935)

[11.12.21.6 Other CSUs called by this CSU 553](#_Toc210647936)

[11.12.21.7 Description of list of LLRs allocated 553](#_Toc210647937)

[11.12.21.7.1 daulibstm32f4xxtim-TIMGetCapture2-LLR-001 553](#_Toc210647938)

[11.12.22 TIMGetFlagStatus 553](#_Toc210647939)

[11.12.22.1 Brief Description 554](#_Toc210647940)

[11.12.22.2 List of HLRs allocated 554](#_Toc210647941)

[11.12.22.3 List of global variables accessed and modified 554](#_Toc210647942)

[11.12.22.4 Parameter list (Input/Output) 554](#_Toc210647943)

[11.12.22.5 Return Value 554](#_Toc210647944)

[11.12.22.6 Other CSUs called by this CSU 554](#_Toc210647945)

[11.12.22.7 Description of list of LLRs allocated 555](#_Toc210647946)

[11.12.22.7.1 daulibstm32f4xxtim-TIMGetFlagStatus-LLR-001 555](#_Toc210647947)

[11.13 daulibstm32f4xxspi 555](#_Toc210647948)

[11.13.1 SPIInit 555](#_Toc210647949)

[11.13.1.1 Brief Description 555](#_Toc210647950)

[11.13.1.2 List of HLRs allocated 555](#_Toc210647951)

[11.13.1.3 List of global variables accessed and modified 556](#_Toc210647952)

[11.13.1.4 Parameter list (Input/Output) 556](#_Toc210647953)

[11.13.1.5 Return Value 556](#_Toc210647954)

[11.13.1.6 Other CSUs called by this CSU 556](#_Toc210647955)

[11.13.1.7 Description of list of LLRs allocated 557](#_Toc210647956)

[11.13.1.7.1 daulibstm32f4xxspi-SPIInit-LLR-001 557](#_Toc210647957)

[11.13.2 SPICmd 558](#_Toc210647958)

[11.13.2.1 Brief Description 558](#_Toc210647959)

[11.13.2.2 List of HLRs allocated 558](#_Toc210647960)

[11.13.2.3 List of global variables accessed and modified 558](#_Toc210647961)

[11.13.2.4 Parameter list (Input/Output) 559](#_Toc210647962)

[11.13.2.5 Return Value 559](#_Toc210647963)

[11.13.2.6 Other CSUs called by this CSU 559](#_Toc210647964)

[11.13.2.7 Description of list of LLRs allocated 559](#_Toc210647965)

[11.13.2.7.1 daulibstm32f4xxspi-SPICmd-LLR-001 559](#_Toc210647966)

[11.13.3 SPII2SReceiveData 560](#_Toc210647967)

[11.13.3.1 Brief Description 560](#_Toc210647968)

[11.13.3.2 List of HLRs allocated 560](#_Toc210647969)

[11.13.3.3 List of global variables accessed and modified 560](#_Toc210647970)

[11.13.3.4 Parameter list (Input/Output) 560](#_Toc210647971)

[11.13.3.5 Return Value 561](#_Toc210647972)

[11.13.3.6 Other CSUs called by this CSU 561](#_Toc210647973)

[11.13.3.7 Description of list of LLRs allocated 561](#_Toc210647974)

[11.13.3.7.1 daulibstm32f4xxspi-SPII2SReceiveData-LLR-001 561](#_Toc210647975)

[11.13.4 SPII2SSendData 561](#_Toc210647976)

[11.13.4.1 Brief Description 561](#_Toc210647977)

[11.13.4.2 List of HLRs allocated 562](#_Toc210647978)

[11.13.4.3 List of global variables accessed and modified 562](#_Toc210647979)

[11.13.4.4 Parameter list (Input/Output) 562](#_Toc210647980)

[11.13.4.5 Return Value 562](#_Toc210647981)

[11.13.4.6 Other CSUs called by this CSU 562](#_Toc210647982)

[11.13.4.7 Description of list of LLRs allocated 562](#_Toc210647983)

[11.13.4.7.1 daulibstm32f4xxspi-SPII2SSendData-LLR-001 563](#_Toc210647984)

[11.13.5 SPII2SITConfig 563](#_Toc210647985)

[11.13.5.1 Brief Description 563](#_Toc210647986)

[11.13.5.2 List of HLRs allocated 563](#_Toc210647987)

[11.13.5.3 List of global variables accessed and modified 563](#_Toc210647988)

[11.13.5.4 Parameter list (Input/Output) 563](#_Toc210647989)

[11.13.5.5 Return Value 564](#_Toc210647990)

[11.13.5.6 Other CSUs called by this CSU 564](#_Toc210647991)

[11.13.5.7 Description of list of LLRs allocated 564](#_Toc210647992)

[11.13.5.7.1 daulibstm32f4xxspi-SPII2SITConfig-LLR-001 564](#_Toc210647993)

[11.13.5.7.2 daulibstm32f4xxspi-SPII2SITConfig-LLR-002 565](#_Toc210647994)

[12 Software Low Level Requirements - MCD Software 565](#_Toc210647995)

[12.1 dauanasensor 565](#_Toc210647996)

[12.1.1 Brief Description 565](#_Toc210647997)

[12.1.2 List of HLRs allocated 565](#_Toc210647998)

[12.1.3 Description of list of LLRs allocated 565](#_Toc210647999)

[12.1.3.1 dauanasensor-LLR-001 566](#_Toc210648000)

[12.1.3.2 dauanasensor-LLR-002 566](#_Toc210648001)

[12.1.3.3 dauanasensor-LLR-003 581](#_Toc210648002)

[12.1.3.4 dauanasensor-LLR-004 581](#_Toc210648003)

[13 Appendix A : Data Dictionary 584](#_Toc210648004)

[14 Appendix B: Data Constants 584](#_Toc210648005)

**List of Tables**

[Table 1: Amendment Record](#_Toc158213879) 54

[Table 2: Reference document 75](#_Toc158213880)

[Table 3: List of Abbreviation 76](#_Toc158213881)

[Table 4: Analog Part number and CRC 131](#_Toc158213882)

[Table 5: Analog input channel configuration format 566](#_Toc158213883)

[Table 6: Cold Junction signal configuration format 581](#_Toc158213884)

**List of Figures**

None

**1 Amendment Record**

Table : Amendment Record

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version No.** | **Description of Amendment** | **Change**  **Request No.** | **Changed By** | **Release**  **Date** |
| 1.1 | Initial Release | N/A | Dhanalakshmi | 04-Jan-2022 |
| 1.2 | 1.Section 9 Traceability is Updated as part of self review comment.  2.Below Sections are updated based on LLD Review comments:  10.1.1.5, 10.1.2.4, 10.1.2.5, 10.1.12.1, 10.2.2.1, 10.3 dauanaapp(entire section numbers are updated), 10.4.3.3, 10.6, 10.7, 10.9.1.6, 10.10.9.1, 10.11.2.4, 10.15.1.1, 10.16.1, 10.17.2.1, 10.18.2.4  3.Below Requirement ID’s are updated based on LLD Review comments:  H398-LLD-ANA-FNC-24, H398-LLD-ANA-FNC-28, H398-LLD-ANA-FNC-43, H398-LLD-ANA-FNC-44, H398-LLD-ANA-FNC-45, H398-LLD-ANA-FNC-56, H398-LLD-ANA-FNC-69, H398-LLD-ANA-FNC-70, H398-LLD-ANA-FNC-71, H398-LLD-ANA-FNC-72, H398-LLD-ANA-FNC-85, H398-LLD-ANA-FNC-107, H398-LLD-ANA-FNC-139, H398-LLD-ANA-FNC-140, H398-LLD-ANA-FNC-143, H398-LLD-ANA-FNC-144, H398-LLD-ANA-FNC-158, H398-LLD-ANA-FNC-161, H398-LLD-ANA-FNC-162, H398-LLD-ANA-FNC-165, H398-LLD-ANA-FNC-166, H398-LLD-ANA-FNC-187, H398-LLD-ANA-FNC-188, H398-LLD-ANA-FNC-189, H398-LLD-ANA-FNC-190, H398-LLD-ANA-FNC-191, H398-LLD-ANA-FNC-192, H398-LLD-ANA-FNC-193, H398-LLD-ANA-FNC-341, H398-LLD-ANA-FNC-360, H398-LLD-ANA-FNC-370, H398-LLD-ANA-FNC-389, H398-LLD-ANA-FNC-421, H398-LLD-ANA-FNC-466, H398-LLD-ANA-FNC-504, H398-LLD-ANA-FNC-526, H398-LLD-ANA-FNC-528, H398-LLD-ANA-FNC-529, H398-LLD-ANA-FNC-530, H398-LLD-ANA-FNC-551, H398-LLD-ANA-FNC-555, H398-LLD-ANA-FNC-566, H398-LLD-ANA-FNC-660, H398-LLD-ANA-FNC-662, H398-LLD-ANA-FNC-663, H398-LLD-ANA-FNC-664, H398-LLD-ANA-FNC-665, H398-LLD-ANA-FNC-666, H398-LLD-ANA-FNC-704, H398-LLD-ANA-FNC-806, H398-LLD-ANA-FNC-807, H398-LLD-ANA-FNC-814, H398-LLD-ANA-FNC-815, H398-LLD-ANA-FNC-872, H398-LLD-ANA-FNC-937, H398-LLD-ANA-FNC-1006, H398-LLD-ANA-FNC-1007, H398-LLD-ANA-FNC-1008, H398-LLD-ANA-FNC-1088, H398-LLD-ANA-FNC-1480, H398-LLD-ANA-FNC-1481, H398-LLD-ANA-FNC-1482, H398-LLD-ANA-FNC-1492, H398-LLD-ANA-FNC-1614, H398-LLD-ANA-FNC-1615, H398-LLD-ANA-FNC-1616, H398-LLD-ANA-FNC-1617, H398-LLD-ANA-FNC-1618, H398-LLD-ANA-FNC-1648, H398-LLD-ANA-FNC-1666, H398-LLD-ANA-FNC-1675, H398-LLD-ANA-FNC-1684, H398-LLD-ANA-FNC-1693, H398-LLD-ANA-FNC-1807  4.Below sections are newly added as per the LLD Review commens:  10.3.1-AppGetA825RCI, 11.11.14-RccApb2PeriphClockCmd, 11.11.16-RccApb1PeriphResetCmd  5.Removed below requirement ID’s from LLD as per LLD review comments:  H398-LLD-ANA-FNC-231  6.Below sections are removed from LLD as per LLD Review comments:  11.5.2 DmaStructInit 11.8.2 GpioStructInit  11.11.11 RccGetClocksFreq 11.12.6 TimSelectOutputTrigger 11.11.15 RccAhb2PeriphClockCmd, 10.19.8 OsTimeDly  7.Removed ‘.c’ extenstions from section 12 as per LLD Review comments.  8.Data dictionary and Data Constants are updated as per LLD and code Review comments.  9.Document Name has been changed as part of self review comment.  10.Below requirement ID’s are updated based on Code Review comments:  H398-LLD-ANA-FNC-215, H398-LLD-ANA-FNC-216, H398-LLD-ANA-FNC-225, H398-LLD-ANA-FNC-226, H398-LLD-ANA-FNC-227, H398-LLD-ANA-FNC-228, H398-LLD-ANA-FNC-230, H398-LLD-ANA-FNC-232, H398-LLD-ANA-FNC-233, H398-LLD-ANA-FNC-246, H398-LLD-ANA-FNC-247, H398-LLD-ANA-FNC-248, H398-LLD-ANA-FNC-660, H398-LLD-ANA-FNC-661, H398-LLD-ANA-FNC-662, H398-LLD-ANA-FNC-663, H398-LLD-ANA-FNC-664, H398-LLD-ANA-FNC-665, H398-LLD-ANA-FNC-666, H398-LLD-ANA-FNC-801, H398-LLD-ANA-FNC-802, H398-LLD-ANA-FNC-803, H398-LLD-ANA-FNC-804, H398-LLD-ANA-FNC-805, H398-LLD-ANA-FNC-806, H398-LLD-ANA-FNC-807, H398-LLD-ANA-FNC-809, H398-LLD-ANA-FNC-810, H398-LLD-ANA-FNC-811, H398-LLD-ANA-FNC-812, H398-LLD-ANA-FNC-813, H398-LLD-ANA-FNC-814, H398-LLD-ANA-FNC-815, H398-LLD-ANA-FNC-1383, H398-LLD-ANA-FNC-1384. H398-LLD-ANA-FNC-1425, H398-LLD-ANA-FNC-1426, H398-LLD-ANA-FNC-1648, H398-LLD-ANA-FNC-1702, H398-LLD-ANA-FNC-1737, H398-LLD-ANA-FNC-1668, H398-LLD-ANA-FNC-1833, H398-LLD-ANA-FNC-1832, H398-LLD-ANA-FNC-1502, H398-LLD-ANA-FNC-1220, H398-LLD-ANA-FNC-650, H398-LLD-ANA-FNC-20, H398-LLD-ANA-FNC-22, H398-LLD-ANA-FNC-23, H398-LLD-ANA-FNC-24, H398-LLD-ANA-FNC-54, H398-LLD-ANA-FNC-56, H398-LLD-ANA-FNC-55, H398-LLD-ANA-FNC-82, H398-LLD-ANA-FNC-84, H398-LLD-ANA-FNC-85, H398-LLD-ANA-FNC-105, H398-LLD-ANA-FNC-106, H398-LLD-ANA-FNC-107, H398-LLD-ANA-FNC-260, H398-LLD-ANA-FNC-261  11.Below sections and Requirement ID’s are updated as part of self review:  10.10.3 HardFault(Section number), 11.5.5.7.3, 11.5.5.7.4, 11.2.4.7.2, H398-LLD-ANA-FNC-1637  12.Below sections are updated as per Code review comments:  10.11.1.4, 10.18, 10.22.4.1, 11.12, 10.10.9.4  13.changed H398-LLD-ANA-FNC-1077 to H398-LLD-ANA-DRQ-1077 and H398-LLD-ANA-FNC-1078 to H398-LLD-ANA-DRQ-1078 and Rationale is update for respective Requirement ID’s.  14.Below sections are added newly as per code review comments:  11.5.5.7.5, 11.5.5.7.6, 11.11.11.1 | 100014  100016 | Dhanalakshmi | 18-Jan-2022 |
| 1.3 | Updated to address QA comments  Following requirements are updated  H398-LLD-ANA-FNC-191  H398-LLD-ANA-FNC-801 to H398-LLD-ANA-FNC-808 | 100014 | Dhanalakshmi | 20-Jan-2022 |
| 1.4 | 1.Updated to address QA comments:  H398-LLD-ANA-FNC-85, H398-LLD-ANA-FNC-391, H398-LLD-ANA-FNC-733, H398-LLD-ANA-FNC-771, H398-LLD-ANA-FNC-1415 and H398-LLD-ANA-FNC-1416  2. DC updated as per QA review comments | 100023 | Dhanalakshmi | 21-Jan-2022 |
| 1.5 | Updated to address the DER audit observations:  dauanaapp-ReadChannelBIT-LLR-001 requirement ID changed from H398-LLD-ANA-FNC-214 to H398-LLD-ANA-FNC-1860  dauanaapp-ReadChannelBIT-LLR-002 requirement ID changed from H398-LLD-ANA-FNC-215 to H398-LLD-ANA-FNC-1861  dauanaapp-ReadChannelBIT-LLR-003 requirement ID changed from H398-LLD-ANA-FNC-216 to H398-LLD-ANA-FNC-1862  The following General sections are updated due to self-review:  4 References  12.1.1 Brief Description | 100041 | Dhanalakshmi D | 24-Feb-2022 |
| 1.6 | The following changes have been made due to Spec REV 3 to 4 changes and due to self-review:  Modified Functional requirements:  H398-LLD-ANA-FNC-24  H398-LLD-ANA-FNC-54  H398-LLD-ANA-FNC-187  H398-LLD-ANA-FNC-189  H398-LLD-ANA-FNC-190  H398-LLD-ANA-FNC-191  H398-LLD-ANA-FNC-192  H398-LLD-ANA-FNC-232  H398-LLD-ANA-FNC-233  H398-LLD-ANA-FNC-260  H398-LLD-ANA-FNC-261  H398-LLD-ANA-FNC-290  H398-LLD-ANA-FNC-291  H398-LLD-ANA-FNC-293  H398-LLD-ANA-FNC-309  H398-LLD-ANA-FNC-310  H398-LLD-ANA-FNC-417  H398-LLD-ANA-FNC-420  H398-LLD-ANA-FNC-465  H398-LLD-ANA-FNC-467  H398-LLD-ANA-FNC-481  H398-LLD-ANA-FNC-491  H398-LLD-ANA-FNC-504  H398-LLD-ANA-FNC-630  H398-LLD-ANA-FNC-660  H398-LLD-ANA-FNC-661  H398-LLD-ANA-FNC-781  H398-LLD-ANA-FNC-803  H398-LLD-ANA-FNC-804  H398-LLD-ANA-FNC-805  H398-LLD-ANA-FNC-806  H398-LLD-ANA-FNC-807  H398-LLD-ANA-FNC-808  H398-LLD-ANA-FNC-809  H398-LLD-ANA-FNC-810  H398-LLD-ANA-FNC-811  H398-LLD-ANA-FNC-812  H398-LLD-ANA-FNC-813  H398-LLD-ANA-FNC-814  H398-LLD-ANA-FNC-815  H398-LLD-ANA-FNC-816  H398-LLD-ANA-FNC-1189  H398-LLD-ANA-FNC-1373  H398-LLD-ANA-FNC-1863  H398-LLD-ANA-FNC-1864  H398-LLD-ANA-FNC-1865  H398-LLD-ANA-FNC-1866  H398-LLD-ANA-FNC-1867  H398-LLD-ANA-FNC-1868  H398-LLD-ANA-FNC-1869  H398-LLD-ANA-FNC-1870  H398-LLD-ANA-FNC-1871  H398-LLD-ANA-FNC-1872  H398-LLD-ANA-FNC-1873  H398-LLD-ANA-FNC-1874  H398-LLD-ANA-FNC-1875  H398-LLD-ANA-FNC-1876  H398-LLD-ANA-FNC-938  H398-LLD-ANA-DRQ-1077  H398-LLD-ANA-FNC-1159  H398-LLD-ANA-FNC-1831  H398-LLD-ANA-FNC-1832  New Functional requirements:  H398-LLD-ANA-FNC-1885  H398-LLD-ANA-FNC-1886  H398-LLD-ANA-FNC-1887  Deleted Functional requirements:  H398-LLD-ANA-FNC-259  H398-LLD-ANA-FNC-262  H398-LLD-ANA-FNC-292  H398-LLD-ANA-FNC-307  H398-LLD-ANA-FNC-423  H398-LLD-ANA-FNC-424  H398-LLD-ANA-FNC-433  H398-LLD-ANA-FNC-434  H398-LLD-ANA-FNC-435  New General requirements:  10.10.7 SpuriousInterrupt  10.10.7.1 Brief Description  10.10.7.2 List of HLRs allocated  10.10.7.3 List of global variables accessed and modified  10.10.7.4 Parameter list (Input/Output)  10.10.7.5 Return Value  10.10.7.6 Other CSUs called by this CSU  10.10.7.7 Description of list of LLRs allocated  Modified General requirements:  4 References  5 Acronyms and Definitions  10.2.1.4 Parameter list (Input/Output)  13 Appendix A : Data Dictionary  14 Appendix B: Data Constants | 100060 | Dhanalakshmi D | 24-March-2022 |
| 1.7 | Updated to address QA comments and due to self-review:  In section 13 Appendix A : Data Dictionary,  The constant data table for the Analog Module MCD Software has been modified. | 100060 | Dhanalakshmi D | 07-April-2022 |
| 1.8 | Updated as per PR 1000xx.  The following requirements have been modified:  H398-LLD-ANA-FNC-260  H398-LLD-ANA-FNC-801  H398-LLD-ANA-FNC-802  H398-LLD-ANA-FNC-803  H398-LLD-ANA-FNC-804  H398-LLD-ANA-FNC-805  H398-LLD-ANA-FNC-806  H398-LLD-ANA-FNC-807  H398-LLD-ANA-FNC-808  H398-LLD-ANA-FNC-809  H398-LLD-ANA-FNC-810  H398-LLD-ANA-FNC-811  H398-LLD-ANA-FNC-812  H398-LLD-ANA-FNC-813  H398-LLD-ANA-FNC-814  H398-LLD-ANA-FNC-815  H398-LLD-ANA-FNC-816  H398-LLD-ANA-FNC-1863  H398-LLD-ANA-FNC-1864  H398-LLD-ANA-FNC-1865  H398-LLD-ANA-FNC-1866  H398-LLD-ANA-FNC-1867  H398-LLD-ANA-FNC-1868  H398-LLD-ANA-FNC-1869  H398-LLD-ANA-FNC-1870  H398-LLD-ANA-FNC-1871  H398-LLD-ANA-FNC-1872  H398-LLD-ANA-FNC-1873  H398-LLD-ANA-FNC-1874  H398-LLD-ANA-FNC-1875  H398-LLD-ANA-FNC-1876 | 100083 | Dhanalakshmi D | 12-April-2022 |
| 1.9 | Updated to address review comments for PR 100083 and due to self-review.  The following requirements have been modified:  Functional requirement:  H398-LLD-ANA-FNC-1876  General requirement:  In section 14 Appendix B: Data Constants:  data constants for the Analog Module Application Software has been modified. | 100083 | Dhanalakshmi D | 12-April-2022 |
| 1.10 | The following requirements have been modified:  Functional requirement:  H398-LLD-ANA-FNC-947  H398-LLD-ANA-FNC-937  General requirement:  In section 14 Appendix B: Data Constants | 100108 | P AFREEN | 30-Nov-2022 |
| 2.0 | The following requirements have been modified:  Functional requirement:  H398-LLD-ANA-FNC-21  H398-LLD-ANA-FNC-25  H398-LLD-ANA-FNC-69  H398-LLD-ANA-FNC-70  H398-LLD-ANA-FNC-105  H398-LLD-ANA-FNC-106  H398-LLD-ANA-FNC-108  H398-LLD-ANA-FNC-177  H398-LLD-ANA-FNC-187  H398-LLD-ANA-FNC-191  H398-LLD-ANA-FNC-192  H398-LLD-ANA-FNC-1861  H398-LLD-ANA-FNC-228  H398-LLD-ANA-FNC-230  H398-LLD-ANA-FNC-246  H398-LLD-ANA-FNC-260  H398-LLD-ANA-FNC-261  H398-LLD-ANA-FNC-306  H398-LLD-ANA-FNC-308  H398-LLD-ANA-FNC-309  H398-LLD-ANA-FNC-320  H398-LLD-ANA-FNC-341  H398-LLD-ANA-FNC-360  H398-LLD-ANA-FNC-417  H398-LLD-ANA-FNC-491  H398-LLD-ANA-FNC-547  H398-LLD-ANA-FNC-549  H398-LLD-ANA-FNC-550  H398-LLD-ANA-FNC-552  H398-LLD-ANA-FNC-555  H398-LLD-ANA-FNC-650  H398-LLD-ANA-FNC-677  H398-LLD-ANA-FNC-704  H398-LLD-ANA-FNC-921  H398-LLD-ANA-FNC-926  H398-LLD-ANA-FNC-1189  H398-LLD-ANA-FNC-1190  H398-LLD-ANA-FNC-1191  H398-LLD-ANA-FNC-1273  H398-LLD-ANA-FNC-1275  H398-LLD-ANA-FNC-1276  H398-LLD-ANA-FNC-1288  H398-LLD-ANA-FNC-1289  H398-LLD-ANA-FNC-1317  H398-LLD-ANA-FNC-1325  H398-LLD-ANA-FNC-1338  H398-LLD-ANA-FNC-1403  H398-LLD-ANA-FNC-1404  H398-LLD-ANA-FNC-1479  H398-LLD-ANA-FNC-1480  H398-LLD-ANA-FNC-1481  H398-LLD-ANA-FNC-1482  H398-LLD-ANA-FNC-1491  H398-LLD-ANA-FNC-1492  H398-LLD-ANA-FNC-1767  H398-LLD-ANA-FNC-1831  H398-LLD-ANA-FNC-1832  H398-LLD-ANA-FNC-1833  Below Sections are modified :  1, 2, 4, 6, 9, 10.1.4.6, 10.1.7.6, 10.1.11.6, 10.1.13.6, 10.2.1.6, 10.3.2.6, 10.3.3.6, 10.3.4.6, 10.3.8.6, 10.3.9.4, 10.3.9.6 , 10.3.10.4, 10.3.10.6, 10.9.1.6, 10.11.1.1, 10.11.1.7, 10.11.2.6, 13,14  The following requirements have been Deleted:  Functional requirement:  H398-LLD-ANA-FNC-163  H398-LLD-ANA-FNC-164  H398-LLD-ANA-FNC-165  H398-LLD-ANA-FNC-166  The following requirements have been Added:  Functional requirements:  H398-LLD-ANA-FNC-1897  H398-LLD-ANA-FNC-1898  H398-LLD-ANA-FNC-1899  H398-LLD-ANA-FNC-2610  H398-LLD-ANA-FNC-2326  H398-LLD-ANA-FNC-2327  H398-LLD-ANA-FNC-2328  H398-LLD-ANA-FNC-2337  H398-LLD-ANA-FNC-2338  H398-LLD-ANA-FNC-2339  H398-LLD-ANA-FNC-2340  H398-LLD-ANA-FNC-2341  H398-LLD-ANA-FNC-2342  H398-LLD-ANA-FNC-2343  H398-LLD-ANA-FNC-2344  H398-LLD-ANA-FNC-2345  H398-LLD-ANA-FNC-2346  H398-LLD-ANA-FNC-2355  H398-LLD-ANA-FNC-2356  H398-LLD-ANA-FNC-2357  H398-LLD-ANA-FNC-2358  H398-LLD-ANA-FNC-2359  H398-LLD-ANA-FNC-2360  H398-LLD-ANA-FNC-1917  H398-LLD-ANA-FNC-1918  H398-LLD-ANA-FNC-1919  H398-LLD-ANA-FNC-1920  H398-LLD-ANA-FNC-1921  H398-LLD-ANA-FNC-1922  H398-LLD-ANA-FNC-1923  H398-LLD-ANA-FNC-1924  H398-LLD-ANA-FNC-1925  H398-LLD-ANA-FNC-1926  H398-LLD-ANA-FNC-1927  H398-LLD-ANA-FNC-1928  H398-LLD-ANA-FNC-1929  H398-LLD-ANA-FNC-1930  H398-LLD-ANA-FNC-1931  H398-LLD-ANA-FNC-1932  H398-LLD-ANA-FNC-1941  H398-LLD-ANA-FNC-1942  H398-LLD-ANA-FNC-1943  H398-LLD-ANA-FNC-2011  H398-LLD-ANA-FNC-2012  H398-LLD-ANA-FNC-2013  H398-LLD-ANA-FNC-2014  H398-LLD-ANA-FNC-2015  H398-LLD-ANA-FNC-2016  H398-LLD-ANA-FNC-2017  H398-LLD-ANA-FNC-2026  H398-LLD-ANA-FNC-2027  H398-LLD-ANA-FNC-2028  H398-LLD-ANA-FNC-2029  H398-LLD-ANA-FNC-2039  H398-LLD-ANA-FNC-2040  H398-LLD-ANA-FNC-2041  H398-LLD-ANA-FNC-2042  H398-LLD-ANA-FNC-2043  H398-LLD-ANA-FNC-2044  H398-LLD-ANA-FNC-2045  H398-LLD-ANA-FNC-2046  H398-LLD-ANA-FNC-2047  H398-LLD-ANA-FNC-2048  H398-LLD-ANA-FNC-2049  H398-LLD-ANA-FNC-2050  H398-LLD-ANA-FNC-2059  H398-LLD-ANA-FNC-2060  H398-LLD-ANA-FNC-2061  H398-LLD-ANA-FNC-2062  H398-LLD-ANA-FNC-2063  H398-LLD-ANA-FNC-2064  H398-LLD-ANA-FNC-2065  H398-LLD-ANA-FNC-2066  H398-LLD-ANA-FNC-2067  H398-LLD-ANA-FNC-2068  H398-LLD-ANA-FNC-2069  H398-LLD-ANA-FNC-2070  H398-LLD-ANA-FNC-2079  H398-LLD-ANA-FNC-2080  H398-LLD-ANA-FNC-2081  H398-LLD-ANA-FNC-2082  H398-LLD-ANA-FNC-2083  H398-LLD-ANA-FNC-2084  H398-LLD-ANA-FNC-2085  H398-LLD-ANA-FNC-2086  H398-LLD-ANA-FNC-2087  H398-LLD-ANA-FNC-2088  H398-LLD-ANA-FNC-2089  H398-LLD-ANA-FNC-2090  H398-LLD-ANA-FNC-2099  H398-LLD-ANA-FNC-2100  H398-LLD-ANA-FNC-2101  H398-LLD-ANA-FNC-2102  H398-LLD-ANA-FNC-2103  H398-LLD-ANA-FNC-2104  H398-LLD-ANA-FNC-2105  H398-LLD-ANA-FNC-2106  H398-LLD-ANA-FNC-2107  H398-LLD-ANA-FNC-2108  H398-LLD-ANA-FNC-2109  H398-LLD-ANA-FNC-2110  H398-LLD-ANA-FNC-2119  H398-LLD-ANA-FNC-2120  H398-LLD-ANA-FNC-2121  H398-LLD-ANA-FNC-2122  H398-LLD-ANA-FNC-2123  H398-LLD-ANA-FNC-2124  H398-LLD-ANA-FNC-2125  H398-LLD-ANA-FNC-2126  H398-LLD-ANA-FNC-2127  H398-LLD-ANA-FNC-2128  H398-LLD-ANA-FNC-2129  H398-LLD-ANA-FNC-2130  H398-LLD-ANA-FNC-2139  H398-LLD-ANA-FNC-2140  H398-LLD-ANA-FNC-2141  H398-LLD-ANA-FNC-2142  H398-LLD-ANA-FNC-2143  H398-LLD-ANA-FNC-2144  H398-LLD-ANA-FNC-2145  H398-LLD-ANA-FNC-2146  H398-LLD-ANA-FNC-2147  H398-LLD-ANA-FNC-2148  H398-LLD-ANA-FNC-2149  H398-LLD-ANA-FNC-2150  H398-LLD-ANA-FNC-2159  H398-LLD-ANA-FNC-2160  H398-LLD-ANA-FNC-2161  H398-LLD-ANA-FNC-2162  H398-LLD-ANA-FNC-2163  H398-LLD-ANA-FNC-2164  H398-LLD-ANA-FNC-2165  H398-LLD-ANA-FNC-2166  H398-LLD-ANA-FNC-2167  H398-LLD-ANA-FNC-2168  H398-LLD-ANA-FNC-2169  H398-LLD-ANA-FNC-2170  H398-LLD-ANA-FNC-2179  H398-LLD-ANA-FNC-2180  H398-LLD-ANA-FNC-2181  H398-LLD-ANA-FNC-2182  H398-LLD-ANA-FNC-2183  H398-LLD-ANA-FNC-2184  H398-LLD-ANA-FNC-2185  H398-LLD-ANA-FNC-2186  H398-LLD-ANA-FNC-2187  H398-LLD-ANA-FNC-2188  H398-LLD-ANA-FNC-2189  H398-LLD-ANA-FNC-2190  H398-LLD-ANA-FNC-2199  H398-LLD-ANA-FNC-2200  H398-LLD-ANA-FNC-2201  H398-LLD-ANA-FNC-2202  H398-LLD-ANA-FNC-2203  H398-LLD-ANA-FNC-2204  H398-LLD-ANA-FNC-2205  H398-LLD-ANA-FNC-2206  H398-LLD-ANA-FNC-2207  H398-LLD-ANA-FNC-2208  H398-LLD-ANA-FNC-2209  H398-LLD-ANA-FNC-2210  H398-LLD-ANA-FNC-2219  H398-LLD-ANA-FNC-2220  H398-LLD-ANA-FNC-2221  H398-LLD-ANA-FNC-2222  H398-LLD-ANA-FNC-2223  H398-LLD-ANA-FNC-2224  H398-LLD-ANA-FNC-2225  H398-LLD-ANA-FNC-2226  H398-LLD-ANA-FNC-2227  H398-LLD-ANA-FNC-2228  H398-LLD-ANA-FNC-2229  H398-LLD-ANA-FNC-2230  H398-LLD-ANA-FNC-1953  H398-LLD-ANA-FNC-1962  H398-LLD-ANA-FNC-1971  H398-LLD-ANA-FNC-1980  H398-LLD-ANA-FNC-1989  H398-LLD-ANA-FNC-1990  H398-LLD-ANA-FNC-2000  H398-LLD-ANA-FNC-2001  H398-LLD-ANA-FNC-2002  H398-LLD-ANA-FNC-2240  H398-LLD-ANA-FNC-2241  H398-LLD-ANA-FNC-2242  H398-LLD-ANA-FNC-2251  H398-LLD-ANA-FNC-2252  H398-LLD-ANA-FNC-2596  H398-LLD-ANA-FNC-2597  H398-LLD-ANA-FNC-2598  H398-LLD-ANA-FNC-2599  H398-LLD-ANA-FNC-2600  H398-LLD-ANA-FNC-2601  H398-LLD-ANA-FNC-2602  H398-LLD-ANA-FNC-2603  H398-LLD-ANA-FNC-2604  H398-LLD-ANA-FNC-2605  H398-LLD-ANA-FNC-2606  H398-LLD-ANA-FNC-2261  H398-LLD-ANA-FNC-2262  H398-LLD-ANA-FNC-2607  H398-LLD-ANA-FNC-2608  H398-LLD-ANA-FNC-2609  H398-LLD-ANA-FNC-2272  H398-LLD-ANA-FNC-2281  H398-LLD-ANA-FNC-2290  H398-LLD-ANA-FNC-2291  H398-LLD-ANA-FNC-2292  H398-LLD-ANA-FNC-2293  H398-LLD-ANA-FNC-2302  H398-LLD-ANA-FNC-2311  H398-LLD-ANA-FNC-2312  H398-LLD-ANA-FNC-2313  H398-LLD-ANA-FNC-2314  H398-LLD-ANA-FNC-2315  H398-LLD-ANA-FNC-2316  H398-LLD-ANA-FNC-2325  H398-LLD-ANA-FNC-2369  H398-LLD-ANA-FNC-2370  H398-LLD-ANA-FNC-2371  H398-LLD-ANA-FNC-2372  H398-LLD-ANA-FNC-2373  H398-LLD-ANA-FNC-2374  H398-LLD-ANA-FNC-2375  H398-LLD-ANA-FNC-2376  H398-LLD-ANA-FNC-2377  H398-LLD-ANA-FNC-2378  H398-LLD-ANA-FNC-2379  H398-LLD-ANA-FNC-2380  H398-LLD-ANA-FNC-2381  H398-LLD-ANA-FNC-2382  H398-LLD-ANA-FNC-2383  H398-LLD-ANA-FNC-2384  H398-LLD-ANA-FNC-2385  H398-LLD-ANA-FNC-2386  H398-LLD-ANA-FNC-2442  H398-LLD-ANA-FNC-2443  H398-LLD-ANA-FNC-2444  H398-LLD-ANA-FNC-2445  H398-LLD-ANA-FNC-2454  H398-LLD-ANA-FNC-2455  H398-LLD-ANA-FNC-2464  H398-LLD-ANA-FNC-2465  H398-LLD-ANA-FNC-2474  H398-LLD-ANA-FNC-2475  H398-LLD-ANA-FNC-2484  H398-LLD-ANA-FNC-2485  H398-LLD-ANA-FNC-2494  H398-LLD-ANA-FNC-2495  H398-LLD-ANA-FNC-2504  H398-LLD-ANA-FNC-2513  H398-LLD-ANA-FNC-2522  H398-LLD-ANA-FNC-2531  H398-LLD-ANA-FNC-2540  H398-LLD-ANA-FNC-2549  H398-LLD-ANA-FNC-2558  H398-LLD-ANA-FNC-2567  H398-LLD-ANA-FNC-2585  H398-LLD-ANA-FNC-2586  H398-LLD-ANA-FNC-2595  H398-LLD-ANA-FNC-2396  H398-LLD-ANA-FNC-2405  H398-LLD-ANA-FNC-2414  H398-LLD-ANA-FNC-2423  H398-LLD-ANA-FNC-2432  H398-LLD-ANA-FNC-2433  Below sections are Newly added:  10.1.12 NvicConfig,10.3.12 ScaleTach,10.3.13 Scalepwmc  10.11.3 LookupTableLookup32,  10.11.4 LookupTableInverse32,  10.18.3 TMRInitTIM12,10.18.4 GetDivConfig,10.18.5 TMRInitTach1,10.18.6 TMRInitTach2,10.18.7 TMRInitTach3,10.18.8 TMRInitTach4,10.18.9 TMRInitTach5,10.18.10 TMRInitTach6,10.18.11 TMRInitPwmc1,10.18.12 TMRInitPwmc2,10.18.13 TMRInitPwmc3,10.18.14 TMRInitPwmc4 ,  10.23 dauanasspi  10.24 dauanapwmc  10.25 dauanatach  11.5.5 DmaDeInit  11.12.6 TimICIit,11.12.7 TimSetIC1Prescaler,11.12.8 TimSetIC2Prescaler,11.12.9 TimSetIC3Prescaler,11.12.10 TimSetIC4Prescaler, 11.12.11 TimDmaCmd,11.12.12 Ti1Config, 11.12.13 Ti2Config,11.12.14 Ti3Config,11.12.15 Ti4Config  11.12.16 TIMSelectInputTrigger  11.12.17 TIMSelectSlaveMode  11.12.18 TIMSelectMasterSlaveMode  11.12.19 TIMUpdateRequestConfig  11.12.20 TIMGetCapture1  11.12.21 TIMGetCapture2  11.12.22 TIMGetFlagStatus  11.13 daulibstm32f4xxspi | 100171 | PRIYANKA B M | 01-Feb-2024 |
| 2.1 | Updated as per self review  Designations got modified in Front page | PR100171 | MUPPINENI SRAVANTHI | 12-02-2024 |
| 2.2 | Updated as per “QA” review  Comments  Section 4:References | PR100171 | MUPPINENI SRAVANTHI | 14-02-2024 |
| 2.3 | The below requirements are modified as per self review  H398-LLD-ANA-FNC-305  H398-LLD-ANA-FNC-2001  H398-LLD-ANA-FNC-2313  H398-LLD-ANA-FNC-2314  H398-LLD-ANA-FNC-2315  H398-LLD-ANA-FNC-2357  H398-LLD-ANA-FNC-2358  H398-LLD-ANA-FNC-304  H398-LLD-ANA-FNC-360  H398-LLD-ANA-FNC-415  H398-LLD-ANA-FNC-937  H398-LLD-ANA-FNC-229  H398-LLD-ANA-FNC-2359  H398-LLD-ANA-FNC-227  H398-LLD-ANA-FNC-228  H398-LLD-ANA-FNC-230  H398-LLD-ANA-FNC-2328  H398-LLD-ANA-FNC-2311  H398-LLD-ANA-FNC-1832  The below requirements are added  H398-LLD-ANA-FNC-2434  H398-LLD-ANA-FNC-2435  H398-LLD-ANA-FNC-2436  H398-LLD-ANA-FNC-2437  H398-LLD-ANA-FNC-2438  H398-LLD-ANA-FNC-2439  H398-LLD-ANA-FNC-2440  H398-LLD-ANA-FNC-2441  H398-LLD-ANA-FNC-2446  H398-LLD-ANA-FNC-2447  The below sections are modified  10.3.3.3  10.3.9.3  10.26.5.3 | PR100217 | MUPPINENI SRAVANTHI | 02-07-2024 |
| 2.4 | The below requirements are updated as per review comments  H398-LLD-ANA-FNC-1831  H398-LLD-ANA-FNC-1832  H398-LLD-ANA-FNC-2448 | PR100217 | MUPPINENI SRAVANTHI | 04-07-2024 |
| 2.5 | The below sections are updated as per “QA” review comments  Updated front page  Table of contents  section 4 | PR100217 | MUPPINENI SRAVANTHI | 10-07-2024 |
| 2.6 | The below requirements are modified as per self review  H398-LLD-ANA-FNC-2326  H398-LLD-ANA-FNC-415  H398-LLD-ANA-FNC-227  H398-LLD-ANA-FNC-2448 | PR100245 | MUPPINENI SRAVANTHI | 08-08-2024 |
| 2.7 | The below requirements are updated ass per review comments H398-LLD-ANA-FNC-2326 | PR100245 | MUPPINENI SRAVANTHI | 13-08-2024 |
| 2.8 | The below requirements are updated ass per “QA” review comments H398-LLD-ANA-FNC-2326 | PR100245 | MUPPINENI SRAVANTHI | 19-08-2024 |
| 2.9 | The below requirements are updated as per sef review  H398-LLD-ANA-FNC-2001  H398-LLD-ANA-FNC-2439 | PR100263 | MUPPINENI SRAVANTHI | 23-08-2024 |
| 2.10 | The below requirements are updated as per self review  H398-LLD-ANA-FNC-227  H398-LLD-ANA-FNC-2441  H398-LLD-ANA-FNC-547  H398-LLD-ANA-FNC-1831  The below requirements are added H398-LLD-ANA-FNC-2449 | PR100275 | MUPPINENI SRAVANTHI | 28-08-2024 |
| 2.11 | The below requirements are updated as per self review  H398-LLD-ANA-FNC-1831  H398-LLD-ANA-FNC-2448  Section 13 Appendix A : Data Dictionary, Analog MCD table | PR100283 | MUPPINENI SRAVANTHI | 29-08-2024 |
| 2.12 | The below requirements are updated:  H398-LLD-ANA-FNC-229  H398-LLD-ANA-FNC-2356  H398-LLD-ANA-FNC-2326  H398-LLD-ANA-FNC-2328  Section 14 Appendix B : Data Constants, Analog application softwate. | PR100360 | Madhumitha M | 20-Jun-2025 |
| 2.13 | The below requirements are updated:  H398-LLD-ANA-FNC-1831 | PR100388 | Afreen P | 10-Oct-2025 |

# 2 Objective

The document H398-003-011-ANA defines the Software Low Level Requirements for the Computer Software Configuration Item (CSCI) Analog Module Application Software of Engine Data Acquisition Unit (EDAU) for Airbus Helicopter model AS532U2.

# 3 Scope

The document H398-003-011-ANA specifies the Software Low Level Requirements of the Analog Module. This document is written according to H398-001-007 and satisfies RTCADO-178B section 10.10 for Software Design Data.

# 4 References

Table : Reference document

|  |  |  |
| --- | --- | --- |
| RTCA | DO-178B | Software Considerations in Airborne Systems and Equipment Certification |
| Howell Instruments, Inc. | SYS2160SRS | Airbus Helicopter AS532U2 Engine Instrument System Requirements Specification |
| 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 | H398-001-007 | Software Design Standards |
| ALTEN Global Technologies Private Limited | H398-002-001-ANA | Software Requirements Specification for Analog Module of Airbus(GVMS) Engine Data Acquisition Unit (EDAU) and Configuration Management Unit plus NVM (CMU+) |
| ALTEN Global Technologies Private Limited | H398-003-001-ANA | Software Architectural Design for Discrete Module of Airbus(GVMS) Engine Data Acquisition Unit (EDAU) and Configuration Management Unit plus NVM (CMU+) |
| STMicroelectronics | DM00031020.pdf | Reference manual for STM32F405xx/07xx, STM32F415xx/17xx, STM32F42xxx and STM32F43xxx advanced ARM®-based 32-bit MCUs |
| STMicroelectronics | DM00046982-with FPU-ref-manual.pdf | STM32F3 and STM32F4 Series Cortex®-M4 programming manual |

# 5 Acronyms and Definitions

Table : List of Abbreviation

|  |  |
| --- | --- |
| **Acronym** | **Definition** |
| ADC | Analog to Digital Converter |
| AF | Alternate Function |
| ARINC | Aeronautical Radio Incorporated |
| CAN | Controller Area Network |
| CBIT | Continuous Built In Test |
| CMS | Control Management System |
| CMSIS | Cortex microcontroller software interface standard |
| CMU+ | Configuration Management Unit Plus NVM |
| CPU | Central Processing Unit |
| CRC | Cyclic Redundancy Check |
| CSCI | Computer Software Configuration Item |
| CSU | Computer Software Unit |
| DMA | Direct Memory Access |
| EEC | Electronic Engine Controller |
| FSMC | Flexible Static Memory Controller |
| GPIO | General Purpose  Input Output |
| H/W | Hardware |
| HLR | High Level Requirement |
| I/O | Input / Output |
| ISR | Interrupt Service Routine |
| LCC | Logical Communication Channel |
| LED | Light Emitting Diode |
| LLR | Low Level Requirements |
| Max | Maximum |
| Min | Minimum |
| Mux | Multiplexer |
| NA | Not Applicable |
| NOC | Normal Operating Channel |
| NSC | Node Service Channel |
| NVIC | Nested Vectored Interrupt Controller |
| O/P | Output |
| OS | Operating System |
| QA | Quality Assurance |
| RAM | Random Access Memory |
| RCI | Redundancy Channel Identifier |
| ROM | Read Only Memory |
| RTCA | Radio Technical Commission for Aeronautics |
| Rx | Receiver |
| SAD | Software Architectural Document |
| SLL | Software Low Level Requirements |
| SRAM | Static Random Access Memory |
| SRS | Software Requirement Specifications |
| SVN | Subversion |
| Tx | Transmission |
| TCB | Task Control Block |

# 6 Document Control

According to the H398-001-004 (SCMP), this document shall be maintained in ALTEN Software & Software Configuration management System (CMS) after the QA review, under the following path [http://192.168.1.230/svn/A21HOWBLDAU/V2\_H398/ACCORD/SW/Branches/SOI2/Documents/Design/LLD] under change control.

# 7 Responsibilities

1. Software Development Team members specify the Software Low Level Requirements.

2. Reviewers review the Software Low Level Requirements based on the DO-178B check points provided as attributes in ReMa.

3. Once all the comments from the Reviewer are 'Looked Into' and the Software Low Level Requirement status is ‘Closed’ in ReMa, QA changes the QA Review Status Requirement to "QA Approved" and closed.

4. Engineering Manager is responsible for base lining the document to SVN through ReMa, as well as the Exported document, into the path http://192.168.1.230/svn/A21HOWBLDAU/V2\_H398/ACCORD/SW/Branches/SOI2/Documents/Design/LLD

# 8 Distributions

The Software Low Level Requirements Document of Analog Module (H398-003-011-ANA) is distributed to Howell Instruments, Inc through a secured File Transform Protocol server.

# 9 Traceability

The Software Low Level Requirements for Analog Module of Engine Data Acquisition Unit of Airbus EDAU and CMU+(H398-003-011-ANA) is derived from the Software High level Requirements present in the Software High Level Requirements Specification Document of Analog Module (H398-002-001-ANA) and Software Architectural Design for Analog Module of Engine Data Acquisition Unit of Airbus GVMS EDAU and CMU+ (H398-003-001-ANA). Traceability of H398-003-011-ANA to H398-002-001-ANA and H398-003-001-ANA is provided in Bi-Directional Traceability Matrix from SLL to SRS (H398-003-012-ANA).

# 10 Software Low Level Requirements- Analog Application Software

This section specifies the Software Low Level Requirements for Analog Module.

Refer Appendix A: Data Structures of H398-003-001-ANA for Data Structures and Enumerations.

## 10.1 dauanaa825

This module contains Implementation of ARINC825 routines

### 10.1.1 RxQueueRemove

Low Level Design Details about CSU RxQueueRemove will follow in the sub sections.

#### 10.1.1.1 Brief Description

The function RxQueueRemove read the a825 message from the receiver message queue and as per the message type (broadcast/peer to peer) load to Destination address and clear the a825 message queue.

#### 10.1.1.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.1.1.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 10.1.1.4 Parameter list (Input/Output)

Inputs: 1) T\_A825\_RX\_QUEUE\* psRxQueue -> CAN message queue

Outputs: 1) T\_A825\_MSG\* psDestination -> Destination address to copy the a825 message.

2) T\_A825\_RX\_QUEUE\* psRxQueue -> CAN message queue

#### 10.1.1.5 Return Value

T\_A825\_RX\_STATUS - Return the status of a825 message

- Return A825\_RX\_QUEUE\_EMPTY if the msg queue is empty

- Return A825\_RX\_OK if A825 msg is successfully loaded to Destination address

#### 10.1.1.6 Other CSUs called by this CSU

CanItConfig

UtilsCopy

#### 10.1.1.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to RxQueueRemove

##### 10.1.1.7.1 dauanaa825-RxQueueRemove-LLR-001

Requirement ID: H398-LLD-ANA-FNC-20

The function shall return A825\_RX\_QUEUE\_EMPTY when A825 receiver message queue is empty (i.e u16\_Rx\_Cntr of psRxQueue is M\_ZERO), otherwise do nothing.

##### 10.1.1.7.2 dauanaa825-RxQueueRemove-LLR-002

Requirement ID: H398-LLD-ANA-FNC-21

The function shall disable the Can Receive or transmit Interrupt:

Call function ‘CanItConfig’ with parameter (M\_CAN1, M\_CAN\_IT\_FMP0, DISABLE)

##### 10.1.1.7.3 dauanaa825-RxQueueRemove-LLR-003

Requirement ID: H398-LLD-ANA-FNC-22

The function shall retrieve the CAN message from the receive queue ps\_Rx\_Head of psRxQueue.

##### 10.1.1.7.4 dauanaa825-RxQueueRemove-LLR-004

Requirement ID: H398-LLD-ANA-FNC-23

The function shall Decrement message counter (u16\_Rx\_Cntr of psRxQueue) as one message is read from the queue.

##### 10.1.1.7.5 dauanaa825-RxQueueRemove-LLR-005

Requirement ID: H398-LLD-ANA-FNC-24

The function shall set message header (ps\_Rx\_Head of psRxQueue) to Rx buffer (as\_Rx\_Buf of psRxQueue) starting address when next rx message address is not in range of message Queue Size (i.e pre-incremented value of ps\_Rx\_Head of psRxQueue is equal to as\_Rx\_Buf with size M\_A825\_RX\_Q\_SIZE of psRxQueue ), otherwise do nothing.

##### 10.1.1.7.6 dauanaa825-RxQueueRemove-LLR-006

Requirement ID: H398-LLD-ANA-FNC-25

The function shall enable the Can Receiver Interrupt:

Call function ‘CanItConfig’ with parameter (M\_CAN1, M\_CAN\_IT\_FMP0, ENABLE).

##### 10.1.1.7.7 dauanaa825-RxQueueRemove-LLR-007

Requirement ID: H398-LLD-ANA-FNC-26

The function shall set the Logical communication channel (bit 26-28) from the received CAN message to ((can message extended identifier bit shifted to right by M\_LCC\_LOC) bitwise AND with M\_LCC\_MSK))

(Ref. to the design document ARINC SPECIFICATION 825-2 for extracting the data from CAN message).

##### 10.1.1.7.8 dauanaa825-RxQueueRemove-LLR-008

Requirement ID: H398-LLD-ANA-FNC-27

The function shall translate CAN MESSAGE to ARINC 825 Broadcast MESSAGE when the message is a broadcast message i.e. LCC of ARINC 825 message is NOC.

- (can message extended identifier bit shifted to right by M\_RCI\_LOC) bitwise AND with M\_RCI\_MSK)

- RCI [bit 0...1]

- (can message extended identifier bit shifted to right by M\_DOC\_LOC) bitwise AND with M\_DOC\_MSK)

- DOC [bit 2 ... 15]

- (can message extended identifier bit shifted to right by M\_PVT\_LOC) bitwise AND with M\_PVT\_MSK)

- PVT [bit 16]

- (can message extended identifier bit shifted to right by M\_LCL\_LOC) bitwise AND with M\_LCL\_MSK)

- LCL [bit 17]

- (can message extended identifier bit shifted to right by M\_RSD\_LOC) bitwise AND with M\_RSD\_MSK)

- RSD [bit 18]

- (can message extended identifier bit shifted to right by M\_SRC\_FID\_LOC) bitwise AND with M\_SRC\_FID\_MSK)

- Source FID [bit 19 ... 25]

-Other bits to spare i.e. ZERO.

(Ref. to the design document ARINC SPECIFICATION 825-2 for extracting the data from CAN message)

##### 10.1.1.7.9 dauanaa825-RxQueueRemove-LLR-009

Requirement ID: H398-LLD-ANA-FNC-28

The function shall translate CAN MESSAGE to ARINC 825 Peer to Peer MESSAGE when the message is a peer to peer message i.e. LCC of ARINC 825 message is NSC.

- (can message extended identifier bit shifted to right by M\_RCI\_LOC) bitwise AND with M\_RCI\_MSK)

- RCI [bit 0...1]

- (can message extended identifier bit shifted to right by M\_SID\_LOC) bitwise AND with M\_SID\_MSK)

- SID [bit 2 ... 8]

- (can message extended identifier bit shifted to right by M\_SER\_FID\_LOC) bitwise AND with M\_SER\_FID\_MSK)

- Server FID [bit 9 ... 15]

- (can message extended identifier bit shifted to right by M\_PVT\_LOC) bitwise AND with M\_PVT\_MSK)

- PVT [bit 16]

- (can message extended identifier bit shifted to right by M\_LCL\_LOC) bitwise AND with M\_LCL\_MSK)

- LCL [bit 17]

- (can message extended identifier bit shifted to right by M\_SMT\_LOC) bitwise AND with M\_SMT\_MSK)

- SMT [bit 18]

- (can message extended identifier bit shifted to right by M\_CFID\_LOC) bitwise AND with M\_CFID\_MSK)

- Client FID [bit 19 ... 25]

-Other bits to spare i.e. ZERO.

(Ref. to the design document ARINC SPECIFICATION 825-2 for extracting the data from CAN message)

##### 10.1.1.7.10 dauanaa825-RxQueueRemove-LLR-010

Requirement ID: H398-LLD-ANA-FNC-29

The function shall do nothing when LCC of ARINC 825 message is other than NOC and NSC.

##### 10.1.1.7.11 dauanaa825-RxQueueRemove-LLR-011

Requirement ID: H398-LLD-ANA-FNC-30

The function shall translate CAN message DLC to A825 paysize

##### 10.1.1.7.12 dauanaa825-RxQueueRemove-LLR-012

Requirement ID: H398-LLD-ANA-FNC-31

The function shall set the payload size to M\_A825\_PAYLOAD\_SIZE when the payload size is greater than M\_A825\_PAYLOAD\_SIZE otherwise do nothing.

##### 10.1.1.7.13 dauanaa825-RxQueueRemove-LLR-013

Requirement ID: H398-LLD-ANA-FNC-32

The function shall Copy the payload data from the CAN message buffer to the arinc payload buffer by calling the function ‘UtilsCopy’.

##### 10.1.1.7.14 dauanaa825-RxQueueRemove-LLR-014

Requirement ID: H398-LLD-ANA-FNC-33

The function shall load the complete arinc message to the destination address (psDestination).

##### 10.1.1.7.15 dauanaa825-RxQueueRemove-LLR-015

Requirement ID: H398-LLD-ANA-FNC-34

The function shall return A825\_RX\_OK.

### 10.1.2 A825Receive

Low Level Design Details about CSU A825Receive will follow in the sub sections.

#### 10.1.2.1 Brief Description

The function A825Receive read the a825 message from the receiver message queue (CAN message) as per the Logical Communication Channels (LCCs) select and load to Destination address.

#### 10.1.2.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.1.2.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 10.1.2.4 Parameter list (Input/Output)

Inputs: T\_LCC\_TYPE eChan->select the Logical Communication Channels (

LCCs)

Outputs: T\_A825\_MSG \*psDestMess -> Destination address to copy the a825

message

#### 10.1.2.5 Return Value

T\_A825\_RX\_STATUS - Return the status of a825 message

- return A825\_RX\_QUEUE\_EMPTY if the msg queue is empty

- return A825\_RX\_OK if A825 msg is successfully loaded to Destination address

- return A825\_RX\_BAD\_CHANNEL if the LCC is invalid

#### 10.1.2.6 Other CSUs called by this CSU

RxQueueRemove

#### 10.1.2.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to A825Receive.

##### 10.1.2.7.1 dauanaa825-A825Receive-LLR-001

Requirement ID: H398-LLD-ANA-FNC-43

The function shall do the following when the selected LCC (eChan) is NOC:

- Call the function ‘RxQueueRemove’ with parameter (psDestMess, Reference to NOC message queue)

- Set the function return value to return value of function RxQueueRemove.

##### 10.1.2.7.2 dauanaa825-A825Receive-LLR-002

Requirement ID: H398-LLD-ANA-FNC-44

The function shall do the following when the selected LCC (eChan) is NSC:

* Call the function ‘RxQueueRemove’ with parameter (psDestMess, Reference to NSC message queue)
* Set the function return value to return value of function RxQueueRemove.

##### 10.1.2.7.3 dauanaa825-A825Receive-LLR-003

Requirement ID: H398-LLD-ANA-FNC-45

The function shall set the return value to A825\_RX\_BAD\_CHANNEL when the selected LCC (eChan) is other than NOC, NSC.

### 10.1.3 RxQueueInsert

Low Level Design Details about CSU RxQueueInsert will follow in the sub sections.

#### 10.1.3.1 Brief Description

The function RxQueueInsert inserts the received message to the a825 queue buffer.

#### 10.1.3.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.1.3.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 10.1.3.4 Parameter list (Input/Output)

Inputs: T\_CAN\_RX\_MSG \*psNewMessage -> The new message to insert in the a825

queue buffer.

T\_A825\_RX\_QUEUE \*psRxQueue -> Address of a825 queue buffer to copy the

a825 message.

Outputs: T\_A825\_RX\_QUEUE \*psRxQueue -> Address of a825 queue buffer to copy

the a825 message

#### 10.1.3.5 Return Value

T\_A825\_RX\_STATUS - Return the status of a825 message

- return A825\_RX\_OVER\_FLOW if buffer is full

- return A825\_RX\_OK if the received message stored successfully

#### 10.1.3.6 Other CSUs called by this CSU

None

#### 10.1.3.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to RxQueueInsert.

##### 10.1.3.7.1 dauanaa825-RxQueueInsert-LLR-001

Requirement ID: H398-LLD-ANA-FNC-54

The function shall return A825\_RX\_OVER\_FLOW when the queue is full (i.e M\_A825\_RX\_Q\_SIZE is equal to buffer queue counter u16\_Rx\_Cntr of psRxQueue), otherwise do nothing.

##### 10.1.3.7.2 dauanaa825-RxQueueInsert-LLR-002

Requirement ID: H398-LLD-ANA-FNC-55

The function shall insert the new message (psNewMessage) to the next empty space of the queue (ps\_Rx\_Tail of psRxQueue) and Increment message counter (u16\_Rx\_Cntr of psRxQueue) by M\_ONE.

##### 10.1.3.7.3 dauanaa825-RxQueueInsert-LLR-003

Requirement ID: H398-LLD-ANA-FNC-56

The function shall set next queue address (ps\_Rx\_Tail of psRxQueue) to the starting of the buffer address as\_Rx\_Buf of psRxQueue when Rx queue size is in range of Rx buffer (i.e pre incremented value of ps\_Rx\_Tail of psRxQueue is greater than or equal to as\_Rx\_Buf with size M\_A825\_RX\_Q\_SIZE of psRxQueue ), otherwise do nothing.

##### 10.1.3.7.4 dauanaa825-RxQueueInsert-LLR-004

Requirement ID: H398-LLD-ANA-FNC-57

The function shall return A825\_RX\_OK.

### 10.1.4 ReceiveIntr

Low Level Design Details about CSU ReceiveIntr will follow in the sub sections.

#### 10.1.4.1 Brief Description

The function ReceiveIntr receives the CAN message and as per the LCC determined, the message loaded to the a825 queue buffer.

#### 10.1.4.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.1.4.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 10.1.4.4 Parameter list (Input/Output)

Inputs : None

Outputs : None

#### 10.1.4.5 Return Value

None

#### 10.1.4.6 Other CSUs called by this CSU

OsIntEnter

CanReceive

RxQueueInsert

OsIntExit

OsSemPost

#### 10.1.4.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to ReceiveIntr.

##### 10.1.4.7.1 dauanaa825-ReceiveIntr-LLR-001

Requirement ID: H398-LLD-ANA-FNC-66

The function shall enter to the ISR by calling function ‘OsIntEnter’.

##### 10.1.4.7.2 dauanaa825-ReceiveIntr-LLR-002

Requirement ID: H398-LLD-ANA-FNC-67

The function shall retrieve the can message from the CAN message queue by calling function 'CanReceive' with parameter (M\_CAN1, M\_CAN\_FIFO0, Address to store the received message).

##### 10.1.4.7.3 dauanaa825-ReceiveIntr-LLR-003

Requirement ID: H398-LLD-ANA-FNC-68

The function shall retrieve the LCC from the CAN message as ((can message extended identifier bit shifted to right by M\_LCC\_LOC) bitwise AND with M\_LCC\_MSK)

- Normal Operating Channel (NOC)

- Node Service Channel (NSC)

- Any other case invalid channel

##### 10.1.4.7.4 dauanaa825-ReceiveIntr-LLR-004

Requirement ID: H398-LLD-ANA-FNC-69

The function shall do the following when the selected LCC is NOC:

* Call the function ‘RxQueueInsert’ with parameter (Reference to the received message, Reference to NOC queue)
* Set the status flag to the return value of function ‘RxQueueInsert’.
* Call OsSemPost with parameters A825 task semaphore when status flag is equal to A825\_RX\_OK orelse, do nothing.

##### 10.1.4.7.5 dauanaa825-ReceiveIntr-LLR-005

Requirement ID: H398-LLD-ANA-FNC-70

The function shall do the following when the selected LCC is NSC:

* Call the function ‘RxQueueInsert’ with parameter (Reference to the received message, Reference to NSC queue)
* Set the status flag to the return value of function ‘RxQueueInsert’.
* Call OsSemPost with parameters A825 task semaphore when status flag is equal to A825\_RX\_OK orelse, do nothing.

##### 10.1.4.7.6 dauanaa825-ReceiveIntr-LLR-006

Requirement ID: H398-LLD-ANA-FNC-71

The function shall set the status flag to A825\_RX\_BAD\_CHANNEL when the LCC is other than NOC & NSC.

##### 10.1.4.7.7 dauanaa825-ReceiveIntr-LLR-007

Requirement ID: H398-LLD-ANA-FNC-72

The function shall increment the error list by one when the status flag is not equal to A825\_RX\_OK and do nothing when the status flag is A825\_RX\_OK.

##### 10.1.4.7.8 dauanaa825-ReceiveIntr-LLR-008

Requirement ID: H398-LLD-ANA-FNC-73

The function shall Exit from the ISR by calling function ‘OsIntExit’.

### 10.1.5 TxFromQueue

Low Level Design Details about CSU TxFromQueue will follow in the sub sections.

#### 10.1.5.1 Brief Description

The function TxFromQueue transmit the CAN frame message.

#### 10.1.5.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.1.5.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 10.1.5.4 Parameter list (Input/Output)

Inputs : T\_A825\_TX\_QUEUE \*psTxQueue -> pointer to a825 message transmission queue.

Outputs: T\_A825\_TX\_QUEUE \*psTxQueue -> pointer to a825 message transmission queue.

#### 10.1.5.5 Return Value

T\_UINT8 - Return the Tx mailbox id

- M\_CAN\_TX\_MAILBOX\_0 if Mailbox 0 selected for Tx

- M\_CAN\_TX\_MAILBOX\_1 if Mailbox 1 selected for Tx

- M\_CAN\_TX\_MAILBOX\_2 if Mailbox 2 selected for Tx

- M\_CAN\_TXSTATUS\_NOMAILBOX if No Mailbox is selected for Tx

#### 10.1.5.6 Other CSUs called by this CSU

CanTransmit

#### 10.1.5.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to TxFromQueue.

##### 10.1.5.7.1 dauanaa825-TxFromQueue-LLR-001

Requirement ID: H398-LLD-ANA-FNC-82

The function shall transmit a A825 message by Calling function ‘CanTransmit’ with parameter (M\_CAN1, message header (ps\_Tx\_Head of psTxQueue)) and stores it in Tx mailbox id.

##### 10.1.5.7.2 dauanaa825-TxFromQueue-LLR-002

Requirement ID: H398-LLD-ANA-FNC-83

The function shall do nothing when the return value of the function CanTransmit is M\_CAN\_TXSTATUS\_NOMAILBOX (i.e Tx mailbox id).

##### 10.1.5.7.3 dauanaa825-TxFromQueue-LLR-003

Requirement ID: H398-LLD-ANA-FNC-84

The function shall decrement message counter (u16\_Tx\_Cntr of psTxQueue) by M\_ONE, when the function ‘CanTransmit’ returns other than M\_CAN\_TXSTATUS\_NOMAILBOX.

##### 10.1.5.7.4 dauanaa825-TxFromQueue-LLR-004

Requirement ID: H398-LLD-ANA-FNC-85

The function shall do the following when the function ‘CanTransmit’ returns other than M\_CAN\_TXSTATUS\_NOMAILBOX:

Set message header (ps\_Tx\_Head of psTxQueue) to Tx buffer (as\_Tx\_Buf of psTxQueue) starting address when next Tx message address is not in range of Tx message Queue Size (i.e as\_Tx\_Buf with size M\_A825\_TX\_Q\_SIZE of psTxQueue is less than or equal to ps\_Tx\_Head of psTxQueue), otherwise do nothing.

##### 10.1.5.7.5 dauanaa825-TxFromQueue-LLR-005

Requirement ID: H398-LLD-ANA-FNC-86

The function shall return the Tx mailbox id.

### 10.1.6 TransmitIntr

Low Level Design Details about CSU TransmitIntr will follow in the sub sections.

#### 10.1.6.1 Brief Description

The function TransmitIntr transmits all the CAN message from a825 message transmission queue till the message queue is empty or all the transmission channels are busy.

#### 10.1.6.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.1.6.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 10.1.6.4 Parameter list (Input/Output)

Inputs : None

Outputs : None

#### 10.1.6.5 Return Value

None

#### 10.1.6.6 Other CSUs called by this CSU

TxFromQueue

CanItConfig

#### 10.1.6.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to TransmitIntr.

##### 10.1.6.7.1 dauanaa825-TransmitIntr-LLR-001

Requirement ID: H398-LLD-ANA-FNC-95

The function shall Loop through all the LCC buffer ids and do the following:

a) Transmits the message from the queue, till the Tx counter for the buffer is not zero or no mailbox is free for transmission.

b) Calls function ‘TxFromQueue’ with parameter (reference to A825 transmit message queue with loop counter as index) to Transmit the message from the queue till the Tx counter for the buffer is not zero

c)Return from the function when function ‘TxFromQueue’ returns M\_CAN\_TXSTATUS\_NOMAILBOX, otherwise do nothing till the Tx counter for the buffer is not zero.

##### 10.1.6.7.2 dauanaa825-TransmitIntr-LLR-002

Requirement ID: H398-LLD-ANA-FNC-96

The function shall Disable the Can Transmit Interrupt by Calling function ‘CanItConfig’ with parameter (M\_CAN1, M\_CAN\_IT\_TME, DISABLE).

### 10.1.7 TxQueueInsert

Low Level Design Details about CSU TxQueueInsert will follow in the sub sections.

#### 10.1.7.1 Brief Description

The function TxQueueInsert insert the CAN message to a825 message transmission queue till the message queue is full.

#### 10.1.7.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.1.7.3 List of global variables accessed and modified

Accessed : U32\_critical\_sr

Modified : U32\_critical\_sr

#### 10.1.7.4 Parameter list (Input/Output)

Inputs: T\_CAN\_TX\_MSG \*psNewMessage -> Pointer to the CAN message

T\_A825\_TX\_QUEUE \*psTxQueue -> Pointer to a825 message

transmission queue

Outputs: T\_A825\_TX\_QUEUE \*psTxQueue -> Pointer to a825 message

transmission queue

#### 10.1.7.5 Return Value

T\_A825\_TX\_STATUS -> Return the Tx status of a825 message.

- return A825\_TX\_OVER\_FLOW if buffer is full

- return A825\_TX\_OK if the Transmitted message stored successfully

#### 10.1.7.6 Other CSUs called by this CSU

CanItConfig

TransmitIntr

#### 10.1.7.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to TxQueueInsert.

##### 10.1.7.7.1 dauanaa825-TxQueueInsert-LLR-001

Requirement ID: H398-LLD-ANA-FNC-105

The function shall do the following:

1. Call CanItConfig with parameters M\_CAN1, M\_CAN\_IT\_TME, DISABLE
2. Do the following when the queue is full (u16\_Tx\_Cntr of psTxQueue) (the buffer queue counter is greater than or equal to M\_A825\_TX\_Q\_SIZE) otherwise do nothing.

* Call CanItConfig with parameters M\_CAN1, M\_CAN\_IT\_TME, ENABLE
* Return A825\_TX\_OVER\_FLOW

##### 10.1.7.7.2 dauanaa825-TxQueueInsert-LLR-002

Requirement ID: H398-LLD-ANA-FNC-106

The function shall do the following:

a) Insert the new message (psNewMessage) to the next empty space of the queue (ps\_Tx\_Tail of psTxQueue) and Increments the message counter (u16\_Tx\_Cntr of psTxQueue) by M\_ONE.

##### 10.1.7.7.3 dauanaa825-TxQueueInsert-LLR-003

Requirement ID: H398-LLD-ANA-FNC-107

The function shall set next queue address (ps\_Tx\_Tail of psTxQueue) to the starting of the buffer (as\_Tx\_Buf of psTxQueue) address when Tx queue size is in range of Tx buffer (i.e pre incremented value of ps\_Tx\_Tail of psTxQueue is greater than or equal to as\_Tx\_Buf with size M\_A825\_TX\_Q\_SIZE of psTxQueue), otherwise do nothing.

##### 10.1.7.7.4 dauanaa825-TxQueueInsert-LLR-004

Requirement ID: H398-LLD-ANA-FNC-108

The function shall do the following:

1. Call CanItConfig with parameters M\_CAN1, M\_CAN\_IT\_TME and ENABLE.
2. Call the function TransmitIntr.
3. Return A825\_TX\_OK.

### 10.1.8 BuildBroadcastExtId

Low Level Design Details about CSU BuildBroadcastExtId will follow in the sub sections.

#### 10.1.8.1 Brief Description

The function BuildBroadcastExtId build the 29 bit CAN identifier for an ARINC 825 broadcast type message.

#### 10.1.8.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.1.8.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 10.1.8.4 Parameter list (Input/Output)

Inputs: T\_A825\_MSG \*p825brdcmsg -> Pointer to an ARINC 825 message

Outputs: None

#### 10.1.8.5 Return Value

T\_UINT32 - 29 bit identifier

#### 10.1.8.6 Other CSUs called by this CSU

None

#### 10.1.8.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to BuildBroadcastExtId.

##### 10.1.8.7.1 dauanaa825-BuildBroadcastExtId-LLR-001

Requirement ID: H398-LLD-ANA-FNC-119

The function shall build the 29 bit CAN extended identifier for an ARINC 825 broadcast type message as below:

((lcl of bid of sid of p825brdcmsg bit shifted to left by M\_LCL\_LOC) bitwise OR

(pvt of bid of sid of p825brdcmsg bit shifted to left by M\_PVT\_LOC) bitwise OR

(doc of bid of sid of p825brdcmsg bit shifted to left by M\_DOC\_LOC) bitwise OR

(rci of bid of sid of p825brdcmsg bit shifted to left by M\_RCI\_LOC) bitwise OR

(rsd of bid of sid of p825brdcmsg bit shifted to left by M\_RSD\_LOC) bitwise OR

(sfid of bid of sid of p825brdcmsg bit shifted to left by M\_SRC\_FID\_LOC) bitwise OR

(lcc of p825brdcmsg bit shifted to left by M\_LCC\_LOC))

- RCI [bit 0..1],

- DOC [bit 2 .. 15],

- PVT [bit 16] ,

- LCL [bit 17],

- RSD [bit 18],

- Source FID [bit 19 .. 25]

- LCC [bit 26 .. 28].

(Ref. to the design document ARINC SPECIFICATION 825-2 for extracting the data from CAN message)

##### 10.1.8.7.2 dauanaa825-BuildBroadcastExtId-LLR-002

Requirement ID: H398-LLD-ANA-FNC-120

The function shall return the extended identifier for the Broadcast msg.

### 10.1.9 BuildPeerToPeerExtId

Low Level Design Details about CSU BuildPeerToPeerExtId will follow in the sub sections.

#### 10.1.9.1 Brief Description

The function BuildPeerToPeerExtId build the 29 bit CAN identifier for an ARINC 825 peer to peer type message.

#### 10.1.9.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.1.9.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 10.1.9.4 Parameter list (Input/Output)

Inputs : T\_A825\_MSG \*p825pe2pemsg -> Pointer to an ARINC 825 message

Outputs : None

#### 10.1.9.5 Return Value

T\_UINT32 29-bit identifier

#### 10.1.9.6 Other CSUs called by this CSU

None

#### 10.1.9.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to BuildPeerToPeerExtId.

##### 10.1.9.7.1 dauanaa825-BuildPeerToPeerExtId-LLR-001

Requirement ID: H398-LLD-ANA-FNC-129

The function shall build the 29 bit CAN extended identifier for an ARINC 825 peer to peer type message as mentioned below:

((cfid of pid of sid of p825pe2pemsg bit shifted to left by M\_CFID\_LOC) bitwise OR

(smt of pid of sid of p825pe2pemsg bit shifted to left by M\_SMT\_LOC) bitwise OR

(lcl of pid of sid of p825pe2pemsg bit shifted to left by M\_LCL\_LOC) bitwise OR

(pvt of pid of sid of p825pe2pemsg bit shifted to left by M\_PVT\_LOC) bitwise OR

(sfid of pid of sid of p825pe2pemsg bit shifted to left by M\_SER\_FID\_LOC) bitwise OR

(sid of pid of sid of p825pe2pemsg bit shifted to left by M\_SID\_LOC) bitwise OR

(rci of pid of sid of p825pe2pemsg bit shifted to left by M\_RCI\_LOC) bitwise OR

(lcc of p825pe2pemsg bit shifted to left by M\_LCC\_LOC))

- RCI [bit 0..1],

- SID[bit 2 .. 8],

- Server FID[bit 9 .. 15]

- PVT[bit 16] ,

- LCL[bit 17],

- SMT[bit 18],

- Client FID[bit 19 .. 25]

- LCC[bit 26 .. 28].

(Refer the Arinc 825 design document for extracting the data from the received Arinc 825 message)

##### 10.1.9.7.2 dauanaa825-BuildPeerToPeerExtId-LLR-002

Requirement ID: H398-LLD-ANA-FNC-130

The function shall return the extended identifier for the peer to peer msg.

### 10.1.10 A825Transmit

Low Level Design Details about CSU A825Transmit will follow in the sub sections.

#### 10.1.10.1 Brief Description

The function A825Transmit transmit the CAN message.

#### 10.1.10.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.1.10.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 10.1.10.4 Parameter list (Input/Output)

Inputs: T\_A825\_MSG\* psTxMessage -> Pointer to the a825 message

Outputs: None

#### 10.1.10.5 Return Value

T\_A825\_TX\_STATUS -> Return the Tx status of a825 message.

- return A825\_TX\_BAD\_CHANNEL if the selected lcc is UNKNOWN Channel Type

- return A825\_TX\_BAD\_PAYLOAD\_SIZE if the payload size is out of range

- return the queue insertion status

#### 10.1.10.6 Other CSUs called by this CSU

BuildBroadcastExtId

BuildPeerToPeerExtId

UtilsCopy

TxQueueInsert

#### 10.1.10.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to A825Transmit.

##### 10.1.10.7.1 dauanaa825-A825Transmit-LLR-001

Requirement ID: H398-LLD-ANA-FNC-139

The function shall store the Normal Operation channel queue to build the broad cast message id in message queue and Call the function 'BuildBroadcastExtId' with parameter (psTxMessage) to build the message when selected lcc (lcc of psTxMessage) is NOC, return value is stored in ext\_id of CAN message to be transmitted.

##### 10.1.10.7.2 dauanaa825-A825Transmit-LLR-002

Requirement ID: H398-LLD-ANA-FNC-140

The function shall store the Node Service Channel queue to build the peer to peer message id in message queue and Call the function 'BuildPeerToPeerExtId' with parameter (psTxMessage) to build the message when selected lcc (lcc of psTxMessage) is NSC, return value is stored in ext\_id of CAN message to be transmitted.

##### 10.1.10.7.3 dauanaa825-A825Transmit-LLR-003

Requirement ID: H398-LLD-ANA-FNC-141

The function shall return A825\_TX\_BAD\_CHANNEL when the selected lcc (lcc of psTxMessage) is other than Normal Operation channel or Node Service channel.

##### 10.1.10.7.4 dauanaa825-A825Transmit-LLR-004

Requirement ID: H398-LLD-ANA-FNC-142

The function shall return A825\_TX\_BAD\_PAYLOAD\_SIZE when the payload size (u8\_paysize of psTxMessage) is greater than M\_A825\_PAYLOAD\_SIZE otherwise do nothing.

##### 10.1.10.7.5 dauanaa825-A825Transmit-LLR-005

Requirement ID: H398-LLD-ANA-FNC-143

The function shall set the remaining CAN message as follows:

- Set the identifier for the message to M\_CAN\_ID\_EXT

- Set the frame for the message to M\_CAN\_RTR\_DATA

- Set the length of the frame to the received payload size (u8\_paysize of psTxMessage).

##### 10.1.10.7.6 dauanaa825-A825Transmit-LLR-006

Requirement ID: H398-LLD-ANA-FNC-144

The function shall copy the payload data by calling the function ‘UtilsCopy’ with parameter as data of CAN message, (u8\_payload of psTxMessage) and payload size (u8\_paysize of psTxMessage).

##### 10.1.10.7.7 dauanaa825-A825Transmit-LLR-007

Requirement ID: H398-LLD-ANA-FNC-145

The function shall insert the message to the Tx queue by Calling the function ‘TxQueueInsert’ with parameter (Reference to the CAN message, Reference to a825 message transmission queue) and

return the return value of the function TxQueueInsert.

### 10.1.11 InitCAN1

Low Level Design Details about CSU InitCAN1 will follow in the sub sections.

#### 10.1.11.1 Brief Description

The function InitCAN1 Initialize CAN 1 I/O peripherals for Rx and Tx of message.

#### 10.1.11.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.1.11.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 10.1.11.4 Parameter list (Input/Output)

Inputs : None

Outputs : None

#### 10.1.11.5 Return Value

None

#### 10.1.11.6 Other CSUs called by this CSU

RccAhb1PeriphClockCmd

RccApb1PeriphClockCmd

GpioPinAFConfig

GpioInit

CanDeInit

CanInit

CanFilterInit

CanItConfig

#### 10.1.11.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to InitCAN1.

##### 10.1.11.7.1 dauanaa825-InitCAN1-LLR-001

Requirement ID: H398-LLD-ANA-FNC-154

The function shall enable GPIOA clock by calling function'RccAhb1PeriphClockCmd' with parameters (M\_RCC\_AHB1PERIPH\_GPIOA, ENABLE).

##### 10.1.11.7.2 dauanaa825-InitCAN1-LLR-002

Requirement ID: H398-LLD-ANA-FNC-155

The function shall Enable CAN clock by Calling function 'RccApb1PeriphClockCmd' with parameter (M\_RCC\_APB1PERIPH\_CAN1, ENABLE).

##### 10.1.11.7.3 dauanaa825-InitCAN1-LLR-003

Requirement ID: H398-LLD-ANA-FNC-156

The function shall Configure pin function of PA11 by Calling function ‘GpioPinAFConfig’ with parameter (M\_GPIOA, M\_GPIO\_PINSOURCE11, M\_GPIO\_AF\_CAN1).

##### 10.1.11.7.4 dauanaa825-InitCAN1-LLR-004

Requirement ID: H398-LLD-ANA-FNC-157

The function shall Configure pin function of PA12 by Calling function 'GpioPinAFConfig' with parameter (M\_GPIOA, M\_GPIO\_PINSOURCE12, M\_GPIO\_AF\_CAN1).

##### 10.1.11.7.5 dauanaa825-InitCAN1-LLR-005

Requirement ID: H398-LLD-ANA-FNC-158

The function shall Configure CAN RX and TX pins as given below

- Set gpio\_pin to M\_GPIOA\_CAN\_RX1 bitwise or M\_GPIOA\_CAN\_TX1

- Set gpio\_mode to GPIO\_MODE\_AF

- Set gpio\_speed to GPIO\_SPEED\_50MHZ

- Set gpio\_otype to GPIO\_OTYPE\_PP

- Set gpio\_pupd to GPIO\_PUPD\_UP

##### 10.1.11.7.6 dauanaa825-InitCAN1-LLR-006

Requirement ID: H398-LLD-ANA-FNC-159

The function shall Initializes the GPIOA peripheral by Calling function 'GpioInit' with parameter (M\_GPIOA, Reference to GPIO init structure).

##### 10.1.11.7.7 dauanaa825-InitCAN1-LLR-007

Requirement ID: H398-LLD-ANA-FNC-160

The function shall initialize the CAN peripheral registers to their default reset values by Calling function CanDeInit with parameter (M\_CAN1).

##### 10.1.11.7.8 dauanaa825-InitCAN1-LLR-008

Requirement ID: H398-LLD-ANA-FNC-161

The function shall set the CAN init structure to the value given below

- Set can\_ttcm to DISABLE

- Set can\_abom to ENABLE

- Set can\_awum to DISABLE

- Set can\_nart to DISABLE

- Set can\_rflm to DISABLE

- Set can\_txfp to DISABLE

- Set can\_sjw to M\_CAN\_SJW\_1TQ

- Set can\_bs1 to M\_CAN\_BS1\_15TQ

- Set can\_bs2 to M\_CAN\_BS2\_5TQ

- Set can\_prescaler to M\_TIME\_QUANTAM\_RESET\_VAL

- Set can\_mode to M\_CAN\_MODE\_NORMAL

And call function CanInit with parameters (M\_CAN1, Reference to CAN init structure)

##### 10.1.11.7.9 dauanaa825-InitCAN1-LLR-009

Requirement ID: H398-LLD-ANA-FNC-162

The function shall set CAN filter init structure as below:

- Set can\_filter\_number to M\_ZERO

- Set can\_filter\_mode to M\_CAN\_FILTERMODE\_IDMASK

- Set can\_filter\_scale to M\_CAN\_FILTERSCALE\_32BIT

- Set can\_filter\_id\_high to M\_HEXA\_ZERO

- Set can\_filter\_id\_low to M\_HEXA\_ZERO

- Set can\_filter\_mask\_id\_high to M\_HEXA\_ZERO

- Set can\_filter\_mask\_id\_low to M\_HEXA\_ZERO

- Set can\_filter\_fifo\_assignment to M\_ZERO

- Set can\_filter\_activation to ENABLE.

- Call function ‘CanFilterInit’ with parameter (Reference to CAN filter init structure)

##### 10.1.11.7.10 dauanaa825-InitCAN1-LLR-014

Requirement ID: H398-LLD-ANA-FNC-1908

The function shall enable receive FIFO 0 message pending Interrupt by calling function ‘CanItConfig’ with parameter (M\_CAN1, M\_CAN\_IT\_FMP0, ENABLE).

### 10.1.12 NvicConfig

Low Level Design Details about CSU NvicConfig will follow in the sub sections.

#### 10.1.12.1 Brief Description

This function configures NVIC peripheral for Rx and Tx of message.

#### 10.1.12.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.1.12.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 10.1.12.4 Parameter list (Input/Output)

Inputs : None

Outputs : None

#### 10.1.12.5 Return Value

None

#### 10.1.12.6 Other CSUs called by this CSU

IntrInstall

NvicInit

#### 10.1.12.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to NvicConfig.

##### 10.1.12.7.1 dauanaa825-NvicConfig-LLR-001

Requirement ID: H398-LLD-ANA-FNC-1897

The function shall Call function ‘IntrInstall’ with parameters INTR\_CAN\_1\_RX\_0 and ReceiveIntr to set up the CAN receive interrupt vector.

##### 10.1.12.7.2 dauanaa825-NvicConfig-LLR-002

Requirement ID: H398-LLD-ANA-FNC-1898

The function shall Call function ‘IntrInstall’ with parameters INTR\_CAN\_1\_TX and TransmitIntr to set up the CAN transmit interrupt vector.

##### 10.1.12.7.3 dauanaa825-NvicConfig-LLR-003

Requirement ID: H398-LLD-ANA-FNC-1899

The function shall Initialize the NVIC init structure (CAN 1 receive interrupt) to the value given below

- Set nvic\_irq\_channel to CAN1\_RX0\_IRQN

- Set nvic\_irq\_channel\_preemption\_priority to NVIC\_PRIORITY\_LEVEL\_0

- Set nvic\_irq\_channel\_subpriority to NVIC\_PRIORITY\_LEVEL\_0

- Set nvic\_irq\_channel\_cmd to ENABLE

-Call function ‘NvicInit’ with parameter (Reference to NVIC init structure)

##### 10.1.12.7.4 dauanaa825-NvicConfig-LLR-004

Requirement ID: H398-LLD-ANA-FNC-2610

The function shall Initialize the NVIC init structure (CAN 1 receive interrupt) to the value given below

- Set nvic\_irq\_channel to CAN1\_TX\_IRQN

- Set nvic\_irq\_channel\_preemption\_priority to NVIC\_PRIORITY\_LEVEL\_0

- Set nvic\_irq\_channel\_subpriority to NVIC\_PRIORITY\_LEVEL\_0

- Set nvic\_irq\_channel\_cmd to ENABLE

-Call function ‘NvicInit’ with parameter (Reference to NVIC init structure)

### 10.1.13 A825Init

Low Level Design Details about CSU A825Init will follow in the sub sections.

#### 10.1.13.1 Brief Description

The function A825Init creates a semaphore to signal the A825 receiver task.

#### 10.1.13.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.1.13.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 10.1.13.4 Parameter list (Input/Output)

Inputs : None

Outputs : None

#### 10.1.13.5 Return Value

None

#### 10.1.13.6 Other CSUs called by this CSU

OsSemCreate

NvicConfig

InitCAN1

#### 10.1.13.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to A825Init.

##### 10.1.13.7.1 dauanaa825-A825Init-LLR-001

Requirement ID: H398-LLD-ANA-FNC-176

The function shall create a Semaphore to signal the A825 receiver task by calling the function ‘OsSemCreate’ with parameter M\_ZERO and update the 'Semaphore A825 task' with the return value of the function 'OsSemCreate '.

##### 10.1.13.7.2 dauanaa825-A825Init-LLR-002

Requirement ID: H398-LLD-ANA-FNC-177

The function shall do the following:

1. Call the function InitCAN1.
2. Call the function NvicConfig.

## 10.2 dauanaa825comm

The Module implements the ARINC 825 Task and its communication with Analog Module.

### 10.2.1 A825CommTask

Low Level Design Details about CSU A825CommTask will follow in the sub sections.

#### 10.2.1.1 Brief Description

This Task is signalled from the ARINC 825 receiver function to process the received message. The task will only service NOC and NSC messages.

#### 10.2.1.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.2.1.3 List of global variables accessed and modified

Accessed: None

Modified: None

#### 10.2.1.4 Parameter list (Input/Output)

Inputs: void \*pData: Pointer to NULL data.

Outputs: void \*pData: Pointer to NULL data.

#### 10.2.1.5 Return Value

None

#### 10.2.1.6 Other CSUs called by this CSU

OsSemPend

A825Receive

AppMessageNOC

AppMessageNSC

#### 10.2.1.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to A825CommTask.

##### 10.2.1.7.1 dauanaa825comm-A825CommTask-LLR-001

Requirement ID: H398-LLD-ANA-FNC-187

The function shall perform as follows:

a) Set pData to pData plus M\_ZERO

b) Loop infinetly to perform the following:

* Call OsSemPend to signal A825comm task Semaphore with parameters (reference of A825comm task Semaphore, M\_ZERO and reference of error).

##### 10.2.1.7.2 dauanaa825comm-A825CommTask-LLR-002

Requirement ID: H398-LLD-ANA-FNC-188

The function shall loop infinite times and set A825 filter error flag to (A825 filter error flag added with M\_ONE) when return value of A825Receive with parameters (EEC, reference of message) is not equal to A825\_RX\_QUEUE\_EMPTY otherwise do nothing.

##### 10.2.1.7.3 dauanaa825comm-A825CommTask-LLR-003

Requirement ID: H398-LLD-ANA-FNC-189

The function shall loop infinite times and perform as follows:

set A825 filter error flag to (A825 filter error flag added with return value of AppMessageNOC with parameter reference of message) when the below conditions are satisfied:

1. return value of A825Receive with parameters (NOC, reference of message) is not equal to A825\_RX\_QUEUE\_EMPTY, otherwise do nothing.
2. sfid of bid of sid of message is equal to M\_HOWELL\_DAU\_FID.

##### 10.2.1.7.4 dauanaa825comm-A825CommTask-LLR-004

Requirement ID: H398-LLD-ANA-FNC-190

The function shall loop infinite times and perform as follows:

set A825 filter error flag to (A825 filter error flag added with M\_ONE) when the below conditions are satisfied:

1. return value of A825Receive with parameters (NOC, reference of message) is not equal to A825\_RX\_QUEUE\_EMPTY, otherwise do nothing.
2. sfid of bid of sid of message is not equal to M\_HOWELL\_DAU\_FID.

##### 10.2.1.7.5 dauanaa825comm-A825CommTask-LLR-005

Requirement ID: H398-LLD-ANA-FNC-191

The function shall loop infinite times and perform as follows:

set A825 filter error flag to (A825 filter error flag added with return value of AppMessageNSC with parameter reference of message) when the below conditions are satisfied:

1. return value of A825Receive with parameters (NSC, reference of message) is not equal to A825\_RX\_QUEUE\_EMPTY, otherwise do nothing.
2. sid of pid of sid of message is equal to ANALOG\_SID logical OR with sid of pid of sid of message is equal to MULTICAST\_SID logical AND with smt of pid of sid of message is equal to M\_ONE.

##### 10.2.1.7.6 dauanaa825comm-A825CommTask-LLR-006

Requirement ID: H398-LLD-ANA-FNC-192

The function shall loop infinite times and perform as follows:

set A825 filter error flag (A825 filter error flag added with M\_ONE) when the below conditions are satisfied:

1. return value of A825Receive with parameters (NSC, reference of message) is not equal to A825\_RX\_QUEUE\_EMPTY, otherwise do nothing.
2. sid of bid of sid of message is not equal to ANALOG\_SID AND sid of pid of sid of message is not equal to MULTICAST\_SID OR smt of pid of sid of message is not equal to M\_ONE.

##### 10.2.1.7.7 dauanaa825comm-A825CommTask-LLR-007

Requirement ID: H398-LLD-ANA-FNC-193

The function shall loop infinite times and set A825 filter error flag to (A825 filter error flag added with M\_ONE) when return value of A825Receive with parameters (TMC, reference of message) is not equal to A825\_RX\_QUEUE\_EMPTY otherwise do nothing.

### 10.2.2 A825CommInit

Low Level Design Details about CSU A825CommInit will follow in the sub sections.

#### 10.2.2.1 Brief Description

The function A825CommInit initializes ARINC825 Communication Task and Stack to handle Gateway commands coming across the ARINC825 Bus.

#### 10.2.2.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.2.2.3 List of global variables accessed and modified

Accessed: None

Modified: A825\_task\_stack

#### 10.2.2.4 Parameter list (Input/Output)

Inputs: None

Outputs: None

#### 10.2.2.5 Return Value

None

#### 10.2.2.6 Other CSUs called by this CSU

OsSemCreate

TbaseTaskSignaling

OsTaskCreate

ErrorHandler

#### 10.2.2.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to A825CommInit.

##### 10.2.2.7.1 dauanaa825comm-A825CommInit-LLR-001

Requirement ID: H398-LLD-ANA-FNC-202

The function shall perform as follows:

1. Set return value of OsSemCreate with parameter M\_ZERO to A825 task semaphore.
2. Call the function ErrorHandler when M\_NULL is equal to A825 task semaphore otherwise do nothing.

##### 10.2.2.7.2 dauanaa825comm-A825CommInit-LLR-003

Requirement ID: H398-LLD-ANA-FNC-204

The function shall call ErrorHandler when M\_OS\_NO\_ERR is not equal to return value of OsTaskCreate with parameters,

* Pointer to the A825CommTask's address.
* M\_NULL pointer passed as argument to the A825CommTask.
* A825CommTask's top-of-stack size i.e. M\_A8252\_TASK\_STK\_SIZE.
* Task Priority M\_A8252\_TASK\_PRIO.

Otherwise, do nothing.

##### 10.2.2.7.3 dauanaa825comm-A825CommInit-LLR-002

Requirement ID: H398-LLD-ANA-FNC-2449

The function shall call TbaseTaskSignaling with parameters M\_A8252\_TASK\_TICKS and Sem\_a825\_task.

## 10.3 dauanaapp

This module contains Application function to call other routines.

### 10.3.1 AppGetA825RCI

Low Level Design Details about CSU AppGetA825RCI will follow in the sub sections.

#### 10.3.1.1 Brief Description

The function returns the Analog Board RCI value

#### 10.3.1.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.3.1.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 10.3.1.4 Parameter list (Input/Output)

Inputs : None

Outputs : None

#### 10.3.1.5 Return Value

T\_UINT8 - Return the Analog board RCI

#### 10.3.1.6 Other CSUs called by this CSU

None

#### 10.3.1.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to AppGetA825RCI.

##### 10.3.1.7.1 dauanaapp-AppGetA825RCI-LLR-001

Requirement ID: H398-LLD-ANA-FNC-214

The function shall returns the analog board RCI.

### 10.3.2 ReadChannelBIT

Low Level Design Details about CSU ReadChannelBIT will follow in the sub sections.

#### 10.3.2.1 Brief Description

The function checks the BIT reading data for the analog as per the channel selected and return the status.

#### 10.3.2.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.3.2.3 List of global variables accessed and modified

Accessed : Ptr\_sensor\_aas\_analog

Modified : None

#### 10.3.2.4 Parameter list (Input/Output)

Inputs : T\_UINT8 u8Chan ->Channel number to be read

Outputs : None

#### 10.3.2.5 Return Value

T\_BOOLEAN - Return TRUE if BIT read data is in range

Return FALSE if BIT read data is not in range

#### 10.3.2.6 Other CSUs called by this CSU

XADCRead

LookupTableLookup

#### 10.3.2.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to ReadChannelBIT.

##### 10.3.2.7.1 dauanaapp-ReadChannelBIT-LLR-001

Requirement ID: H398-LLD-ANA-FNC-1860

The function shall calculate the average of ADC reading by calling the function 'XADCRead' with parameter (u8Chan, XADC\_MUX\_1).

##### 10.3.2.7.2 dauanaapp-ReadChannelBIT-LLR-002

Requirement ID: H398-LLD-ANA-FNC-1861

The function shall calculate the Lookup scale reading of the BIT by calling the function 'LookupTableLookup' with parameter (address of s\_Table of Sensor\_aas\_analog with indices u8Chan and XADC\_MUX\_1 of Ptr\_sensor\_aas\_analog)), average reading value of ADC, address to store the status).

Note: Refer to MCD requirement for more information on Sensor data.

##### 10.3.2.7.3 dauanaapp-ReadChannelBIT-LLR-003

Requirement ID: H398-LLD-ANA-FNC-1862

The function shall return FALSE when the return value of function 'LookupTable’ is less than the Sensor fault min boundary (i.e i16\_Min\_Range of Sensor\_aas\_analog with indices u8Chan and XADC\_MUX\_1 of Ptr\_sensor\_aas\_analog) or greater than the Sensor fault max boundary (i.e i16\_Max\_Range of Sensor\_aas\_analog with indices u8Chan and XADC\_MUX\_1 of Ptr\_sensor\_aas\_analog) Otherwise return TRUE.

Note: Refer to MCD requirement for more information on Sensor data.

### 10.3.3 ScaleChannel

Low Level Design Details about CSU ScaleChannel will follow in the sub sections.

#### 10.3.3.1 Brief Description

The function ScaleChannel calculates the scaled reading for the analog sensor as per the channel selected.

#### 10.3.3.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.3.3.3 List of global variables accessed and modified

Accessed : Ptr\_sensor\_aas\_analog, Excitation\_off, Cal\_aasdata

Modified : None

#### 10.3.3.4 Parameter list (Input/Output)

Inputs : T\_UINT8 u8Chan ->Channel no of the ADC (Channel no - 0 to 6 )

T\_UINT8 u8Mux ->MUX channel no of the ADC (MUX Channel no - 0 to 16 )

Outputs : T\_UINT8 \*pu8Stat -> Sensor Status flag

#### 10.3.3.5 Return Value

T\_SINT16 - Return the scaled value of the read analog sensor

- Return ZERO if the selected channel state is DISABLE

#### 10.3.3.6 Other CSUs called by this CSU

ReadChannelBIT

XADCRead

LookupTableLookup

ScaleColdJunction

LookupTableInverse

#### 10.3.3.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to ScaleChannel.

##### 10.3.3.7.1 dauanaapp-ScaleChannel-LLR-001

Requirement ID: H398-LLD-ANA-FNC-225

The function shall set Sensor Status flag (pu8Stat) to APP\_CHAN\_OK and return M\_ZERO when the Sensor channel state (i.e u8ChanState of Sensor\_aas\_analog with indices u8Chan and u8Mux of Ptr\_sensor\_aas\_analog) is DISABLE, Otherwise Do nothing.

Note: Refer to MCD requirement for more information on Sensor data.

##### 10.3.3.7.2 dauanaapp-ScaleChannel-LLR-002

Requirement ID: H398-LLD-ANA-FNC-226

The function shall

1. Read BIT channel by calling the function ‘ReadChannelBIT' with parameter (u8Chan).
2. Set Sensor Status flag (pu8Stat) to APP\_CHAN\_ERROR and Return default value of the sensor (i.e i16DefaultVal of Sensor\_aas\_analog with indices u8Chan and u8Mux of Ptr\_sensor\_aas\_analog) when the function 'ReadChannelBIT' return FALSE, Otherwise Do nothing.

Note: Refer to MCD requirement for more information on Sensor data.

##### 10.3.3.7.3 dauanaapp-ScaleChannel-LLR-003

Requirement ID: H398-LLD-ANA-FNC-227

The function shall

1. Calculate the average of ADC reading by calling the function 'XADCRead' with parameter u8Chan, u8Mux.
2. Store ADC Negative reading to average of ADC reading less than M\_ZERO.
3. Set Overflow to FALSE.

##### 10.3.3.7.4 dauanaapp-ScaleChannel-LLR-004

Requirement ID: H398-LLD-ANA-FNC-228

The function shall do the following when Thermocouple Channel on the Channel 1 and Mux 3 or 4(Tck1 & Tck2) (i.e. (u8Chan is M\_ZERO) AND ((u8Mux is M\_THREE) OR (u8Mux is M\_FOUR))):

1)Gets the Cold Junction Value by calling function ‘ScaleColdJunction’ with parameter (Zero, address to store the status).

2)Swaps the rows and columns of lookup table by calling the function ‘LookupTableInverse’ with parameter (Reference to External ADC Sensor table (i.e(pisTable of Sensor\_aas\_analog with indices u8Chan and u8Mux of Ptr\_sensor\_aas\_analog) , Cold junction value, address to store the status) and store return value of ‘LookupTableInverse’ in Cold Junction Value.

3) Adds the Average ADC reading to cold junction ADC readings and store back in temporary Average ADC reading.

4) Set the overflow to TRUE when Temporary Average ADC reading less than Average ADC reading is not equal to cold junction ADC readings less than M\_ZERO.

5) Set Average ADC reading to Temporary Average ADC reading.

##### 10.3.3.7.5 dauanaapp-ScaleChannel-LLR-011

Requirement ID: H398-LLD-ANA-FNC-2434

The function shall set the RTD channel Index to M\_ZERO when channel is equal to XADC\_CHAN\_1 AND mux channel is equal to XADC\_MUX\_8.

##### 10.3.3.7.6 dauanaapp-ScaleChannel-LLR-012

Requirement ID: H398-LLD-ANA-FNC-2435

The function shall set the RTD channel Index to M\_ONE when channel is equal to XADC\_CHAN\_2 AND mux channel is equal to XADC\_MUX\_2.



##### 10.3.3.7.7 dauanaapp-ScaleChannel-LLR-013

 Requirement ID: H398-LLD-ANA-FNC-2436

The function shall set the RTD channel Index to M\_FIVE when channel is equal to XADC\_CHAN\_2 AND mux channel is equal to XADC\_MUX\_6.

##### 10.3.3.7.8 dauanaapp-ScaleChannel-LLR-014

Requirement ID: H398-LLD-ANA-FNC-2437

The function shall set the RTD channel Index to M\_SIX when channel is equal to XADC\_CHAN\_2 AND mux channel is equal to XADC\_MUX\_7.

##### 10.3.3.7.9 dauanaapp-ScaleChannel-LLR-015

 Requirement ID: H398-LLD-ANA-FNC-2438

The function shall set the RTD channel Index to M\_EIGHT when channel is equal to XADC\_CHAN\_3 AND mux channel (u8Mux) is equal to XADC\_MUX\_2.

##### 10.3.3.7.10 dauanaapp-ScaleChannel-LLR-005

Requirement ID: H398-LLD-ANA-FNC-229

The function shall do the following When (Excitation\_off is equal to TRUE AND (XADC\_CHAN\_1 is equal to u8Chan AND XADC\_MUX\_8 is equal to u8Mux) OR XADC\_CHAN\_2 is equal to u8Chan OR (XADC\_CHAN\_3 is equal to u8Chan AND XADC\_MUX\_2 is equal to u8Mux)).

1. Set offset reading to i16\_Offset of calRTDOffside with index of (RTD channel Index) of Cal\_aasdata)
2. Set Gain to f32\_Gain of calRTDOffside with index of (RTD channel Index ) of Cal\_aasdata

Otherwise

1. Set offset reading to i16\_Offset of cal\_xadc with index of (u8Chan and u8Mux) of Cal\_aasdata)
2. Set Gain to f32\_Gain of cal\_xadc with index of (u8Chan and u8Mux) of Cal\_aasdata.

Set Average ADC reading to Zero when Average ADC reading is equal to -1 and Gain is less than 1.0

Set Temporary Average ADC reading to product of (Gain and Average ADC reading).

##### 10.3.3.7.11 dauaanapp-ScaleChannel-LLR-016

Requirement ID: H398-LLD-ANA-FNC-2441

The function shall

1. Set overflow to TRUE when (((Temporary Average ADC reading less than M\_ZERO) is not equal to ((Average ADC reading less than M\_ZERO) AND (gain is greater than or equal to M\_ZERO)) OR (Temporary Average ADC reading less than M\_ZERO) is equal to ((Average ADC reading less than M\_ZERO) AND (gain is less than M\_ZERO)))
2. Set Average ADC reading to Temporary Average ADC reading
3. Set Temporary Average ADC reading to addtion of (Average ADC reading and offset)

##### 10.3.3.7.12 dauanaapp-ScaleChannel-LLR-017

Requirement ID: H398-LLD-ANA-FNC-2446

The function shall

1. Set overflow to TRUE when ((Temporary Average ADC reading less than Average ADC reading) is not equal to (offset is less than M\_ZERO))
2. Set Average ADC reading to Temporary Average ADC reading.

##### 10.3.3.7.11 dauanaapp-ScaleChannel-LLR-006

Requirement ID: H398-LLD-ANA-FNC-230

The function shall do the following when overflow is equal to TRUE

1. Set pu8Stat value to APP\_RANGE\_ERROR
2. Set status to FALSE.
3. Set Average ADC reading to 0x8000 when ADC Negative reading is equal to TRUE otherwise set to 0x7FFF.
4. calculate the actual ADC reading by calling the function ‘LookupTableLookup’with parameters (Reference to External ADC Sensor table (i.e(sTable of Sensor\_aas\_analog with indices u8Chan and u8Mux of Ptr\_sensor\_aas\_analog), Calculated ADC reading, Status).

Otherwise

calculate the actual ADC reading by calling the function ‘LookupTableLookup’with parameters (Reference to External ADC Sensor table (i.e(sTable of Sensor\_aas\_analog with indices u8Chan and u8Mux of Ptr\_sensor\_aas\_analog, Calculated ADC reading, Status).

Note: Refer to MCD requirement for more information on Sensor data.

##### 10.3.3.7.12 dauanaapp-ScaleChannel-LLR-007

Requirement ID: H398-LLD-ANA-FNC-232

The function shall do the following when (FALSE is equal to Status) Logically AND with (actual ADC reading for the selected sensor is equal to (pi16Y of sTable of Sensor\_aas\_analog with indices u8Chan and u8Mux of Ptr\_sensor\_aas\_analog added with M\_ONE)) Logically OR with (actual ADC reading for the selected sensor is less than the Min range of the selected sensor (i.e i16MinRange of Sensor\_aas\_analog with indices u8Chan and u8Mux of Ptr\_sensor\_aas\_analog):

1. Set the actual ADC reading to the Default value of the selected sensor (i.e i16DefaultVal of Sensor\_aas\_analog with indices u8Chan and u8Mux of Ptr\_sensor\_aas\_analog) when Default value state of the selected Sensor (i.e u8DefaultState of Sensor\_aas\_analog with indices u8Chan and u8Mux of Ptr\_sensor\_aas\_analog) is ENABLE.
2. Set the actual ADC reading to the Min Range value of the selected Sensor (i.e i16MinRangeof Sensor\_aas\_analog with indices u8Chan and u8Mux of Ptr\_sensor\_aas\_analog) when Default value state of selected Sensor (i.e u8DefaultState of Sensor\_aas\_analog with indices u8Chan and u8Mux of Ptr\_sensor\_aas\_analog) is not ENABLE.
3. Set the Sensor Status flag(pu8Stat) to APP\_RANGE\_ERROR,

Note: Refer to MCD requirement for more information on Sensor data.

##### 10.3.3.7.13 dauanaapp-ScaleChannel-LLR-008

Requirement ID: H398-LLD-ANA-FNC-233

The function shall do the following when (FALSE is equal to Status) Logically AND with (actual ADC reading for the selected sensor is equal to content of (content of pi16Y of sTable of Sensor\_aas\_analog with indices u8Chan and u8Mux of Ptr\_sensor\_aas\_analog added with (pi16Y of sTable of Sensor\_aas\_analog with indices u8Chan and u8Mux of Ptr\_sensor\_aas\_analog))) Logically OR with actual ADC reading for the selected sensor is greater than the Max range of the selected sensor (i.e i16MaxRange of Sensor\_aas\_analog with indices u8Chan and u8Mux of Ptr\_sensor\_aas\_analog):

1. Set the actual ADC reading to the Default value of the selected sensor (i.e i16DefaultVal of Sensor\_aas\_analog with indices u8Chan and u8Mux of Ptr\_sensor\_aas\_analog) when Default value state of the selected Sensor (i.e u8DefaultState of Sensor\_aas\_analog with indices u8Chan and u8Mux of Ptr\_sensor\_aas\_analog) is ENABLE.
2. Set the actual ADC reading to the Max range value of the selected Sensor (i.e i16MaxRange of Sensor\_aas\_analog with indices u8Chan and u8Mux of Ptr\_sensor\_aas\_analog) when Default value state of selected Sensor (i.e u8DefaultState of Sensor\_aas\_analog with indices u8Chan and u8Mux of Ptr\_sensor\_aas\_analog) is not ENABLE
3. Set the Sensor Status flag to APP\_RANGE\_ERROR

Note: Refer to MCD requirement for more information on Sensor data.

##### 10.3.3.7.14 dauanaapp-ScaleChannel-LLR-009

Requirement ID: H398-LLD-ANA-FNC-234

The function shall Set the Sensor Status flag (pu8Stat) to APP\_CHAN\_OK when the actual ADC reading for the selected sensor is in Range (i.e not is less than the Min range of the selected sensor and not greater than the Max range of the selected sensor).

##### 10.3.3.7.15 dauanaapp-ScaleChannel-LLR-010

Requirement ID: H398-LLD-ANA-FNC-235

The function shall Return the calculated ADC reading.

### 10.3.4 ScaleColdJunction

Low Level Design Details about CSU ScaleColdJunction will follow in the sub sections.

#### 10.3.4.1 Brief Description

The function ScaleColdJunction calculate the scaled reading for the Cold Junction sensor as per the channel selected.

#### 10.3.4.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.3.4.3 List of global variables accessed and modified

Accessed : Ptr\_sensor\_aas\_analog

Modified : None

#### 10.3.4.4 Parameter list (Input/Output)

Inputs : T\_UINT8 u8Chan ->Channel no of the Cold Junction sensor

Outputs : T\_UINT8 \* pu8\_status -> Sensor Status flag

#### 10.3.4.5 Return Value

T\_SINT16- Return the scaled value of the read Cold Junction sensor

- Return ZERO if the selected channel state is DISABLE

#### 10.3.4.6 Other CSUs called by this CSU

IadcRead

LookupTableLookup

#### 10.3.4.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to ScaleColdJunction.

##### 10.3.4.7.1 dauanaapp-ScaleColdJunction-LLR-001

Requirement ID: H398-LLD-ANA-FNC-244

The function shall set Sensor Status flag (pu8\_status) to APP\_CHAN\_OK and return M\_ZERO when the selected Internal ADC channel state (i.e u8ChanState of Sensor\_as\_coldjunction with index u8Chan of Ptr\_sensor\_aas\_analog) is DISABLE, Otherwise Do nothing.

Note: Refer to MCD requirement for more information on Sensor data.

##### 10.3.4.7.2 dauanaapp-ScaleColdJunction-LLR-002

Requirement ID: H398-LLD-ANA-FNC-245

The function shall calculate the average of Internal ADC reading by calling the function 'IadcRead' with parameter (u8Chan).

##### 10.3.4.7.3 dauanaapp-ScaleColdJunction-LLR-003

Requirement ID: H398-LLD-ANA-FNC-246

The function shall calculate the actual Internal ADC reading by calling the function 'LookupTableLookup' with parameter (Reference to Cold Junction Sensor table (i.e pi16X of sTable of Sensor\_as\_coldjunction with index u8Chan of Ptr\_sensor\_aas\_analog, pi16Y of sTable of Sensor\_as\_coldjunction with index u8Chan of Ptr\_sensor\_aas\_analog), calculated average of Internal ADC reading, reference to Status).

Note: Refer to MCD requirement for more information on Sensor data.

##### 10.3.4.7.4 dauanaapp-ScaleColdJunction-LLR-004

Requirement ID: H398-LLD-ANA-FNC-247

The function shall do the following when the actual Internal ADC reading for the cold junction is less than the Min range of the selected sensor for the cold junction (i.e i16MinRange of Sensor\_as\_coldjunction with index u8Chan of Ptr\_sensor\_aas\_analog):

1. Set the actual Internal ADC reading to the Default value of the cold junction (i.e i16DefaultVal of Sensor\_as\_coldjunction with index u8Chan of Ptr\_sensor\_aas\_analog) when the Default state of cold junction (i.e u8DefaultState of Sensor\_as\_coldjunction with index u8Chan of Ptr\_sensor\_aas\_analog) is ENABLE.
2. Set the actual Internal ADC reading to the Min Range value of the cold junction (i.e i16MinRange of Sensor\_as\_coldjunction with index u8Chan of Ptr\_sensor\_aas\_analog) when the Default state of cold junction (i.e u8DefaultState of Sensor\_as\_coldjunction with index u8Chan of Ptr\_sensor\_aas\_analog) is not ENABLE.
3. Set the Sensor Status flag(pu8\_status) to APP\_RANGE\_ERROR.

Note: Refer to MCD requirement for more information on Sensor data.

##### 10.3.4.7.5 dauanaapp-ScaleColdJunction-LLR-005

Requirement ID: H398-LLD-ANA-FNC-248

The function shall do the following when the actual Internal ADC reading for the cold junction is greater than the Max range of the selected sensor for the cold junction (i.e i16MaxRange of Sensor\_as\_coldjunction with index u8Chan of Ptr\_sensor\_aas\_analog):

1. Set the actual Internal ADC reading to the Default value of the cold junction (i.e i16DefaultVal of Sensor\_as\_coldjunction with index u8Chan of Ptr\_sensor\_aas\_analog) when the Default value state of cold junction (i.e u8DefaultState of Sensor\_as\_coldjunction with index u8Chan of Ptr\_sensor\_aas\_analog) is ENABLE
2. Set the actual Internal ADC reading to the Max range value of the cold junction (i.e i16MaxRange of Sensor\_as\_coldjunction with index u8Chan of Ptr\_sensor\_aas\_analog) when the Default value state of cold junction (i.e u8DefaultState of Sensor\_as\_coldjunction with index u8Chan of Ptr\_sensor\_aas\_analog) is not ENABLE.
3. Set the Sensor Status flag(pu8\_status) to APP\_RANGE\_ ERROR

Note: Refer to MCD requirement for more information on Sensor data.

##### 10.3.4.7.6 dauanaapp-ScaleColdJunction-LLR-006

Requirement ID: H398-LLD-ANA-FNC-249

The function shall Set the Sensor Status flag(pu8\_status) to APP\_CHAN\_OK when the actual Internal ADC reading for the cold junction is in Range (i.e not less than the Min range of the selected sensor for the cold junction and not greater than the Max range of the selected sensor for the cold junction).

##### 10.3.4.7.7 dauanaapp-ScaleColdJunction-LLR-007

Requirement ID: H398-LLD-ANA-FNC-250

The function shall Return the calculated Internal ADC reading for the cold junction.

### 10.3.5 SendReadings

Low Level Design Details about CSU SendReadings will follow in the sub sections.

#### 10.3.5.1 Brief Description

The function SendReadings transmits all the sensor reading through A825.

#### 10.3.5.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.3.5.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 10.3.5.4 Parameter list (Input/Output)

Inputs : None

Outputs : None

#### 10.3.5.5 Return Value

None

#### 10.3.5.6 Other CSUs called by this CSU

UtilsCopy

A825Transmit

10.3.5.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to SendReadings.

##### 10.3.5.7.1 dauanaapp-SendReadings-LLR-002

Requirement ID: H398-LLD-ANA-FNC-260

The function shall Send all the Low Level readings through A825 by doing the following

- Loop through all the Low Level Voltage Inputs (M\_ZERO to M\_APP\_MAX\_DOC\_CHANNEL-1)

- Save (M\_ANALOG\_START (ALOG) added to each low level channel index) to Data Object Code in broad cast id of the A825 Tx message

- Copy reading into payload by calling the function 'UtilsCopy' with parameter (Payload contents, reference to ai16\_Anlog of sXadc of communication data structure, M\_A825\_PAYLOAD\_SIZE).

- Transmit message by calling the function 'A825Transmit' with parameter (Reference to the Tx message)

##### 10.3.5.7.2 dauanaapp-SendReadings-LLR-003

Requirement ID: H398-LLD-ANA-FNC-261

The function shall do the following:

- Loop through all the Low Level Voltage Inputs (M\_ZERO to M\_TWO-1)

-Save (M\_ANLAOG\_STAT\_START (ALOG\_STATS) added to each low level channel index) to Data Object Code in broad cast id of the A825 Tx message.

-Copy the low level channel status into payload by calling function 'UtilsCopy' with parameter (Payload contents, reference to au8\_Status of sXadc of communication data structure, M\_A825\_PAYLOAD\_SIZE).

-Transmit the message (status for each low level channel) by calling the function 'A825Transmit' with parameter (Reference to the Tx message).

### 10.3.6 SendCrcPn

Low Level Design Details about CSU SendCrcPn will follow in the sub sections.

#### 10.3.6.1 Brief Description

The function SendCrcPn responds to the ACQUIRE\_CRC\_PN command.

It transmits Partnumber and CRCs of Application and configuration.

#### 10.3.6.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.3.6.3 List of global variables accessed and modified

Accessed : U32\_checksum\_flight

U32\_checksum\_config

Ptr\_sensor\_aas\_analog

Modified : None

#### 10.3.6.4 Parameter list (Input/Output)

None

#### 10.3.6.5 Return Value

None

#### 10.3.6.6 Other CSUs called by this CSU

A825Transmit

#### 10.3.6.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to SendCrcPn.

##### 10.3.6.7.1 dauanaapp-SendCrcPn-LLR-001

Requirement ID: H398-LLD-ANA-FNC-271

The function shall transmit the Application CRC, Part number and Configuration CRC, along with the below mentioned fields of an A825 NOC message by calling the function 'A825Transmit' with parameter 'reference to Arinc 825 message'

- lcc to NOC

- sfid to M\_HOWELL\_DAU\_FID

- rsd to M\_ZERO

- lcl to M\_ONE

- pvt to M\_ONE

- doc to DOC\_ID(its corresponding DATA is as mentioned in the table below)

- rci to received RCI

- payload size to M\_EIGHT bytes.

- payload to DATA

Table: Analog Part number and CRC

Table : Analog Part number and CRC

|  |  |
| --- | --- |
| **DOC\_ID** | **DATA** |
| M\_ANALOG\_PN\_START | Revno\_id[0]..[7] (Bootloader Part number first 8 bytes) |
| M\_ANALOG\_PN\_START plus M\_ONE | Revno\_id[8]..[15] (Bootloader Part number second 8 bytes) |
| M\_ANALOG\_PN\_START plus M\_TWO | payload [0] with (crc word bitwise AND M\_EXTRACT\_LSB)  payload [1] with ((crc word shifted to right by M\_SHIFT\_BY\_8) bitwise AND M\_EXTRACT\_LSB),  payload [2] with ((crc word shifted to right by M\_SHIFT\_BY\_16) bitwise AND M\_EXTRACT\_LSB),  payload [3] with (crc word shifted to right by M\_SHIFT\_BY\_24),  payload [4...7] with M\_ZERO  (Bootloader CRC) |
| M\_ANALOG\_PN\_START plus M\_THREE | Revno\_id[0]..[7] (Analog Flight application Part number first 8 bytes) |
| M\_ANALOG\_PN\_START plus M\_FOUR | Revno\_id[8]..[15] (Analog Flight application Part number Second 8 bytes) |
| M\_ANALOG\_PN\_START plus M\_FIVE | payload [0] with (U32\_checksum\_flight bitwise AND M\_EXTRACT\_LSB),  payload [1] with ((U32\_checksum\_flight shifted to right by M\_SHIFT\_BY\_8) bitwise AND M\_EXTRACT\_LSB),  payload [2] with ((U32\_checksum\_flight shifted to right by M\_SHIFT\_BY\_16) bitwise AND M\_EXTRACT\_LSB),  payload [3] with (U32\_checksum\_flight shifted to right by M\_SHIFT\_BY\_24), payload [4...7] with M\_ZERO  (Analog Flight application CRC) |
| M\_ANALOG\_PN\_START plus M\_SIX | part\_rev\_no of Ptr\_sensor\_aas\_analog [0]...[7] (Analog MCD Part number first 8 bytes) |
| M\_ANALOG\_PN\_START plus M\_SEVEN | part\_rev\_no of Ptr\_sensor\_aas\_analog [8] ... [15] (Analog MCD Part number second 8 bytes) |
| M\_ANALOG\_PN\_START plus M\_EIGHT | payload [0] with (U32\_checksum\_config bitwise AND M\_EXTRACT\_LSB), payload [1] with ((U32\_checksum\_config shifted to right by M\_SHIFT\_BY\_8) bitwise AND M\_EXTRACT\_LSB),  payload [2] with ((U32\_checksum\_config shifted to right by M\_SHIFT\_BY\_16) bitwise AND M\_EXTRACT\_LSB),  payload [3] with (U32\_checksum\_config shifted to right by M\_SHIFT\_BY\_24),  payload [4...7] with M\_ZERO  (Analog MCD CRC) |
| M\_ANALOG\_PN\_START plus M\_NINE | Revno\_id[0]..[7] (Boot Config Part number first 8 bytes) |
| M\_ANALOG\_PN\_START plus M\_TEN | Revno\_id[8]..[15] (Boot Config Part number second 8 bytes) |
| M\_ANALOG\_PN\_START plus M\_ELEVEN | payload [0] with (crc word bitwise AND M\_EXTRACT\_LSB)  payload [1] with ((crc word shifted to right by M\_SHIFT\_BY\_8) bitwise AND M\_EXTRACT\_LSB),  payload [2] with ((crc word shifted to right by M\_SHIFT\_BY\_16) bitwise AND M\_EXTRACT\_LSB),  payload [3] with (crc word shifted to right by M\_SHIFT\_BY\_24),  payload [4...7] with M\_ZERO  (Boot Config CRC) |
| M\_ANALOG\_PN\_START plus M\_FIFTEEN | Revno\_id[0]..[7] (Software loader Part number first 8 bytes) |
| M\_ANALOG\_PN\_START plus M\_SIXTEEEN | Revno\_id[8]..[15] (Software loader Part number second 8 bytes) |
| M\_ANALOG\_PN\_START plus M\_SEVENTEEN | payload [0] with (crc word bitwise AND M\_EXTRACT\_LSB)  payload [1] with ((crc word shifted to right by M\_SHIFT\_BY\_8) bitwise AND M\_EXTRACT\_LSB),  payload [2] with ((crc word shifted to right by M\_SHIFT\_BY\_16) bitwise AND M\_EXTRACT\_LSB),  payload [3] with (crc word shifted to right by M\_SHIFT\_BY\_24),  payload [4...7] with M\_ZERO  (Software loader CRC) |
| M\_ANALOG\_PN\_START plus M\_EIGHTEEN | Revno\_id[0]..[7] (Calibration Part number first 8 bytes) |
| M\_ANALOG\_PN\_START plus M\_NINTEEN | Revno\_id[8]..[15] (Calibration Part number second 8 bytes) |
| M\_ANALOG\_PN\_START plus M\_TWENTY | payload [0] with (crc word bitwise AND M\_EXTRACT\_LSB)  payload [1] with ((crc word shifted to right by M\_SHIFT\_BY\_8) bitwise AND M\_EXTRACT\_LSB),  payload [2] with ((crc word shifted to right by M\_SHIFT\_BY\_16) bitwise AND M\_EXTRACT\_LSB),  payload [3] with (crc word shifted to right by M\_SHIFT\_BY\_24),  payload [4...7] with M\_ZERO  (Calibration CRC) |

### 10.3.7 ScaleReadings

Low Level Design Details about CSU ScaleReadings will follow in the sub sections.

#### 10.3.7.1 Brief Description

The function ScaleReadings scales readings of all the sensor.

#### 10.3.7.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.3.7.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 10.3.7.4 Parameter list (Input/Output)

Inputs : None

Outputs : None

#### 10.3.7.5 Return Value

None

#### 10.3.7.6 Other CSUs called by this CSU

ScaleChannel

ScaleColdJunction

#### 10.3.7.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to ScaleReadings.

##### 10.3.7.7.1 dauanaapp-ScaleReadings-LLR-001

Requirement ID: H398-LLD-ANA-FNC-280

The function shall Scale the Readings of all the Low Level Channels by looping through all the Low Level channels (M\_ZERO to 40 minus one)

* Read the Scaled value for Each LL Channels by calling the function ‘ScaleChannel' with parameter (ADC channel no, ADC Mux no, reference to Status)
* Increments the ADC Mux number.
* The function performs the following when ADC Mux is greater than XADC\_MUX\_CHANS-M\_ONE:
* *Reset the mux for the next channel.*
* *Increments the ADC channel.*
* Clear status bits of All the channel (i.e. status bits of the APP Comm data list bitwise AND with (Negation of (M\_APP\_STATUS\_MASK bit shifted to left by ((reminder of loop counter and M\_FOUR) multiplied by M\_APP\_STATUS\_BITS))))
* Save the status of All the channel (i.e. status bits of the APP Comm data list bitwise OR with (status bit shifted to left by ((reminder of loop counter and M\_FOUR) multiplied by M\_APP\_STATUS\_BITS))).
* Increments the buffer and status buffer.
* Increment message status and reset status buffer when Status Buffer divided by M\_FOUR is greater than or equal to M\_EIGHT.

##### 10.3.7.7.2 dauanaapp-ScaleReadings-LLR-002

Requirement ID: H398-LLD-ANA-FNC-281

The function shall reset the ADC mux number, ADC channel.

* Increment the buffer and status buffer by 3
* The loop index is looping from APP\_TACH\_1 to APP\_TACH\_CHANS minus one
* Set ai16\_Analog with index buffer of sXadc of App\_s\_comm to function ScaleTach with parameters loop counter and status
* Clear status bits of All the channel (i.e au8\_Status with indexes status messages and status buffer divided by M\_FOUR of sXadc of App\_s\_comm set to au8\_Status with indexes status messages and status buffer divided by M\_FOUR of sXadc of App\_s\_comm bitwise AND with (negation of (M\_APP\_STATUS\_MASK left shift (buffer percentage M\_FOUR) multiply with M\_APP\_STATUS\_BITS))))
* Save status of the channel (i.e au8\_Status with indexes status messages and status buffer divided by M\_FOUR of sXadc of App\_s\_comm set to au8\_Status with indexes status messages and status buffer divided by M\_FOUR of sXadc of App\_s\_comm bitwise OR with (status left shift by ((buffer percentage M\_FOUR) multiply M\_APP\_STATUS\_BITS)))
* Increments the buffer and status buffer.
* Increments status messages and set status buffer to 0 when status buffer divided by M\_FOUR greater than or equal to 8.

##### 10.3.7.7.3 dauanaapp-ScaleReadings-LLR-003

Requirement ID: H398-LLD-ANA-FNC-2326

The function shall perform the following:

* Set Pwm\_timer\_count\_limit to M\_TWENTY\_LAKH divided by ai16\_Analog with index M\_FORTYNINE of sXadc of App\_s\_comm when division of ai16\_Analog with index M\_FORTYNINE of sXadc of App\_s\_comm and M\_TEN is greater than M\_ZERO.
* Set Pwm\_timer\_count\_limit M\_TWENTY\_LAKH divided by ai16\_Analog with index M\_FIFTYONEof sXadc of App\_s\_comm when division of ai16\_Analog with index M\_FORTYNINE of sXadc of App\_s\_comm and M\_TEN is less than or equal to M\_ZERO and division of ai16\_Analog with index M\_FIFTYONEof sXadc of App\_s\_comm and M\_TEN is greater than M\_ZERO.

##### 10.3.7.7.4 dauanaapp-ScaleReadings-LLR-004

Requirement ID: H398-LLD-ANA-FNC-2327

The function shall perform the following:

* Set u8\_buffer to 41.
* Set u8\_stat\_mess to 1.
* Set u8\_stat\_buffer to 9.
* Perform the following until M\_NUM\_OF\_PWM\_INPUTS minus M\_ONE
* Set ai16\_Analog with index u8\_buffer of sXadc of App\_s\_comm by return of ScalePwmc with parameters u8\_loop\_ctr and address of u8\_status.
* Set au8\_Status with index u8\_stat\_mess and division of u8\_stat\_buffer and M\_FOUR of sXadc of App\_s\_comm to au8\_Status with index u8\_stat\_mess and division of u8\_stat\_buffer and M\_FOUR of sXadc of App\_s\_comm bitwise AND with (nagation of (M\_APP\_STATUS\_MASK left shifted with (multiplication of ( modulus of u8\_buffer and M\_FOUR) and M\_APP\_STATUS\_BITS))).
* Set au8\_Status with index u8\_stat\_mess and division of u8\_stat\_buffer and M\_FOUR of sXadc of App\_s\_comm to au8\_Status with index u8\_stat\_mess and division of u8\_stat\_buffer and M\_FOUR of sXadc of App\_s\_comm bitwise OR with (u8\_status

left shifted with (multiplication of (modulas of u8\_buffer and M\_FOUR) and M\_APP\_STATUS\_BITS))).

* Increment the u8\_buffer.
* Increment the u8\_stat\_buffer.
* Increment u8\_stat\_mess and set u8\_stat\_buffer to 0 when divition of u8\_stat\_buffer and M\_FOUR is greater than or equal to 8.

##### 10.3.7.7.5 dauanaapp-ScaleReadings-LLR-005

Requirement ID: H398-LLD-ANA-FNC-2328

The function shall perform the following:

* DAC Output Parameters of index of size DAC\_CHAN\_1 is getting updated with ai16\_Analog of index size of 18 of sXadc of App\_s\_comm.
* DAC Output Parameters of index of size DAC\_CHAN\_2 is getting updated with ai16\_Analog of index size of 19 of sXadc of App\_s\_comm.
* DAC Output Parameters of index of size DAC\_CHAN\_3 is equal to the sum of DAC Output Parameters of index of size DAC\_CHAN\_1 and DAC Output Parameters of index of size DAC\_CHAN\_2.
* totr corrective resistor value of index size of 0 is getting updated with division of ai16\_Analog of index size of M\_THIRTEEN of sXadc of App\_s\_comm and M\_TEN.
* tot raw of index size of 0 is getting updated with division of ai16\_Analog of index size of M\_THREE of sXadc of App\_s\_comm and M\_TEN.
* tot corrective value of index size of M\_ZERO is getting updated with tot corrective a multiplied with (tot raw of index size of M\_ZERO multiply with tot raw of index size of M\_ZERO) added with (tot corrective b multiply with totr corrective resistor of index size of M\_ZERO) added with tot corrective c.
* tot value of index size of M\_ZERO is getting updated with multiplication of tot raw of index size of 0 and totr corrective resistor of index size of M\_ZERO.
* DAC Output Parameters of index size of DAC\_CHAN\_4 is getting updated with multiplication of tot value of index size of M\_ZERO and M\_TEN.
* totr corrective resistor value of index size of 1 is getting updated with division of ai16\_Analog of index size of 14 of sXadc of App\_s\_comm and 10.
* tot raw of index size of 1 is getting updated with division of ai16\_Analog of index size of 4 of sXadc of App\_s\_comm and 10.
* tot corrective value of index size of 1 is getting updated with tot corrective a multiplied with (tot raw of index size of 1 multiply with tot raw of index size of 1) added with (tot corrective b multiply with totr corrective resistor of index size of 1) added with tot corrective c.
* tot value of index size of 1 is getting updated with multiplication of tot raw of index size of 1 and totr corrective resistor of index size of 1.
* DAC Output Parameters of index size of DAC\_CHAN\_5 is getting updated with multiplication of tot value of index size of 1 and 10.
* DAC Output Parameters of index of size DAC\_CHAN\_6 is getting updated with ai16\_Analog of index size of 17 of sXadc of App\_s\_comm.
* DAC Output Parameters of index of size DAC\_CHAN\_7 is getting updated with ai16\_Analog of index size of 35 of sXadc of App\_s\_comm.

### 10.3.8 AppMessageNOC

Low Level Design Details about CSU AppMessageNOC will follow in the sub sections.

#### 10.3.8.1 Brief Description

The function AppMessageNOC check for the Broad Cast msg in Normal Operation Channel queue and Transmit the message when any present.

#### 10.3.8.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.3.8.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 10.3.8.4 Parameter list (Input/Output)

Inputs : None

Outputs : None

#### 10.3.8.5 Return Value

None

#### 10.3.8.6 Other CSUs called by this CSU

SendReadings

SendCrcPn

#### 10.3.8.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to AppMessageNOC.

##### 10.3.8.7.1 dauanaapp-AppMessageNOC-LLR-001

Requirement ID: H398-LLD-ANA-FNC-290

The function shall call SendReadings when doc of bid of sid of Ptr\_mesg is equal to ACQUIRE.

##### 10.3.8.7.2 dauanaapp-AppMessageNOC-LLR-002

Requirement ID: H398-LLD-ANA-FNC-291

The function shall call SendCrcPn when doc of bid of sid of Ptr\_mesg is equal to ACQUIRE\_CRC\_PN.

##### 10.3.8.7.3 dauanaapp-AppMessageNOC-LLR-004

Requirement ID: H398-LLD-ANA-FNC-293

The function shall set error flag to M\_ONE when doc of bid of sid of Ptr\_mesg is not equal to ACQUIRE, ACQUIRE\_CRC\_PN.

##### 10.3.8.7.4 dauanaapp-AppMessageNOC-LLR-005

Requirement ID: H398-LLD-ANA-FNC-294

The function shall return error flag.

### 10.3.9 AppMessageNSC

Low Level Design Details about CSU AppMessageNSC will follow in the sub sections.

#### 10.3.9.1 Brief Description

The function AppMessageNSC checks for the peer to peer message in Node Service Channel to transmit the message when any present.

#### 10.3.9.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.3.9.3 List of global variables accessed and modified

Accessed : None

Modified : Excitation\_off

#### 10.3.9.4 Parameter list (Input/Output)

Inputs : T\_A825\_MSG \*Ptr\_mesg

Outputs : None

#### 10.3.9.5 Return Value

None

#### 10.3.9.6 Other CSUs called by this CSU

GpioSetBits

GpioResetBits

A825Transmit

#### 10.3.9.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to AppMessageNSC.

##### 10.3.9.7.1 dauanaapp-AppMessageNSC-LLR-001

Requirement ID: H398-LLD-ANA-FNC-303

The function shall perform as follows when A825 node service code is equal to NSC\_SET\_RCI.

1. Set rci of peer to peer ID to (u8\_payload of index M\_TWO of A825 message bitwise AND with M\_THREE).
2. Set rci of broadcast ID to rci of peer to peer ID
3. Set sid of pid of sid of temp message to ANALOG\_SID.
4. Call the function A825Transmit with parameter temp message.

##### 10.3.9.7.2 dauanaapp-AppMessageNSC-LLR-002

Requirement ID: H398-LLD-ANA-FNC-304

The function shall call M\_HW\_EXCITATION\_ON and set excitation\_Off to FALSE when CMD\_ON is equal to u8\_payload of index M\_A825\_CMD\_SIZE of temp message and A825 node service code is equal to NSC\_EXCITE\_SWITCH.

##### 10.3.9.7.3 dauanaapp-AppMessageNSC-LLR-003

Requirement ID: H398-LLD-ANA-FNC-305

The function shall call M\_HW\_EXCITATION\_OFF and set excitation\_Off to TRUE when CMD\_ON is not equal to u8\_payload of index M\_A825\_CMD\_SIZE of temp message and A825 node service code is equal to NSC\_EXCITE\_SWITCH.

##### 10.3.9.7.4 dauanaapp-AppMessageNSC-LLR-004

Requirement ID: H398-LLD-ANA-FNC-306

The function shall do all following when A825 node service code is equal to NSC\_EXCITE\_SWITCH.

1. Set sid of pid of sid of temp message to ANALOG\_SID.
2. Call the function A825Transmit with parameter temp message.

##### 10.3.9.7.5 dauanaapp-AppMessageNSC-LLR-006

Requirement ID: H398-LLD-ANA-FNC-308

The function shall perform as follows when A825 node service code is equal to NSC\_RESET:

* Set aircr of M\_SCB to ((M\_HEX\_5FA left shift by M\_SCB\_AIRCR\_VECTKEY\_POS) bitwise OR with (aircr of M\_SCB bitwise AND with M\_SCB\_AIRCR\_PRIGROUP\_MSK) bitwise OR with M\_SCB\_AIRCR\_SYSRESETREQ\_MSK) when sid of pid of sid of temp message to ANALOG\_SID else do nothing.

##### 10.3.9.7.6 dauanaapp-AppMessageNSC-LLR-007

Requirement ID: H398-LLD-ANA-FNC-309

The function shall set error to M\_ONE when A825 node service code is not equal to NSC\_SET\_RCI,NSC\_EXCITE\_SWITCH,NSC\_RESET.

##### 10.3.9.7.7 dauanaapp-AppMessageNSC-LLR-008

Requirement ID: H398-LLD-ANA-FNC-310

The function shall do the following when M\_ZERO is equal to error flag:

1. Set the smt of pid of sid of temp message to M\_ZERO.
2. Set sid of pid of sid of temp message to ANALOG\_SID.
3. call A825Transmit with parameter reference of temp message.

##### 10.3.9.7.8 dauanaapp-AppMessageNSC-LLR-009

Requirement ID: H398-LLD-ANA-FNC-311

The function shall return error flag.

### 10.3.10 AppTask

Low Level Design Details about CSU AppTask will follow in the sub sections.

#### 10.3.10.1 Brief Description

The function AppTask initiates the Tx of all the message present in the Tx queue.

#### 10.3.10.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.3.10.3 List of global variables accessed and modified

None

#### 10.3.10.4 Parameter list (Input/Output)

Inputs: None

Outputs: void \*pdata – pointer to null data (not used)

#### 10.3.10.5 Return Value

Accessed : None

Modified : None

#### 10.3.10.6 Other CSUs called by this CSU

OsSemPend

ScaleReadings

#### 10.3.10.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to AppTask.

##### 10.3.10.7.1 dauanaapp-AppTask-LLR-001

Requirement ID: H398-LLD-ANA-FNC-320

The function shall set pData to M\_NULL and enter to an infinite loop for continuous check for the message to transmission as per the following:

- Pend on semaphore by calling OsSemPend with the following parameters Allocated semaphore for the task sem app task, pend timeout value as M\_ZERO, Address of local error code.

- call ScaleReadings to Scale reading of all the sensor.

### 10.3.11 APPInit

Low Level Design Details about CSU APPInit will follow in the sub sections.

#### 10.3.11.1 Brief Description

The function APPInit creates the application task Tx/Rx of all A825 msg.

#### 10.3.11.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.3.11.3 List of global variables accessed and modified

Accessed : App\_task\_stk

Modified : None

#### 10.3.11.4 Parameter list (Input/Output)

Inputs : None

Outputs : None

#### 10.3.11.5 Return Value

None

#### 10.3.11.6 Other CSUs called by this CSU

OSTaskCreate

ErrorHandler

TbaseTaskSignaling

OsSemCreate

AppTask

#### 10.3.11.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to APPInit.

##### 10.3.11.7.1 dauanaapp-APPInit-LLR-001

Requirement ID: H398-LLD-ANA-FNC-329

The function shall

1. Call the function 'OsSemCreate ' with parameter (M\_ZERO) to initialize the Semaphore App task.
2. Call the function 'ErrorHandler' when the function 'OsSemCreate' returns M\_NULL otherwise do nothing.

##### 10.3.11.7.2 dauanaapp-APPInit-LLR-002

Requirement ID: H398-LLD-ANA-FNC-330

The function shall Call the function 'TbaseTaskSignaling' with parameter (APP\_TASK\_TICKS, Semaphore App task).

##### 10.3.11.7.3 dauanaapp-APPInit-LLR-003

Requirement ID: H398-LLD-ANA-FNC-331

The function shall

1. Call the function ‘OSTaskCreate’ with parameter (function AppTask, M\_NULL, reference to AppTaskStk with M\_APP\_TASK\_STK\_SIZE, M\_APP\_TASK\_PRIO) to Create the application task in system.
2. call the function 'ErrorHandler' when the function 'OsTaskCreate' returns other than M\_OS\_NO\_ERR, otherwise do nothing.

### 10.3.12 ScaleTach

Low Level Design Details about CSU ScaleTach will follow in the sub sections.

#### 10.3.12.1 Brief Description

The function calculate the Tach channel reading.

#### 10.3.12.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.3.12.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 10.3.12.4 Parameter list (Input/Output)

Inputs: T\_UINT8 u8Chan

Outputs: T\_UINT8 pu8\_status

#### 10.3.12.5 Return Value

calculated Tach channel reading u16\_reading

#### 10.3.12.6 Other CSUs called by this CSU

None

#### 10.3.12.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to ScaleTach

##### 10.3.12.7.1 dauanaapp-ScaleTach-LLR-001

Requirement ID: H398-LLD-ANA-FNC-2337

The function shall set pu8\_status to APP\_CHAN\_OK and Return M\_ZERO when u8ChanState of Sensor\_as\_tach with index u8Chan of Ptr\_sensor\_aas\_analog is not equal to ENABLE other-wise do nothing.

##### 10.3.12.7.2 dauanaapp-ScaleTach-LLR-002

Requirement ID: H398-LLD-ANA-FNC-2338

The function shall set u16 reading to return value of function TACHRead with parameter u8Chan plus M\_ROUNDOFF\_VAL\_TACHCHAN

##### 10.3.12.7.3 dauanaapp-ScaleTach-LLR-003

Requirement ID: H398-LLD-ANA-FNC-2339

The function shall set pu8\_status to APP\_RANGE\_ERROR when u16 reading is less than u16MinRange of Sensor\_as\_tach with index u8Chan of Ptr\_sensor\_aas\_analog

##### 10.3.12.7.4 dauanaapp-ScaleTach-LLR-004

Requirement ID: H398-LLD-ANA-FNC-2340

The function shall set u16 reading to u16DefaultVal of Sensor\_as\_tach with index u8Chan of Ptr\_sensor\_aas\_analog when

* u8DefaultState of Sensor\_as\_tach with index u8Chan of Ptr\_sensor\_aas\_analog is equal to ENABLE
* u16 reading is less than u16MinRange of Sensor\_as\_tach with index u8Chan of Ptr\_sensor\_aas\_analog

##### 10.3.12.7.5 dauanaapp-ScaleTach-LLR-005

Requirement ID: H398-LLD-ANA-FNC-2341

The function shall set u16 reading to u16MinRange of Sensor\_as\_tach with index u8Chan of Ptr\_sensor\_aas\_analog when

* u8DefaultState of Sensor\_as\_tach with index u8Chan of Ptr\_sensor\_aas\_analog is not equal to ENABLE
* u16 reading is less than u16MinRange of Sensor\_as\_tach with index u8Chan of Ptr\_sensor\_aas\_analog

##### 10.3.12.7.6 dauanaapp-ScaleTach-LLR-006

Requirement ID: H398-LLD-ANA-FNC-2342

The function shall set pu8\_status to APP\_RANGE\_ERROR when u16 reading is greater than u16MaxRange of Sensor\_as\_tach with index u8Chan of Ptr\_sensor\_aas\_analog and u16 reading is greater than u16MinRange of Sensor\_as\_tach with index u8Chan of Ptr\_sensor\_aas\_analog.

##### 10.3.12.7.7 dauanaapp-ScaleTach-LLR-007

Requirement ID: H398-LLD-ANA-FNC-2343

The function shall set u16 reading to u16DefaultVal of Sensor\_as\_tach with index u8Chan of Ptr\_sensor\_aas\_analog when the below conditions are satisfied:

* u16 reading is greater than u16MaxRange of Sensor\_as\_tach with index u8Chan of Ptr\_sensor\_aas\_analog.
* u8DefaultState of Sensor\_as\_tach with index \_u8Chan of Ptr\_sensor\_aas\_analog equal to ENABLE.
* u16 reading is greater than u16MinRange of Sensor\_as\_tach with index u8Chan of Ptr\_sensor\_aas\_analog.

##### 10.3.12.7.8 dauanaapp-ScaleTach-LLR-008

Requirement ID: H398-LLD-ANA-FNC-2344

The function shall set u16 reading to u16MaxRange of Sensor\_as\_tach with index u8Chan of Ptr\_sensor\_aas\_analog when the below conditions are satisfied:

* u16 reading is greater than u16MaxRange of Sensor\_as\_tach with index u8Chan of Ptr\_sensor\_aas\_analog.
* u8DefaultState of Sensor\_as\_tach with index \_u8Chan of Ptr\_sensor\_aas\_analog not equal to ENABLE.
* u16 reading is greater than u16MinRange of Sensor\_as\_tach with index u8Chan of Ptr\_sensor\_aas\_analog.

##### 10.3.12.7.9 dauanaapp-ScaleTach-LLR-009

Requirement ID: H398-LLD-ANA-FNC-2345

The function shall set pu8\_status to APP\_CHAN\_OK When u16 reading is less than u16MaxRange of Sensor\_as\_tach with index u8Chan of Ptr\_sensor\_aas\_analog and u16 reading is greater than u16MinRange of Sensor\_as\_tach with index u8Chan of Ptr\_sensor\_aas\_analog.

##### 10.3.12.7.10 dauanaapp-ScaleTach-LLR-010

Requirement ID: H398-LLD-ANA-FNC-2346

The function shall Return the calculated Tach channel reading u16 reading.

### 10.3.13 Scalepwmc

Low Level Design Details about CSU Scalepwmc will follow in the sub sections.

#### 10.3.13.1 Brief Description

The calculated PWMC channel reading

#### 10.3.13.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.3.13.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 10.3.13.4 Parameter list (Input/Output)

Inputs: T\_UINT8 u8Chan

Outputs: T\_UINT8 pu8\_status

#### 10.3.13.5 Return Value

Return the calculated Tach channel reading u16\_reading

#### 10.3.13.6 Other CSUs called by this CSU

None

#### 10.3.13.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to ScalePwmc.

##### 10.3.13.7.1 dauanaapp-ScalePwmc-LLR-001

Requirement ID: H398-LLD-ANA-FNC-2355

The function shall set frequency index to M\_FORTY\_NINE when u8Chan is equal to M\_ZERO and set frequency index to M\_FIFTY\_ONE when u8Chan is equal to M\_ONE.

##### 10.3.13.7.2 dauanaapp-ScalePwmc-LLR-002

Requirement ID: H398-LLD-ANA-FNC-2356

The function shall do the following:

* Set average pwm by calling the function PWMCRead with index u8Chan.
* Set time period to division of M\_ONE and ai16\_Analog with index frequency index of sXadc of App\_s\_comm and M\_FP\_TEN when ai16\_Analog with index frequency index of sXadc of App\_s\_comm is not equal to M\_ZERO.

Otherwise,

Set u16 reading to (multiplication of (division of average pwm and (multiplication of time period and M\_THOUSAND)) and M\_THOUSAND.

##### 10.3.13.7.3 dauanaapp-ScalePwmc-LLR-003

Requirement ID: H398-LLD-ANA-FNC-2357

The function shall do the following when u16 reading is less than M\_EIGHTY\_FOUR

* Set u16 reading to M\_HUNDREAD.
* Set pu8\_status to APP\_RANGE\_ERROR.

##### 10.3.13.7.4 dauanaapp-ScalePwmc-LLR-004

Requirement ID: H398-LLD-ANA-FNC-2358

The function shall do the following when u16 reading is greater than M\_NINE\_ONE\_SIX.

* Set u16 reading to M\_NINE\_HUNDREAD.
* Set pu8\_status to APP\_RANGE\_ERROR.

##### 10.3.13.7.5 dauanaapp-ScalePwmc-LLR-005

Requirement ID: H398-LLD-ANA-FNC-2359

The function shall set pu8\_status to APP\_CHAN\_OK when u16 reading is less than M\_NINE\_ONE\_SIX and greater than M\_EIGHTY\_FOUR.

##### 10.3.13.7.6 dauanaapp-ScalePwmc-LLR-006

Requirement ID: H398-LLD-ANA-FNC-2360

The function shall return from functionality with u16 reading.

## 10.4 dauanacbit

This module contains the routine to Implement the Continuous Built in Test.

### 10.4.1 StackTest

Low Level Design Details about CSU StackTest will follow in the sub sections.

#### 10.4.1.1 Brief Description

The function performs Stack Test.

#### 10.4.1.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.4.1.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 10.4.1.4 Parameter list (Input/Output)

Inputs : T\_UINT32 \*buffer - Pointer to the top-of-stack address

T\_UINT32 u32\_test\_value - length of stack.

T\_UINT32 length [IN]- length of stack

Outputs : T\_UINT32 \*buffer - Pointer to the top-of-stack address

#### 10.4.1.5 Return Value

None

#### 10.4.1.6 Other CSUs called by this CSU

ErrorHandler

#### 10.4.1.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to StackTest.

##### 10.4.1.7.1 dauanacbit-StackTest-LLR-001

Requirement ID: H398-LLD-ANA-FNC-341

The function shall do the stack overflow condition check for the last 30 percent of the stack

by performing the following

1. Calculate the Stack test start index by calling M\_GET\_70PERC\_STACK with parameter length.
2. Decrement stack location by looping through M\_ZERO to stack size
3. Read eleven contiguous memory locations for the pattern u32\_test\_value (M\_ZERO to M\_TEN)

* when the pattern u32\_test\_value is not matching with the pattern present in the stack location call ErrorHandler function, otherwise do nothing.
* Decrement the stack location.

### 10.4.2 ErrorHandler

Low Level Design Details about CSU ErrorHandler will follow in the sub sections.

#### 10.4.2.1 Brief Description

This function handles the Continuous Built in Test errors.

#### 10.4.2.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.4.2.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 10.4.2.4 Parameter list (Input/Output)

Inputs : None

Outputs : None

#### 10.4.2.5 Return Value

None

#### 10.4.2.6 Other CSUs called by this CSU

WdogKickWatchDog

GpioSetBits

#### 10.4.2.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to ErrorHandler.

##### 10.4.2.7.1 dauanacbit-ErrorHandler-LLR-001

Requirement ID: H398-LLD-ANA-FNC-350

The function shall disable all interrupts.

##### 10.4.2.7.2 dauanacbit-ErrorHandler-LLR-002

Requirement ID: H398-LLD-ANA-FNC-351

The function shall call the function WdogKickWatchDog to reset the watchdog counter and call the function GpioSetBits (M\_HW\_LED\_HB\_ON) with parameter (M\_GPIOB, M\_GPIOB\_LED\_HB) in an infinite loop to make the heartbeat LED steady glow.

### 10.4.3 CbitTask

Low Level Design Details about CSU CbitTask will follow in the sub sections.

#### 10.4.3.1 Brief Description

The function initiates the Continuous Built in Test.

#### 10.4.3.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.4.3.3 List of global variables accessed and modified

Accessed : Os\_task\_idle\_stk, Init\_task\_stk, App\_task\_stk, A825\_task\_stack

Modified : None

#### 10.4.3.4 Parameter list (Input/Output)

Inputs : void \*pData - Not used

Outputs : None

#### 10.4.3.5 Return Value

None

#### 10.4.3.6 Other CSUs called by this CSU

OSSemPend

StackTest

#### 10.4.3.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to CbitTask.

##### 10.4.3.7.1 dauanacbit-CbitTask-LLR-001

Requirement ID: H398-LLD-ANA-FNC-360

The function shall do o the following

1. Set pData to M\_NULL
2. The function shall Loop infinetly and perform the following

* call 'OsSemPend' for waiting on a semaphore with parameters semaphore for cbit task, M\_ZERO and reference to error.
* Call StackTest with parameters reference to Cbit\_task\_stack with index of (M\_CBIT\_TASK\_STK\_SIZE), M\_CBIT\_TASK\_STK\_SIZE and 0xDEADDEAD.
* Call StackTest with parameters reference to Os\_task\_idle\_stk with index of (M\_OS\_IDLE\_TASK\_STK\_SIZE), M\_OS\_IDLE\_TASK\_STK\_SIZE and 0xDEADFEED.
* Call StackTest with with parameters reference to App\_task\_stk with index of (M\_APP\_TASK\_STK\_SIZE), M\_APP\_TASK\_STK\_SIZE and 0xDEADF00D.
* Call StackTest with with parameters reference to A825\_task\_stack with index of (M\_A8252\_TASK\_STK\_SIZE), M\_A8252\_TASK\_STK\_SIZE and 0xDEADBEEF.
* Call StackTest with with parameters reference to Init\_task\_stk with index of (M\_INIT\_TASK\_STK\_SIZE), M\_INIT\_TASK\_STK\_SIZE and 0xDEADD0D0.

### 10.4.4 CbitInit

Low Level Design Details about CSU CbitInit will follow in the sub sections.

#### 10.4.4.1 Brief Description

The function initializes stack and creates Task to implement CBIT.

#### 10.4.4.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.4.4.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 10.4.4.4 Parameter list (Input/Output)

Inputs : None

Outputs : None

#### 10.4.4.5 Return Value

None

#### 10.4.4.6 Other CSUs called by this CSU

TbaseTaskSignaling

OsSemCreate

OsTaskCreate

ErrorHandler

CbitTask

#### 10.4.4.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to CbitInit.

##### 10.4.4.7.1 dauanacbit-CbitInit-LLR-001

Requirement ID: H398-LLD-ANA-FNC-369

The function shall

1. call 'OsSemCreate' with parameter (M\_ZERO) to create a semaphore
2. call the function 'ErrorHandler' when the function 'OsSemCreate' return M\_NULL otherwise do nothing.

##### 10.4.4.7.2 dauanacbit-CbitInit-LLR-002

Requirement ID: H398-LLD-ANA-FNC-370

The function shall call 'TbaseTaskSignaling' with parameter (M\_CBIT\_TASK\_TICKS,Semaphore for CBIT task) to initialize the task signalling parameters.

##### 10.4.4.7.3 dauanacbit-CbitInit-LLR-003

Requirement ID: H398-LLD-ANA-FNC-371

The function 'OsTaskCreate' shall be called to create the Task for CBIT by passing following arguments and call the function 'ErrorHandler' when the function 'OsTaskCreate' returns other than M\_OS\_NO\_ERR otherwise do nothing.

- Pointer to the ContinuousBitTask's address.

- NULL pointer passed as argument to the ContinuousBitTask.

- ContinuousBitTask's top-of-stack i.e. Cbit\_task\_Stack[M\_CBIT\_TASK\_STK\_SIZE].

- Task priority as M\_CBIT\_TASK\_PRIO

## 10.5 dauanacputest

The dauanacputest CSC defines the implementation of cputest functions.

### 10.5.1 AluTest

Low Level Design Details about CSU AluTest will follow in the sub sections.

#### 10.5.1.1 Brief Description

The AluTest function performs the arithmetical and logical operations on registers during PBIT.

#### 10.5.1.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD(H398-003-012-ANA).

#### 10.5.1.3 List of global variables accessed and modified

Accessed: Cpu\_test\_res

Modified: None

#### 10.5.1.4 Parameter list (Input/Output)

Inputs: None

Outputs: None

#### 10.5.1.5 Return Value

None

#### 10.5.1.6 Other CSUs called by this CSU

None

#### 10.5.1.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to AluTest.

##### 10.5.1.7.1 dauanacputest-AluTest-LLR-001

Requirement ID: H398-LLD-ANA-FNC-381

The function shall perform the addition operation on registers (R1, R2) with operands M\_CONST\_SIX and M\_CONST\_FIVE and when the obtained result (R2) matches with the value M\_CONST\_ELEVEN then perform carry flag test otherwise exits from the subroutine.

##### 10.5.1.7.2 dauanacputest-AluTest-LLR-002

Requirement ID: H398-LLD-ANA-FNC-382

The function shall perform the addition operation on registers (R1, R2) with operands M\_CONST\_SIX and M\_CONST\_FIVE and store the status of the carry flag from APSR into the register (R4). When the obtained result (R4) matches with the value M\_CONST\_CARRY\_FLAG, then perform zero flag test otherwise exits from the subroutine.

##### 10.5.1.7.3 dauanacputest-AluTest-LLR-003

Requirement ID: H398-LLD-ANA-FNC-383

The function shall perform the addition operation on registers (R1, R2) with operands M\_CONST\_PATTERN and M\_CONST\_PATTERN and store the status of the zero flag from APSR into the register (R4). When the obtained result (R4) matches with the value M\_CONST\_OVERFLOW\_FLAG perform subtract test otherwise exit from the subroutine.

##### 10.5.1.7.4 dauanacputest-AluTest-LLR-004

Requirement ID: H398-LLD-ANA-FNC-384

The function shall perform the addition operation on registers (R1, R2) with operands M\_CONST\_SIX and M\_CONST\_FIVE, compare R2 with M\_CONST\_ELEVEN and store the status of the zero flag from APSR into the register (R4). When the obtained result (R4) matches with the value M\_CONST\_ZERO\_FLAG, then perform overflow flag test otherwise exits from the subroutine.

##### 10.5.1.7.5 dauanacputest-AluTest-LLR-005

Requirement ID: H398-LLD-ANA-FNC-385

The function shall perform the subtract operation on registers (R1, R2) with operands M\_CONST\_SIX and M\_CONST\_FIVE and when the obtained result(R1) matches with the value M\_CONST\_ONE then perform multiply test otherwise exits from the subroutine.

##### 10.5.1.7.6 dauanacputest-AluTest-LLR-006

Requirement ID: H398-LLD-ANA-FNC-386

The function shall perform the multiply operation on registers (R1, R2) with operands M\_CONST\_THREE and M\_CONST\_TWO and when the obtained result (R1) matches with the value M\_CONST\_SIX then perform division test otherwise exit from the subroutine.

##### 10.5.1.7.7 dauanacputest-AluTest-LLR-007

Requirement ID: H398-LLD-ANA-FNC-387

The function shall perform the division operation on registers (R1, R2) with operands M\_CONST\_FOUR and M\_CONST\_TWO and when the obtained result (R1) matches with the value M\_CONST\_TWO then perform logical and test otherwise exit from the subroutine.

##### 10.5.1.7.8 dauanacputest-AluTest-LLR-008

Requirement ID: H398-LLD-ANA-FNC-388

The function shall perform the logical AND operation on registers (R1, R2) with operands M\_CONST\_HEX\_ONE and M\_CONST\_SEVENTEEN and when the obtained result (R3) matches with the value M\_CONST\_ONE then perform logical or test otherwise exit from the subroutine.

##### 10.5.1.7.9 dauanacputest-AluTest-LLR-009

Requirement ID: H398-LLD-ANA-FNC-389

The function shall perform the logical OR operation on registers (R1, R2) with operands M\_CONST\_SIXTEEN and M\_CONST\_HEX\_ONE and when the obtained result (R3) matches with the value M\_CONST\_SEVENTEEN then perform exclusive or test otherwise exit from the subroutine.

##### 10.5.1.7.10 dauanacputest-AluTest-LLR-010

Requirement ID: H398-LLD-ANA-FNC-390

The function shall perform the logical XOR operation on registers (R1, R2) with operands M\_CONST\_SEVENTEEN and M\_CONST\_HEX\_ONE and when the obtained result (R3) matches with the value M\_CONST\_SIXTEEN then branch to Test result otherwise exit from the subroutine.

##### 10.5.1.7.11 dauanacputest-AluTest-LLR-011

Requirement ID: H398-LLD-ANA-FNC-391

The function shall initialize R4 with M\_CONST\_ONE and load the address of Cpu\_test\_res to the R2 register, set the global variable Cpu\_test\_res to TRUE by writing R4 to content of R2 address.

## 10.6 dauanacrt0

This module copies and resets the vector table for STM32F40x and sets the Stack to Main.

### 10.6.1 Crt0TransferData

Low Level Design Details about CSU Crt0TransferData will follow in the sub sections.

#### 10.6.1.1 Brief Description

The function Crt0TransferData Transfers data from ROM to RAM location to start Execution.

#### 10.6.1.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.6.1.3 List of global variables accessed and modified

Accessed: :

\_data\_rom

\_data\_start

\_data\_end

\_bss\_start

\_bss\_end

Modified: :

\_data\_start

\_bss\_start

#### 10.6.1.4 Parameter list (Input/Output)

Inputs: None

Outputs: None

#### 10.6.1.5 Return Value

None

#### 10.6.1.6 Other CSUs called by this CSU

MainFunc

#### 10.6.1.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to Crt0TransferData.

##### 10.6.1.7.1 dauanacrt0-Crt0TransferData-LLR-001

Requirement ID: H398-LLD-ANA-FNC-401

The function shall move data section from ROM to RAM by looping through \_data\_start to \_data\_end.

##### 10.6.1.7.2 dauanacrt0-Crt0TransferData-LLR-002

Requirement ID: H398-LLD-ANA-FNC-402

The function shall fill Zero to bss memory section by looping through \_bss\_start to \_bss\_end.

##### 10.6.1.7.3 dauanacrt0-Crt0TransferData-LLR-003

Requirement ID: H398-LLD-ANA-FNC-403

The function shall call MainFunc to branch to main function.

## 10.7 dauanahw

This module contains clock references, Gpio pin information routines.

### 10.7.1 InitIO

Low Level Design Details about CSU InitIO will follow in the sub sections.

#### 10.7.1.1 Brief Description

The function InitIO initializes Clock reference and Pin setting for Port A, B, C, D, E, H.

#### 10.7.1.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.7.1.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 10.7.1.4 Parameter list (Input/Output)

Inputs : None

Outputs : None

#### 10.7.1.5 Return Value

None

#### 10.7.1.6 Other CSUs called by this CSU

RccAhb3PeriphClockCmd

RccApb2PeriphClockCmd

RccAhb1PeriphClockCmd

GpioInit

GpioPinAFConfig

GpioSetBits

#### 10.7.1.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to InitIO.

##### 10.7.1.7.1 dauanahw-InitIO-LLR-001

Requirement ID: H398-LLD-ANA-FNC-413

The function shall Call the function RccAhb3PeriphClockCmd with parameter (M\_RCC\_AHB3PERIPH\_FSMC, ENABLE) to Enable FSMC clock.

##### 10.7.1.7.2 dauanahw-InitIO-LLR-002

Requirement ID: H398-LLD-ANA-FNC-414

The function shall Call the function RccApb2PeriphClockCmd with parameter (M\_RCC\_APB2PERIPH\_SYSCFG, ENABLE) to Enable system configuration clock.

##### 10.7.1.7.3 dauanahw-InitIO-LLR-003

Requirement ID: H398-LLD-ANA-FNC-415

The function shall perform the following

1. Call the function RccAhb1PeriphClockCmd with parameter (M\_RCC\_AHB1PERIPH\_GPIOA, ENABLE) to Enable M\_GPIOA clock.

##### 10.7.1.7.4 dauanahw-InitIO-LLR-004

Requirement ID: H398-LLD-ANA-FNC-416

The function shall Call the function RccAhb1PeriphClockCmd with parameter (M\_RCC\_AHB1PERIPH\_GPIOB, ENABLE) to Enable M\_GPIOB clock.

##### 10.7.1.7.5 dauanahw-InitIO-LLR-005

Requirement ID: H398-LLD-ANA-FNC-417

The function shall Set the members of Init structure to configure LED driving pin and clock for all ports as follows:

Calls the function GpioSetBits with parameters M\_GPIOB and M\_GPIOB\_DAC\_SEL.

* Calls the function GpioSetBits with parameters M\_GPIOB and M\_GPIOB\_DAC\_SEL.
* Calls the function GpioSetBits with parameters M\_GPIOB and M\_GPIOB\_DECODER1.
* Calls the function GpioSetBits with parameters M\_GPIOB and M\_GPIOB\_DECODER2.
* Calls the function GpioSetBits with parameters M\_GPIOB and M\_GPIOB\_DECODER3.
* Set the gpio\_pin of Init structure to (M\_GPIOB\_LED\_HB BITWISE OR M\_GPIOB\_DAC\_SEL BITWISE OR M\_GPIOB\_DECODER1 BITWISE OR M\_GPIOB\_DECODER2 BITWISE OR M\_GPIOB\_DECODER3).
* Set the gpio\_speed of Init structure to GPIO\_SPEED\_50MHZ (Set the value for All the ports)
* Set the gpio\_mode of Init structure to GPIO\_MODE\_OUT
* Set the gpio\_otype of Init structure to GPIO\_OTYPE\_PP
* Set the gpio\_pupd of Init structure to GPIO\_PUPD\_NOPULL

##### 10.7.1.7.6 dauanahw-InitIO-LLR-006

Requirement ID: H398-LLD-ANA-FNC-418

The function shall Call the function GpioInit with parameter (M\_GPIOB, Reference to the init structure) to Configure LED driving pin as per the setting of the init structure.

##### 10.7.1.7.7 dauanahw-InitIO-LLR-007

Requirement ID: H398-LLD-ANA-FNC-419

The function shall Call the function RccAhb1PeriphClockCmd with parameter (M\_RCC\_AHB1PERIPH\_GPIOC, ENABLE) to Enable M\_GPIOC clock.

##### 10.7.1.7.8 dauanahw-InitIO-LLR-008

Requirement ID: H398-LLD-ANA-FNC-420

The function shall configure pin 0 and 1 of GPIO port C by calling 'GpioInit' with parameters M\_GPIOC and gpio init structure with the members initialized as follows

a) set GPIO port output speed register(gpio\_speed) with GPIO\_SPEED\_50MHZ,

b) set gpio\_pin to M\_GPIOC\_ADC\_IN1 (gpio pin 0),

c) set GPIO port mode register(gpio\_mode) to GPIO\_MODE\_AN and

d) set GPIO port pull-up/pull-down register(gpio\_pupd) to GPIO\_PUPD\_NOPULL.

##### 10.7.1.7.9 dauanahw-InitIO-LLR-009

Requirement ID: H398-LLD-ANA-FNC-421

The function shall Set the members of Init structure to configure output control pins for port c as follows:

a) Set the gpio\_pin of Init structure to M\_GPIOC\_I\_CNTRL bitwise OR M\_GPIOC\_XADC\_CONVSTbitwise ORM\_GPIOC\_XADC\_MUX\_0bitwise ORM\_GPIOC\_XADC\_MUX\_1 bitwise OR M\_GPIOC\_XADC\_MUX\_2 bitwise OR M\_GPIOC\_XADC\_MUX\_3

b) Set the gpio\_mode of Init structure to GPIO\_MODE\_OUT

c) Set the gpio\_otype of Init structure to GPIO\_OTYPE\_PP

d) Set the gpio\_pupd of Init structure to GPIO\_PUPD\_NOPULL

##### 10.7.1.7.10 dauanahw-InitIO-LLR-010

Requirement ID: H398-LLD-ANA-FNC-422

The function shall Call the function GpioInit with parameter (M\_GPIOC, Reference to the init structure) to Configure output control pins for port c as per the setting of the init structure.

##### 10.7.1.7.11 dauanahw-InitIO-LLR-013

Requirement ID: H398-LLD-ANA-FNC-425

The function shall Call the function RccAhb1PeriphClockCmd with parameter (M\_RCC\_AHB1PERIPH\_GPIOD, ENABLE) to Enable M\_GPIOD clock.

##### 10.7.1.7.12 dauanahw-InitIO-LLR-014

Requirement ID: H398-LLD-ANA-FNC-426

The function shall do the following to Configure FSMC pins of port D as follows (Config for the Data bus (D0, D1, D2, D3, D13, D14, D15), chip selection, Output Enable, Write Enable)

- Call the function GpioPinAFConfig with parameter

(M\_GPIOD, M\_GPIO\_PINSOURCE0, M\_GPIO\_AF\_FSMC)

- Call the function GpioPinAFConfig with parameter

(M\_GPIOD, M\_GPIO\_PINSOURCE1, M\_GPIO\_AF\_FSMC)

- Call the function GpioPinAFConfig with parameter

(M\_GPIOD, M\_GPIO\_PINSOURCE4, M\_GPIO\_AF\_FSMC)

- Call the function GpioPinAFConfig with parameter

(M\_GPIOD, M\_GPIO\_PINSOURCE5, M\_GPIO\_AF\_FSMC)

- Call the function GpioPinAFConfig with parameter

(M\_GPIOD, M\_GPIO\_PINSOURCE7, M\_GPIO\_AF\_FSMC)

- Call the function GpioPinAFConfig with parameter

(M\_GPIOD, M\_GPIO\_PINSOURCE8, M\_GPIO\_AF\_FSMC)

- Call the function GpioPinAFConfig with parameter

(M\_GPIOD, M\_GPIO\_PINSOURCE9, M\_GPIO\_AF\_FSMC)

- Call the function GpioPinAFConfig with parameter

(M\_GPIOD, M\_GPIO\_PINSOURCE10, M\_GPIO\_AF\_FSMC)

- Call the function GpioPinAFConfig with parameter

(M\_GPIOD, M\_GPIO\_PINSOURCE14, M\_GPIO\_AF\_FSMC)

- Call the function GpioPinAFConfig with parameter

(M\_GPIOD, M\_GPIO\_PINSOURCE15, M\_GPIO\_AF\_FSMC)

##### 10.7.1.7.13 dauanahw-InitIO-LLR-015

Requirement ID: H398-LLD-ANA-FNC-427

The function shall Set the members of Init structure to configure FSMC pins of port D as alternate function output pins as follows:

- Set the gpio\_pin of Init structure to M\_GPIOD\_D2 bitwise OR M\_GPIOD\_D3 bitwise OR M\_GPIOD\_OE bitwise OR M\_GPIOD\_WE bitwise OR M\_GPIOD\_NE1 bitwise OR M\_GPIOD\_D13 bitwise OR M\_GPIOD\_D14 bitwise OR M\_GPIOD\_D15 bitwise OR M\_GPIOD\_D0 bitwise OR M\_GPIOD\_D1

- Set the gpio\_mode of Init structure to GPIO\_MODE\_AF

- Set the gpio\_otype of Init structure to GPIO\_OTYPE\_PP

- Set the gpio\_pupd of Init structure to GPIO\_PUPD\_NOPULL

##### 10.7.1.7.14 dauanahw-InitIO-LLR-016

Requirement ID: H398-LLD-ANA-FNC-428

The function shall Call the function GpioInit with parameter (M\_GPIOD, Reference to the init structure) to Configure FSMC pins of port D as per the setting of the init structure.

##### 10.7.1.7.15 dauanahw-InitIO-LLR-017

Requirement ID: H398-LLD-ANA-FNC-429

The function shall Call the function RccAhb1PeriphClockCmd with parameter (M\_RCC\_AHB1PERIPH\_GPIOE, ENABLE) to Enable M\_GPIOE clock.

##### 10.7.1.7.16 dauanahw-InitIO-LLR-018

Requirement ID: H398-LLD-ANA-FNC-430

The function shall do the following to Configure FSMC pins of port E as follows:

(Config for the Data bus (D4, D5, D6, D7, D8, D9, D10, D11, D12))

- Call the function GpioPinAFConfig with parameter

(M\_GPIOE, M\_GPIO\_PINSOURCE7, M\_GPIO\_AF\_FSMC)

- Call the function GpioPinAFConfig with parameter

(M\_GPIOE, M\_GPIO\_PINSOURCE8, M\_GPIO\_AF\_FSMC)

- Call the function GpioPinAFConfig with parameter

(M\_GPIOE, M\_GPIO\_PINSOURCE9, M\_GPIO\_AF\_FSMC)

- Call the function GpioPinAFConfig with parameter

(M\_GPIOE, M\_GPIO\_PINSOURCE10, M\_GPIO\_AF\_FSMC)

- Call the function GpioPinAFConfig with parameter

(M\_GPIOE, M\_GPIO\_PINSOURCE11, M\_GPIO\_AF\_FSMC)

- Call the function GpioPinAFConfig with parameter

(M\_GPIOE, M\_GPIO\_PINSOURCE12, M\_GPIO\_AF\_FSMC)

- Call the function GpioPinAFConfig with parameter

(M\_GPIOE, M\_GPIO\_PINSOURCE13, M\_GPIO\_AF\_FSMC)

- Call the function GpioPinAFConfig with parameter

(M\_GPIOE, M\_GPIO\_PINSOURCE14, M\_GPIO\_AF\_FSMC)

- Call the function GpioPinAFConfig with parameter

(M\_GPIOE, M\_GPIO\_PINSOURCE15, M\_GPIO\_AF\_FSMC)

##### 10.7.1.7.17 dauanahw-InitIO-LLR-019

Requirement ID: H398-LLD-ANA-FNC-431

The function shall Set the members of Init structure to configure FSMC pins of port E as alternate function output pins as follows:

- Set the gpio\_pin of Init structure to M\_GPIOE\_D4 bitwise OR M\_GPIOE\_D5 bitwise OR M\_GPIOE\_D6 bitwise OR M\_GPIOE\_D7 bitwise OR M\_GPIOE\_D8 bitwise OR M\_GPIOE\_D9 bitwise OR M\_GPIOE\_D10 bitwise OR M\_GPIOE\_D11 bitwise OR M\_GPIOE\_D12

- Set the gpio\_mode of Init structure to GPIO\_MODE\_AF

- Set the gpio\_otype of Init structure to GPIO\_OTYPE\_PP

- Set the gpio\_pupd of Init structure to GPIO\_PUPD\_NOPULL

##### 10.7.1.7.18 dauanahw-InitIO-LLR-020

Requirement ID: H398-LLD-ANA-FNC-432

The function shall Call the function GpioInit with parameter (M\_GPIOE, Reference to the init structure) to Configure FSMC pins of port E as per the setting of the init structure.

### 10.7.2 HwInit

Low Level Design Details about CSU HwInit will follow in the sub sections.

#### 10.7.2.1 Brief Description

The function HwInit sets the clock configuration, Enable Flash, enable interrupt vector table in RAM to perform all H/W related operation.

#### 10.7.2.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.7.2.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 10.7.2.4 Parameter list (Input/Output)

Inputs : None

Outputs : None

#### 10.7.2.5 Return Value

None

#### 10.7.2.6 Other CSUs called by this CSU

RccDeInit

RccHseConfig

RccApb1PeriphClockCmd

PwrMainRegulatorModeConfig

RccHclkConfig

RccPclk1Config

IntrInit

RccPclk2Config

FlashSetLatency

FlashInstructionCacheCmd

InitIO

FlashDataCacheCmd

FlashPrefetchBufferCmd

RccPllConfig

RccGetFlagStatus

RccSysClkConfig

RccGetSysClkSource

NvicSetVectorTable

NvicPriorityGroupConfig

NvicSetPriority

RccPllCmd

IntrInstall

RccWaitForHseStartUp

RccAhb1PeriphClockCmd

TbaseIntrHandler

PendSvHandler

#### 10.7.2.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to HwInit.

##### 10.7.2.7.1 dauanahw-HwInit-LLR-001

Requirement ID: H398-LLD-ANA-FNC-444

The function shall reset the RCC clock configuration to the default reset state by calling 'RccDeInit'.

##### 10.7.2.7.2 dauanahw-HwInit-LLR-002

Requirement ID: H398-LLD-ANA-FNC-445

The function shall enable the External High Speed oscillator (HSE) by calling 'RccHseConfig' with parameter M\_RCC\_HSE\_ON.

##### 10.7.2.7.3 dauanahw-HwInit-LLR-003

Requirement ID: H398-LLD-ANA-FNC-446

The function shall Call the function 'RccWaitForHseStartUp' for HSE start-up.

##### 10.7.2.7.4 dauanahw-HwInit-LLR-004

Requirement ID: H398-LLD-ANA-FNC-447

The function shall enable the Low Speed APB(APB1) peripheral power interface clock by calling 'RccApb1PeriphClockCmd' with parameters M\_RCC\_APB1PERIPH\_PWR, ENABLE.

##### 10.7.2.7.5 dauanahw-HwInit-LLR-005

Requirement ID: H398-LLD-ANA-FNC-448

The function shall set Regulator voltage scaling output selection bit to scale 1 mode in the power control register by calling 'PwrMainRegulatorModeConfig' with parameter M\_PWR\_REGULATOR\_VOLTAGE\_SCALE1.

##### 10.7.2.7.6 dauanahw-HwInit-LLR-006

Requirement ID: H398-LLD-ANA-FNC-449

The function shall configure the AHB clock (HCLK) to system clock(SYSCLK) by calling 'RccHclkConfig' with parameter M\_RCC\_SYSCLK\_DIV1.

##### 10.7.2.7.7 dauanahw-HwInit-LLR-007

Requirement ID: H398-LLD-ANA-FNC-450

The function shall configure the Low Speed APB clock (PCLK1) to HCLK/4 by calling 'RccPclk1Config' with parameter M\_RCC\_HCLK\_DIV4.

##### 10.7.2.7.8 dauanahw-HwInit-LLR-008

Requirement ID: H398-LLD-ANA-FNC-451

The function shall configure the High Speed APB clock (PCLK2) to HCLK/2 by calling 'RccPclk2Config' with parameter M\_RCC\_HCLK\_DIV2.

##### 10.7.2.7.9 dauanahw-HwInit-LLR-009

Requirement ID: H398-LLD-ANA-FNC-452

The function shall set the code latency value by calling 'FlashSetLatency' with parameter M\_FLASH\_LATENCY\_5.

##### 10.7.2.7.10 dauanahw-HwInit-LLR-010

Requirement ID: H398-LLD-ANA-FNC-453

The function shall enable the instruction Cache feature by calling 'FlashInstructionCacheCmd' with parameter ENABLE.

##### 10.7.2.7.11 dauanahw-HwInit-LLR-011

Requirement ID: H398-LLD-ANA-FNC-454

The function shall enable the data Cache feature by calling 'FlashDataCacheCmd' with parameter ENABLE.

##### 10.7.2.7.12 dauanahw-HwInit-LLR-012

Requirement ID: H398-LLD-ANA-FNC-455

The function shall enable pre-fetch buffer by calling 'FlashPrefetchBufferCmd' with parameter ENABLE.

##### 10.7.2.7.13 dauanahw-HwInit-LLR-013

Requirement ID: H398-LLD-ANA-FNC-456

The function shall configure the main PLL clock source, multiplication and division factors by calling 'RccPllConfig' with parameters M\_RCC\_PLLSOURCE\_HSE, M\_HW\_PLL\_M, M\_HW\_PLL\_N, M\_HW\_PLL\_P and M\_HW\_PLL\_Q.

##### 10.7.2.7.14 dauanahw-HwInit-LLR-014

Requirement ID: H398-LLD-ANA-FNC-457

The function shall enable main PLL by calling 'RccPllCmd' with parameter ENABLE.

##### 10.7.2.7.15 dauanahw-HwInit-LLR-015

Requirement ID: H398-LLD-ANA-FNC-458

The function shall wait till PLL is ready i.e., until 'RccGetFlagStatus' with parameter M\_RCC\_FLAG\_PLLRDY returns other than RESET

##### 10.7.2.7.16 dauanahw-HwInit-LLR-016

Requirement ID: H398-LLD-ANA-FNC-459

The function shall set the PLL as system clock by calling 'RccSysClkConfig' with parameter M\_RCC\_SYSCLKSOURCE\_PLLCLK.

##### 10.7.2.7.17 dauanahw-HwInit-LLR-017

Requirement ID: H398-LLD-ANA-FNC-460

The function shall wait till PLL is used as system clock source i.e., until 'RccGetSysClkSource' returns M\_PLL\_USD\_AS\_SYSCLK.

##### 10.7.2.7.18 dauanahw-HwInit-LLR-018

Requirement ID: H398-LLD-ANA-FNC-461

The function shall initialize the interrupt vector table in RAM by calling 'IntrInit’.

##### 10.7.2.7.19 dauanahw-HwInit-LLR-019

Requirement ID: H398-LLD-ANA-FNC-462

The function shall set the Vector Table base address at RAM by calling 'NvicSetVectorTable' with parameters M\_NVIC\_VECTTAB\_RAM and M\_HEX\_ZERO.

##### 10.7.2.7.20 dauanahw-HwInit-LLR-020

Requirement ID: H398-LLD-ANA-FNC-463

The function shall configure the group priority and sub-priority by calling 'NvicPriorityGroupConfig' with parameter M\_NVIC\_PRIORITYGROUP\_4.

##### 10.7.2.7.21 dauanahw-HwInit-LLR-021

Requirement ID: H398-LLD-ANA-FNC-464

The function shall load the reference to the TbaseIntrHandler into the vector table for system tick interrupt into RAM by calling 'IntrInstall' with parameters INTR\_SYS\_TICK and reference to TbaseIntrHandler.

##### 10.7.2.7.22 dauanahw-HwInit-LLR-022

Requirement ID: H398-LLD-ANA-FNC-465

The function shall set priority of the system tick interrupt by calling 'NvicSetPriority' with parameters SYSTICK\_IRQN and M\_SET\_PRIORITY\_15.

##### 10.7.2.7.23 dauanahw-HwInit-LLR-023

Requirement ID: H398-LLD-ANA-FNC-466

The function shall Call the function 'IntrInstall' with parameter (INTR\_PEND\_SV, PendSVHandler) to Setting up the PendSV interrupt into vector table in RAM.

##### 10.7.2.7.24 dauanahw-HwInit-LLR-024

Requirement ID: H398-LLD-ANA-FNC-467

The function shall set priority of the PendSV interrupt by calling 'NvicSetPriority' with parameters PENDSV\_IRQN and M\_SET\_PRIORITY\_14.

##### 10.7.2.7.25 dauanahw-HwInit-LLR-025

Requirement ID: H398-LLD-ANA-FNC-468

The function shall enable the CRC peripheral clock by calling 'RccAhb1PeriphClockCmd' with parameters M\_RCC\_AHB1PERIPH\_CRC and ENABLE.

##### 10.7.2.7.26 dauanahw-HwInit-LLR-026

Requirement ID: H398-LLD-ANA-FNC-469

The function shall initialize the FPU context save not in lazy mode (i.e. M\_FPU\_CNTXT\_NOT\_LAZYMODE) in fpccr (Floating-point context control register) of M\_FPU.

##### 10.7.2.7.27 dauanahw-HwInit-LLR-027

Requirement ID: H398-LLD-ANA-FNC-470

The function shall enable full access privileges for coprocessors (i.e. M\_ENABLE\_FP\_COPROC) in cpacr (Coprocessor access control register) of M\_SCB.

##### 10.7.2.7.28 dauanahw-HwInit-LLR-029

Requirement ID: H398-LLD-ANA-FNC-1886

Set shcsr of M\_SCB to M\_SCB\_SHCSR\_USGFAULTENA\_MSK bitwise OR with M\_SCB\_SHCSR\_BUSFAULTENA\_MSK bitwise OR with M\_SCB\_SHCSR\_MEMFAULTENA\_MSK.

##### 10.7.2.7.29 dauanahw-HwInit-LLR-030

Requirement ID: H398-LLD-ANA-FNC-1887

Set ccr of M\_SCB to ccr of M\_SCB bitwise OR with M\_DIVIDE\_ZERO

##### 10.7.2.7.30 dauanahw-HwInit-LLR-028

Requirement ID: H398-LLD-ANA-FNC-471

The function shall initialize GPIO pins by calling 'InitIO'.

## 10.8 dauanaiadc

This module contains Implementation of Internal ADC.

### 10.8.1 IadcRead

Low Level Design Details about CSU IadcRead will follow in the sub sections.

#### 10.8.1.1 Brief Description

The function IadcRead returns average reading of the selected Internal ADC channel passed as parameter.

#### 10.8.1.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.8.1.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 10.8.1.4 Parameter list (Input/Output)

Inputs : T\_UINT8 u8\_chan : ADC channel number

Outputs : None

#### 10.8.1.5 Return Value

T\_SINT16-Return the average reading of the corresponding channel.

#### 10.8.1.6 Other CSUs called by this CSU

None

#### 10.8.1.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to IadcRead.

##### 10.8.1.7.1 dauanaiadc-IadcRead-LLR-001

Requirement ID: H398-LLD-ANA-FNC-481

The function shall return the (average reading of Internal ADC of the corresponding channel(u8\_chan) right shift by M\_IADC\_POWER).

### 10.8.2 TransferCompleteIntr

Low Level Design Details about CSU TransferCompleteIntr will follow in the sub sections.

#### 10.8.2.1 Brief Description

The function TransferCompleteIntr clears the pending interrupt bits of DMA2 stream 0 and apply filter to the new reading of Internal ADC.

#### 10.8.2.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.8.2.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 10.8.2.4 Parameter list (Input/Output)

Inputs : None

Outputs : None

#### 10.8.2.5 Return Value

None

#### 10.8.2.6 Other CSUs called by this CSU

DmaClearItPendingBit

#### 10.8.2.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to TransferCompleteIntr.

##### 10.8.2.7.1 dauanaiadc-TransferCompleteIntr-LLR-001

Requirement ID: H398-LLD-ANA-FNC-490

The function shall Clear the DMA2 STREAM0's interrupt pending bit by calling 'DmaClearItPendingBit' with M\_DMA2\_STREAM0 and M\_DMA\_IT\_TCIF0 as parameters.

##### 10.8.2.7.2 dauanaiadc-TransferCompleteIntr-LLR-002

Requirement ID: H398-LLD-ANA-FNC-491

The function shall apply filter to the new reading of IADC for all the IADC channels as follows by looping through M\_ZERO to M\_IADC\_CHANS-one:

previous average reading of the channel added to (Reading of Internal ADC subtracted by (average reading of Internal ADC of corresponding channel right shifted by M\_IADC\_POWER))

Note: Initial average reading of the channel is Zero

### 10.8.3 ConfigureDma

Low Level Design Details about CSU ConfigureDma will follow in the sub sections.

#### 10.8.3.1 Brief Description

This function Initializes the DMA2 Stream 0 for ADC 1 communication.

#### 10.8.3.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.8.3.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 10.8.3.4 Parameter list (Input/Output)

Inputs : None

Outputs : None

#### 10.8.3.5 Return Value

None

#### 10.8.3.6 Other CSUs called by this CSU

RccAhb1PeriphClockCmd

IntrInstall,

NvicInit

DmaItConfig

DmaInit

DmaCmd

TransferCompleteIntr

#### 10.8.3.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to ConfigureDma.

10.8.3.7.1 dauanaiadc-ConfigureDma-LLR-001

Requirement ID: H398-LLD-ANA-FNC-500

The function shall enable the AHB1 peripheral clock for DMA 2 by calling 'RccAhb1PeriphClockCmd' with parameters M\_RCC\_AHB1PERIPH\_DMA2, ENABLE.

##### 10.8.3.7.2 dauanaiadc-ConfigureDma-LLR-002

Requirement ID: H398-LLD-ANA-FNC-501

The function shall load the reference to the TransferCompleteIntr into the vector table for DMA 2 stream 0 interrupt by calling 'IntrInstall' with parameters INTR\_DMA\_2\_STREAM\_0 and function TransferCompleteIntr.

##### 10.8.3.7.3 dauanaiadc-ConfigureDma-LLR-003

Requirement ID: H398-LLD-ANA-FNC-502

The function shall initialize NVIC peripheral by calling 'NvicInit' with reference to nvic init structure as a parameter with the members initialized as follows

a) nvic\_irq\_channel is set with DMA2\_STREAM0\_IRQN,

b) nvic\_irq\_channel\_preemption\_priority is set with NVIC\_PRIORITY\_LEVEL\_8,

c) nvic\_irq\_channel\_subpriority is set with NVIC\_PRIORITY\_LEVEL\_0

d) nvic\_irq\_channel\_cmd is set with ENABLE.

##### 10.8.3.7.4 dauanaiadc-ConfigureDma-LLR-004

Requirement ID: H398-LLD-ANA-FNC-503

The function shall enable DMA2 STREAM0 interrupt by calling 'DmaItConfig' with parameters M\_DMA2\_STREAM0, M\_DMA\_IT\_TC and ENABLE.

##### 10.8.3.7.5 dauanaiadc-ConfigureDma-LLR-005

Requirement ID: H398-LLD-ANA-FNC-504

The function shall initialize DMA2 STREAM0 by calling 'DmaInit' with M\_DMA2\_STREAM0 and reference to dma init structure as parameters with the members initialized as follows

a) dma\_channel is set with M\_DMA\_CHANNEL\_0

b) dma\_memory0\_baseaddr is set to Reading of Internal ADC,

c) dma\_peripheral\_baseaddr is set with reference of dr of M\_ADC1,

d) dma\_dir is set with M\_DMA\_DIR\_PERIPHERAL\_TO\_MEMORY,

e) dma\_buffersize is set with M\_IADC\_CHANS,

f) dma\_peripheral\_inc is set with M\_DMA\_PERIPHERALINC\_DISABLE,

g) dma\_memory\_inc is set with M\_DMA\_MEMORYINC\_ENABLE,

h) dma\_peripheral\_datasize is set with M\_DMA\_PERIPH\_DATASIZE\_HALFWORD,

i) dma\_memory\_datasize is set with M\_DMA\_MEMORY\_DATASIZE\_HALFWORD,

j) dma\_mode is set with M\_DMA\_MODE\_CIRCULAR,

k) dma\_priority is set with M\_DMA\_PRIORITY\_MEDIUM,

l) dma\_fifo\_mode is set with M\_DMA\_FIFOMODE\_DISABLE,

m) dma\_fifo\_threshold is set with M\_DMA\_FIFOTHRESHOLD\_FULL,

n) dma\_memory\_burst is set with M\_DMA\_MEMORYBURST\_SINGLE and

o) dma\_peripheral\_burst is set with M\_DMA\_PERIPHERALBURST\_SINGLE.

##### 10.8.3.7.6 dauanaiadc-ConfigureDma-LLR-006

Requirement ID: H398-LLD-ANA-FNC-505

The function shall enable DMA2 STREAM0 by calling 'DmaCmd' with parameters M\_DMA2\_STREAM0 and ENABLE.

### 10.8.4 TriggerInternalAdc

Low Level Design Details about CSU TriggerInternalAdc will follow in the sub sections.

#### 10.8.4.1 Brief Description

This function clears TIM 6 interrupt pending bit and triggers the ADC 1 for software conversion.

#### 10.8.4.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.8.4.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 10.8.4.4 Parameter list (Input/Output)

Inputs : None

Outputs : None

#### 10.8.4.5 Return Value

None

#### 10.8.4.6 Other CSUs called by this CSU

TimClearITPendingBit

AdcSoftwareStartConv

#### 10.8.4.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to TriggerInternalAdc.

##### 10.8.4.7.1 dauanaiadc-TriggerInternalAdc-LLR-001

Requirement ID: H398-LLD-ANA-FNC-514

The function shall clear TIM6 interrupt pending bit by calling 'TimClearITPendingBit' with parameters M\_TIM6 and M\_TIM\_IT\_UPDATE.

##### 10.8.4.7.2 dauanaiadc-TriggerInternalAdc-LLR-002

Requirement ID: H398-LLD-ANA-FNC-515

The function shall set Start conversion of regular channels bit in ADC control register 2 for ADC 1 by calling 'AdcSoftwareStartConv' with parameter M\_ADC1.

### 10.8.5 IadcInit

Low Level Design Details about CSU IadcInit will follow in the sub sections.

#### 10.8.5.1 Brief Description

The function IadcInit initializes ADC 1 and DMA 2 for internal ADC.

#### 10.8.5.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.8.5.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 10.8.5.4 Parameter list (Input/Output)

Inputs : None

Outputs : None

#### 10.8.5.5 Return Value

None

#### 10.8.5.6 Other CSUs called by this CSU

RccApb2PeriphClockCmd

AdcCommonInit

ConfigureDma

AdcInit

AdcRegularChannelConfig

AdcDmaReqAfterLastTransferCmd

AdcDmaCmd

AdcCmd

TMRInitTIM6

TriggerInternalAdc

#### 10.8.5.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to IadcInit.

##### 10.8.5.7.1 dauanaiadc-IadcInit-LLR-001

Requirement ID: H398-LLD-ANA-FNC-524

The function shall call the function 'RccApb2PeriphClockCmd’ with parameter (M\_RCC\_APB2PERIPH\_ADC1, ENABLE) to Enable ADC 1 clock.

##### 10.8.5.7.2 dauanaiadc-IadcInit-LLR-002

Requirement ID: H398-LLD-ANA-FNC-525

The function shall set fields of ADC common control register as follows

a) Multi ADC mode selection field is set with M\_ADC\_MODE\_INDEPENDENT,

b) ADC prescaler field is set with M\_ADC\_PRESCALER\_DIV8,

c) Direct memory access mode for multi ADC mode field is set with M\_ADC\_DMAACCESSMODE\_DISABLED,

d) Delay between 2 sampling phases field is set with M\_ADC\_TWOSAMPLINGDELAY\_5CYCLES.

##### 10.8.5.7.3 dauanaiadc-IadcInit-LLR-003

Requirement ID: H398-LLD-ANA-FNC-526

The function shall initialize ADC common control register by calling 'AdcCommonInit' with parameter address of ADC common control register.

##### 10.8.5.7.4 dauanaiadc-IadcInit-LLR-004

Requirement ID: H398-LLD-ANA-FNC-527

The function shall initialize DMA 2 for ADC by calling 'ConfigureDma'.

##### 10.8.5.7.5 dauanaiadc-IadcInit-LLR-005

Requirement ID: H398-LLD-ANA-FNC-528

The function shall set fields of ADC init structure as follows

a) Resolution field is set with M\_ADC\_RESOLUTION\_12B,

b) Scan mode field is set with ENABLE.

##### 10.8.5.7.6 dauanaiadc-IadcInit-LLR-006

Requirement ID: H398-LLD-ANA-FNC-529

The function shall set fields of ADC init structure as follows

a) Continuous conversion field is set with DISABLE,

b) External trigger enable for regular channels field is set with M\_ADC\_EXTN\_TRIG\_CONV\_EDGE\_NONE,

c) External event select for regular group field is set with M\_ADC\_EXTERNALTRIGCONV\_T1\_CC1,

d) Data alignment field is set with M\_ADC\_DATAALIGN\_RIGHT.

##### 10.8.5.7.7 dauanaiadc-IadcInit-LLR-007

Requirement ID: H398-LLD-ANA-FNC-530

The function shall set Regular channel sequence length field with M\_IADC\_CHANS of ADC init structure .

##### 10.8.5.7.8 dauanaiadc-IadcInit-LLR-008

Requirement ID: H398-LLD-ANA-FNC-531

The function shall initialize ADC 1 by calling 'AdcInit' with parameters M\_ADC1 and address of ADC init structure.

##### 10.8.5.7.9 dauanaiadc-IadcInit-LLR-009

Requirement ID: H398-LLD-ANA-FNC-532

The function shall configure DMA 10 for ADC 1 by calling 'AdcRegularChannelConfig' with parameters M\_ADC1, M\_ADC\_CHANNEL\_10, M\_ONE and M\_ADC\_SAMPLETIME\_56CYCLES.

##### 10.8.5.7.10 dauanaiadc-IadcInit-LLR-010

Requirement ID: H398-LLD-ANA-FNC-533

The function shall configure DMA 11 for ADC 1 by calling 'AdcRegularChannelConfig' with parameters M\_ADC1, M\_ADC\_CHANNEL\_11, M\_TWO and M\_ADC\_SAMPLETIME\_56CYCLES.

##### 10.8.5.7.11 dauanaiadc-IadcInit-LLR-011

Requirement ID: H398-LLD-ANA-FNC-534

The function shall enable DMA request after last transfer by calling 'AdcDmaReqAfterLastTransferCmd' with parameters M\_ADC1 and ENABLE.

##### 10.8.5.7.12 dauanaiadc-IadcInit-LLR-012

Requirement ID: H398-LLD-ANA-FNC-535

The function shall enable DMA for ADC 1 by calling 'AdcDmaCmd' with parameters M\_ADC1 and ENABLE.

##### 10.8.5.7.13 dauanaiadc-IadcInit-LLR-013

Requirement ID: H398-LLD-ANA-FNC-536

The function shall enable ADC 1 by calling 'AdcCmd' with parameters M\_ADC1 and ENABLE.

##### 10.8.5.7.14 dauanaiadc-IadcInit-LLR-014

Requirement ID: H398-LLD-ANA-FNC-537

The function shall configure timer 6 to trigger internal ADC converting rate by calling 'TMRInitTIM6' with to function TriggerInternalAdc as parameter.

## 10.9 dauanainit

This module Initializes the tasks routines.

### 10.9.1 InitTask

Low Level Design Details about CSU InitTask will follow in the sub sections.

#### 10.9.1.1 Brief Description

The function InitTask initializes all the tasks:

- Time base of the system

- ARINC 825 interface

- Poller task

- Internal ADC hardware

- External M\_ADC hardware

- Application task

- Heart beat LED.

#### 10.9.1.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.9.1.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 10.9.1.4 Parameter list (Input/Output)

Inputs : void \* p\_data ->Reserve Future use

Outputs : void \* p\_data ->Reserve Future use

#### 10.9.1.5 Return Value

None

#### 10.9.1.6 Other CSUs called by this CSU

WdogInit

TbaseInit

A825Init

IadcInit

A825CommInit

XADCInit

APPInit

CbitInit

OsSemPend

GpioToggleBits

WdogKickWatchDog

TACHInit

#### 10.9.1.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to InitTask.

##### 10.9.1.7.1 dauanainit-InitTask-LLR-001

Requirement ID: H398-LLD-ANA-FNC-547

The function shall do the following:

a. Call ' WdogInit ' to initialize the independent watchdog.

b. Call ' SSPIinitSPI2 ' to initialize SPI2 bus.

b. Call ' WdogKickWatchDog ' to Reload watchdog.

##### 10.9.1.7.2 dauanainit-InitTask-LLR-002

Requirement ID: H398-LLD-ANA-FNC-548

The function shall do the following:

a. Call 'TbaseInit' to initialize time base

b. Call 'WdogKickWatchDog' to Reload watchdog.

##### 10.9.1.7.3 dauanainit-InitTask-LLR-003

Requirement ID: H398-LLD-ANA-FNC-549

The function shall do the following:

a. Call 'A825Init' to initialize ARINC 825 interface.

b. Call 'WdogKickWatchDog' to Reload watchdog.

##### 10.9.1.7.4 dauanainit-InitTask-LLR-004

Requirement ID: H398-LLD-ANA-FNC-550

The function shall do the following:

a. Call 'IadcInit' to Initialize Internal ADC hardware.

b. Call 'WdogKickWatchDog' to Reload watchdog.

##### 10.9.1.7.5 dauanainit-InitTask-LLR-005

Requirement ID: H398-LLD-ANA-FNC-551

The function shall do the following:

a. Call 'A825CommInit' to Initialize Arinc 825 Task.

b. Call 'WdogKickWatchDog' to Reload watchdog.

##### 10.9.1.7.6 dauanainit-InitTask-LLR-006

Requirement ID: H398-LLD-ANA-FNC-552

The function shall do the following:

a. Call 'XADCInit' to Initialize external ADC hardware.

b. Call 'WdogKickWatchDog' to Reload watchdog.

c. Call ‘TACHInit’ to Initialize tach channels.

d. Call 'WdogKickWatchDog' to Reload watchdog.

e. Call 'XDACinit' to Initialize XDAC channels.

f. Call 'WdogKickWatchDog' to Reload watchdog.

##### 10.9.1.7.7 dauanainit-InitTask-LLR-007

Requirement ID: H398-LLD-ANA-FNC-553

The function shall do the following:

a. Call 'APPInit' to Initialize application task.

b. Call 'WdogKickWatchDog' to Reload watchdog.

##### 10.9.1.7.8 dauanainit-InitTask-LLR-008

Requirement ID: H398-LLD-ANA-FNC-554

The function shall do the following:

a. Call 'CbitInit' to Initialize CBIT task.

b. Call 'WdogKickWatchDog' to Reload watchdog.

##### 10.9.1.7.9 dauanainit-InitTask-LLR-009

Requirement ID: H398-LLD-ANA-FNC-555

The function shall loop infinitely and perform the following operations within loop

a) pend on semaphore by calling OsSemPend with the following parameters

- Allocated semaphore for the task sem init task

- pend timeout value as M\_ZERO

- Address of local error code.

b) call WdogKickWatchDog to reload watchdog counter.

c) call GpioToggleBits (M\_HW\_LED\_HB\_TOGGLE) with reference to GPIO port B (M\_GPIOB) and M\_GPIOB\_LED\_HB (GPIO pin 2 (M\_GPIO\_PIN\_2)) as parameters to toggle the heartbeat LED.

### 10.9.2 InitInit

Low Level Design Details about CSU InitInit will follow in the sub sections.

#### 10.9.2.1 Brief Description

This function installs semaphore into timebase and creates init OS task

#### 10.9.2.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.9.2.3 List of global variables accessed and modified

Accessed : Init\_task\_stk

Modified : None

#### 10.9.2.4 Parameter list (Input/Output)

Inputs : None

Outputs : None

#### 10.9.2.5 Return Value

None

#### 10.9.2.6 Other CSUs called by this CSU

TbaseTaskSignaling

OsTaskCreate

OsSemCreate

ErrorHandler

InitTask

#### 10.9.2.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to InitInit.

##### 10.9.2.7.1 dauanainit-InitInit-LLR-001

Requirement ID: H398-LLD-ANA-FNC-564

The function shall call 'OsSemCreate' with parameter (M\_ZERO) to create a semaphore and call the function 'ErrorHandler' when the function 'OsSemCreate' returns M\_NULL otherwise do nothing.

##### 10.9.2.7.2 dauanainit-InitInit-LLR-002

Requirement ID: H398-LLD-ANA-FNC-565

The function shall install task signaling parameters for the init task by calling 'TbaseTaskSignaling' with parameters.

a) M\_INIT\_TASK\_TICKS as task ticks and

b) return value of function OsSemCreate with parameter M\_ZERO.

##### 10.9.2.7.3 dauanainit-InitInit-LLR-003

Requirement ID: H398-LLD-ANA-FNC-566

The function shall create the Init Task as the first task to run RTOS by calling 'OsTaskCreate' with parameters (Pointer to function InitTask, M\_HW\_NULL as task entry point, Reference to top of stack of Init task stack (i.e. Init\_task\_stk), M\_INIT\_TASK\_PRIO as task priority) and call the function 'ErrorHandler' when the function 'OsTaskCreate' returns other than M\_OS\_NO\_ERR otherwise do nothing.

## 10.10 dauanaintr

This module initializes the vector table and allows installing and uninstalling interrupts into the table.

### 10.10.1 ResetIsr

Low Level Design Details about CSU ResetIsr will follow in the sub sections.

#### 10.10.1.1 Brief Description

The function ResetIsr is the reset routine on power-up.

#### 10.10.1.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.10.1.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 10.10.1.4 Parameter list (Input/Output)

Inputs : None

Outputs : None

#### 10.10.1.5 Return Value

None

#### 10.10.1.6 Other CSUs called by this CSU

Crt0TransferData

#### 10.10.1.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to ResetIsr.

##### 10.10.1.7.1 dauanaintr-ResetIsr-LLR-001

Requirement ID: H398-LLD-ANA-FNC-576

The function shall call the function 'Crt0TransferData' to run the reset routine.

### 10.10.2 NonMaskable

Low Level Design Details about CSU NonMaskable will follow in the sub sections.

#### 10.10.2.1 Brief Description

The function NonMaskable is an interrupt error handler for nonMaskable errors.

#### 10.10.2.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.10.2.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 10.10.2.4 Parameter list (Input/Output)

Inputs : None

Outputs : None

#### 10.10.2.5 Return Value

None

#### 10.10.2.6 Other CSUs called by this CSU

ErrorHandler

#### 10.10.2.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to NonMaskable.

##### 10.10.2.7.1 dauanaintr-NonMaskable-LLR-001

Requirement ID: H398-LLD-ANA-FNC-585

The function shall call the function 'ErrorHandler' to handle the CBIT errors.

### 10.10.3 HardFault

Low Level Design Details about CSU HardFault will follow in the sub sections.

#### 10.10.3.1 Brief Description

The function HardFault is an exception that occurs because of an error during exception processing, or because an exception cannot be managed by any other exception handling mechanism. HardFaults have a fixed priority of -1, meaning it have higher priority than any exception with configurable priority.

#### 10.10.3.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.10.3.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 10.10.3.4 Parameter list (Input/Output)

Inputs : None

Outputs : None

#### 10.10.3.5 Return Value

None

#### 10.10.3.6 Other CSUs called by this CSU

ErrorHandler

#### 10.10.3.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to HardFault.

##### 10.10.3.7.1 dauanaintr-HardFault-LLR-001

Requirement ID: H398-LLD-ANA-FNC-594

The function shall call the function 'ErrorHandler' to handle the CBIT errors.

### 10.10.4 MemManage

Low Level Design Details about CSU MemManage will follow in the sub sections.

#### 10.10.4.1 Brief Description

The function MemManage is an interrupt error handler for memory management errors.

#### 10.10.4.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.10.4.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 10.10.4.4 Parameter list (Input/Output)

Inputs : None

Outputs : None

#### 10.10.4.5 Return Value

None

#### 10.10.4.6 Other CSUs called by this CSU

ErrorHandler

#### 10.10.4.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to MemManage.

##### 10.10.4.7.1 dauanaintr-MemManage-LLR-001

Requirement ID: H398-LLD-ANA-FNC-603

The function shall call the function 'ErrorHandler' to handle the CBIT errors.

##### 10.10.4.7.2 dauanaintr-MemManage-LLR-002

Requirement ID: H398-LLD-ANA-FNC-2440

The function shall perform the following

1. Performs assembly instruction (mrs) to move the value in process stack pointer(psp) into the general-purpose register (R0) when BITWISE AND of lr and 0x04 returns true.Otherwise move the value in main stack pointer(msp) into the general-purpose register (R0).
2. Set u8\_data of memory management to reference of p\_mmfsr.
3. Set err\_loc of memory management error to reference of p\_mmar when b\_mmar\_valid of memory management error returns true otherwise set err\_loc of memory management to M\_ZERO.
4. Set fault\_instruction\_loc of memory management error to sp with index of SIX when BITWISE OR of (b\_inst\_acc\_viol of memory management error and b\_data\_acc\_viol of memory management error) returns TRUE, otherwise set fault\_instruction\_loc of memory management error to M\_ZERO.
5. Set xpsr of memory management error to sp with index of SEVEN.

### 10.10.5 BusFault

Low Level Design Details about CSU BusFault will follow in the sub sections.

#### 10.10.5.1 Brief Description

The function BusFault is an interrupt error handler for bus fault errors.

#### 10.10.5.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.10.5.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 10.10.5.4 Parameter list (Input/Output)

Inputs : None

Outputs : None

#### 10.10.5.5 Return Value

None

#### 10.10.5.6 Other CSUs called by this CSU

ErrorHandler

#### 10.10.5.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to BusFault.

##### 10.10.5.7.1 dauanaintr-BusFault-LLR-001

Requirement ID: H398-LLD-ANA-FNC-612

The function shall call the function 'ErrorHandler' to handle the CBIT errors.

### 10.10.6 UsageFault

Low Level Design Details about CSU UsageFault will follow in the sub sections.

#### 10.10.6.1 Brief Description

The function UsageFault is an interrupt error handler for usage fault errors.

#### 10.10.6.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.10.6.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 10.10.6.4 Parameter list (Input/Output)

Inputs : None

Outputs : None

#### 10.10.6.5 Return Value

None

#### 10.10.6.6 Other CSUs called by this CSU

ErrorHandler

#### 10.10.6.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to UsageFault.

##### 10.10.6.7.1 dauanaintr-UsageFault-LLR-001

Requirement ID: H398-LLD-ANA-FNC-621

The function shall call the function 'ErrorHandler' to handle the CBIT errors.

### 10.10.7 SpuriousInterrupt

Low Level Design Details about CSU SpuriousInterrupt will follow in the sub sections.

#### 10.10.7.1 Brief Description

The function SpuriousInterrupt is an interrupt error handler for spurious interrupts.

#### 10.10.7.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.10.7.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 10.10.7.4 Parameter list (Input/Output)

Inputs : None

Outputs : None

#### 10.10.7.5 Return Value

None

#### 10.10.7.6 Other CSUs called by this CSU

ErrorHandler

#### 10.10.7.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to SpuriousInterrupt.

##### 10.10.7.7.1 dauanaintr-SpuriousInterrupt-LLR-001

Requirement ID: H398-LLD-ANA-FNC-1885

The function shall call the function 'ErrorHandler' to handle the CBIT errors.

### 10.10.8 IntrInit

Low Level Design Details about CSU IntrInit will follow in the sub sections.

#### 10.10.8.1 Brief Description

The function IntrInit shall copies the ISR functions from ROM to RAM.

#### 10.10.8.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.10.8.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 10.10.8.4 Parameter list (Input/Output)

Inputs : None

Outputs : None

#### 10.10.8.5 Return Value

None

#### 10.10.8.6 Other CSUs called by this CSU

None

#### 10.10.8.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to IntrInit.

##### 10.10.8.7.1 dauanaintr-IntrInit-LLR-001

Requirement ID: H398-LLD-ANA-FNC-630

The function shall loop from index INTR\_RESET to INTR\_MAX-1 and copy all interrupt function from ROM to RAM when interrupt function in ROM is not equal to M\_NULL, otherwise copy spurious interrupt handler to RAM

### 10.10.9 IntrInstall

Low Level Design Details about CSU IntrInstall will follow in the sub sections.

#### 10.10.9.1 Brief Description

The function IntrInstall loads the ISR into the vector table for the received interrupt index.

#### 10.10.9.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.10.9.3 List of global variables accessed and modified

Accessed : U32\_critical\_sr

Modified : U32\_critical\_sr

#### 10.10.9.4 Parameter list (Input/Output)

Inputs : T\_UINT8 u8\_index - interrupt number index in the vector table,

T\_INTR\_FN intr\_function - ISR fucntion.

Outputs : None

#### 10.10.9.5 Return Value

None

#### 10.10.9.6 Other CSUs called by this CSU

SaveStatusReg

RestoreStatusReg

#### 10.10.9.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to IntrInstall.

##### 10.10.9.7.1 dauanaintr-IntrInstall-LLR-001

Requirement ID: H398-LLD-ANA-FNC-639

The function shall call ‘SaveStatusReg’ (M\_OS\_ENTER\_CRITICAL) to disable interrupts before loading the ISR into the interrupt vector table and return value gets stored in U32\_critical\_sr.

##### 10.10.9.7.2 dauanaintr-IntrInstall-LLR-002

Requirement ID: H398-LLD-ANA-FNC-640

The function shall store the intr\_function into index u8\_index of the interrupt vector table in RAM when u8\_index is less than INTR\_MAX otherwise do nothing.

##### 10.10.9.7.3 dauanaintr-IntrInstall-LLR-003

Requirement ID: H398-LLD-ANA-FNC-641

The function shall call ‘RestoreStatusReg’ (M\_OS\_EXIT\_CRITICAL) with parameter U32\_critical\_sr to restore the priority mask register after the loading is complete.

### 10.10.10 IntrGetCurrentIntrFn

Low Level Design Details about CSU IntrGetCurrentIntrFn will follow in the sub sections.

#### 10.10.10.1 Brief Description

The function IntrGetCurrentIntrFn returns the current installed interrupt function.

#### 10.10.10.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.10.10.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 10.10.10.4 Parameter list (Input/Output)

Inputs : T\_UINT8 u8\_index- interrupt number index in the vector table,

Outputs : None

#### 10.10.10.5 Return Value

T\_INTR\_FN- ISR function

#### 10.10.10.6 Other CSUs called by this CSU

None

#### 10.10.10.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to IntrGetCurrentIntrFn.

##### 10.10.10.7.1 dauanaintr-IntrGetCurrentIntrFn-LLR-001

Requirement ID: H398-LLD-ANA-FNC-650

The function shall Set intr function to Lfntable ram with index u8 index when u8 index is less than INTR\_MAX and return intr function.

## 10.11 dauanalookup

The dauanalookup CSC contains Implementation of Lookup Table for scale reading of the Analog sensors.

### 10.11.1 LookupTableLookup

Low Level Design Details about CSU LookupTableLookup will follow in the sub sections.

#### 10.11.1.1 Brief Description

The function LookupTableLookup calculates the Sensor unit value for the corresponding ADC reading.

#### 10.11.1.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD(H398-003-012-ANA).

#### 10.11.1.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 10.11.1.4 Parameter list (Input/Output)

Inputs: T\_SENSOR\_TABLE \* ps\_Table -> Reference to the Sensor table

T\_SINT16 i16\_Val ->Actual ADC reading for the sensor

T\_UINT8 \* pu8\_Stat ->Status flag

Outputs : T\_UINT8 \* pu8\_Stat ->Status flag

Note:

Reference to the Sensor table from module configuration data.

#### 10.11.1.5 Return Value

T\_SINT16 - Calculated Sensor unit value for the corresponding ADC reading

#### 10.11.1.6 Other CSUs called by this CSU

None

#### 10.11.1.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to LookupTableLookup.

##### 10.11.1.7.1 dauanalookup-LookupTableLookup-LLR-001

Requirement ID: H398-LLD-ANA-FNC-660

The function shall perform as following:

1. Set number of segments to content of pi16X of ps Table.
2. Perform as follows when i16 Valuve is less than or equal to content of (pi16X of ps Table added with M\_ONE).
3. Set pu8 State to TRUE when i16 Valuve is equal to content of (pi16X of ps Table added with M\_ONE).
4. Set pu8 State to FALSE when i16 Valuve is less than content of (pi16X of ps Table added with M\_ONE).
5. return content of (pi16Y of ps Table added with M\_ONE).

##### 10.11.1.7.2 dauanalookup-LookupTableLookup-LLR-002

Requirement ID: H398-LLD-ANA-FNC-661

The function shall perform as following when i16 Valuve is greater than or equal to content of (pi16X of ps Table added with M\_ONE added with number of segments subtracted by M\_ONE):

1. Set pu8 State to TRUE when i16 Valuve is equal to content of (pi16X of ps Table added with M\_ONE added with number of segments subtracted by M\_ONE).
2. Set pu8 State to FALSE when i16 Valuve is greater than content of (pi16X of ps Table added with M\_ONE added with number of segments subtracted by M\_ONE).
3. return content (pi16Y of ps Table added with M\_ONE added with number of segments subtracted by M\_ONE).

##### 10.11.1.7.3 dauanalookup-LookupTableLookup-LLR-003

Requirement ID: H398-LLD-ANA-FNC-662

The function shall perform as follows when i16 Valuve is greater than content of (pi16X of ps Table added with M\_ONE) and less than content of (pi16X of ps Table added with M\_ONE added with number of segments subtracted by M\_ONE):

1. Set pu8 State to TRUE.
2. Loop until (negated value of (i16 Valuve is greater than or equal to pi16X of ps Table added with M\_ONE) AND (i16 Valuve is less than or equal to content of (pi16X of ps Table added with M\_ONE added with M\_ONE)) and perform as follows:

* Set number of segments to (number of segments right shift by M\_ONE) when number of segments is greater than M\_ONE otherwise do nothing.

##### 10.11.1.7.4 dauanalookup-LookupTableLookup-LLR-004

Requirement ID: H398-LLD-ANA-FNC-663

The function shall perform as following when i16 Valuve is greater than content of (pi16X of ps Table added with M\_ONE) and less than content of (pi16X of ps Table added with M\_ONE added with number of segments subtracted by M\_ONE):

Loop until (negated value of (i16 Valuve is greater than or equal to pi16X of ps Table added with M\_ONE) AND (i16 Valuve is less than or equal to content of (pi16X of ps Table added with M\_ONE added with M\_ONE)) and perform as follows:

1. Set pi16X of ps Table added with M\_ONE to (pi16X of ps Table added with M\_ONE added with number of segments)
2. Set pi16Y of ps Table added with M\_ONE to (pi16Y of ps Table added with M\_ONE added with number of segments)

When i16 Valuve is greater than or equal to pi16X of ps Table added with M\_ONE.

##### 10.11.1.7.5 dauanalookup-LookupTableLookup-LLR-005

Requirement ID: H398-LLD-ANA-FNC-664

The function shall perform as following when i16 Valuve is greater than content of (pi16X of ps Table added with M\_ONE) and less than content of (pi16X of ps Table added with M\_ONE added with number of segments subtracted by M\_ONE):

Loop until (negated value of (i16 Valuve is greater than or equal to pi16X of ps Table added with M\_ONE) and (i16 Valuve is less than or equal to content of (pi16X of ps Table added with M\_ONE added with M\_ONE)) and perform as follows:

1. Set pi16X of ps Table added with M\_ONE to (pi16X of ps Table added with M\_ONE subtracted with number of segments)
2. Set pi16Y of ps Table added with M\_ONE to (pi16Y of ps Table added with M\_ONE subtracted with number of segments)

When i16 Valuve is less than pi16X of ps Table added with M\_ONE.

##### 10.11.1.7.6 dauanalookup-LookupTableLookup-LLR-006

Requirement ID: H398-LLD-ANA-FNC-665

The function shall perform as following:

* Set segment x to (content of (pi16X of ps Table added with M\_ONE added with M\_ONE) subtracted with content of pi16X of ps Table added with M\_ONE).
* Set delta x to i16 Valuve subtracted with content of pi16X of ps Table added with M\_ONE.
* Set segment y to (content of (pi16Y of ps Table added with M\_ONE added with M\_ONE) subtracted with content of pi16Y of ps Table added with M\_ONE).

##### 10.11.1.7.7 dauanalookup-LookupTableLookup-LLR-007

Requirement ID: H398-LLD-ANA-FNC-666

The function shall return ((delta x multiplied with segment y) divided by segment x added with content of pi16Y of ps Table added with M\_ONE).

### 10.11.2 LookupTableInverse

Low Level Design Details about CSU LookupTableInverse will follow in the sub sections.

#### 10.11.2.1 Brief Description

The function LookupTableInverse swaps the rows and columns of lookup table.

#### 10.11.2.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD(H398-003-012-ANA).

#### 10.11.2.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 10.11.2.4 Parameter list (Input/Output)

Inputs : T\_SENSOR\_TABLE \*table -> Reference to the Sensor table

T\_SINT16 dep\_var -> Independent Variable to find Dependent value

T\_UINT8 \*stat-> Reference to Lookup status data types

Outputs : None

Note:

Reference to the Sensor table from module configuration data.

#### 10.11.2.5 Return Value

T\_SINT16 - Return the results of the Lookup of the swapped tables.

#### 10.11.2.6 Other CSUs called by this CSU

LookupTableLookup

#### 10.11.2.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to LookupTableInverse.

##### 10.11.2.7.1 dauanalookup-LookupTableInverse-LLR-001

Requirement ID: H398-LLD-ANA-FNC-675

The function shall return with M\_ZERO when lookup table is empty or lookup status is empty i.e. pi16X of table OR pi16Y of table OR stat is equal to M\_ZERO.

##### 10.11.2.7.2 dauanalookup-LookupTableInverse-LLR-002

Requirement ID: H398-LLD-ANA-FNC-676

The function shall do the following when lookup table or lookup status is not empty i.e. pi16X of table AND pi16Y of table AND stat is not equal to M\_ZERO:

1. Set first set of swap variables to pi16Y of table
2. Set second set of swap variables to pi16X of table

##### 10.11.2.7.3 dauanalookup-LookupTableInverse-LLR-003

Requirement ID: H398-LLD-ANA-FNC-677

The function shall return the value returned by LookupTableLookup function with parameters reference to swapped lookup table, dep\_var and stat.

### 10.11.3 LookupTableLookup32

Low Level Design Details about CSU LookupTableLookup32 will follow in the sub sections.

#### 10.11.3.1 Brief Description

The function LookupTableLookup32 calculates the Sensor unit value for the corresponding ADC reading.

#### 10.11.3.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD(H398-003-012-ANA).

#### 10.11.3.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 10.11.3.4 Parameter list (Input/Output)

Inputs: T\_SENSOR\_TABLE\_32 \*table

T\_SINT32 ind\_var

T\_LOOKUP\_STAT \*stat

Outputs : T\_LOOKUP\_STAT \*stat

Note:

Reference to the Sensor table from module configuration data.

#### 10.11.3.5 Return Value

T\_SINT16 - Return the Sensor unit value for the calculated 'index's

#### 10.11.3.6 Other CSUs called by this CSU

None

#### 10.11.3.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to LookupTableLookup32.

##### 10.11.3.7.1 dauanalookup-LookupTableLookup32-LLR-001

Requirement ID: H398-LLD-ANA-FNC-1917

The function shall return M\_ZERO when pi32X of table is equal to M\_HW\_NULL logical OR with pi32y of table is equal to M\_HW\_NULL logical OR with stat is equal to M\_HW\_NULL.

##### 10.11.3.7.2 dauanalookup-LookupTableLookup32-LLR-002

Requirement ID: H398-LLD-ANA-FNC-1918

The function shall perform the following when Not of (pi32X of table is equal to M\_HW\_NULL logical OR with pi32y of table is equal to M\_HW\_NULL logical OR with stat is equal to M\_HW\_NULL) .

* snum points is set to value at reference of Array x incremented by one
* increment reference of Array y by one.
* S lastx is set to Array x with index as difference of snum points with M\_ONE.
* rl temp is set to Array x.
* perform the following when rl temp is lessthan or equal to Slastx and independent variable is less than rl temp.
* Estatus is set to LS\_OUTSIDE.
* Stat is set to Estatus.
* Return Array y.

##### 10.11.3.7.3 dauanalookup-LookupTableLookup32-LLR-003

Requirement ID: H398-LLD-ANA-FNC-1919

The function shall perform the following when Not of (pi32X of table is equal to M\_HW\_NULL logical OR with pi32y of table is equal to M\_HW\_NULL logical OR with stat is equal to M\_HW\_NULL) and rl temp is less than or equal to Slastx and ind\_var is greater than or equal to Slastx.

* Set reference of Array y to addition of reference of Array y and (snum points minus M\_ONE).
* Estatus is set to LS\_OUTSIDE when ind\_var is greater than Slastx, otherwise do nothing.
* Set Stat to Estatus.
* Return of array y.

##### 10.11.3.7.4 dauanalookup-LookupTableLookup32-LLR-004

Requirement ID: H398-LLD-ANA-FNC-1920

The function shall loop until (NOT of (ind\_var is less than rl temp) AND (ind\_var is greater than or equal to value at (reference of Array x minus M\_ONE)) is TRUE and perform the following.

* Set snum points to snum points RIGHTSHIFT by M\_ONE when snum points is greater than M\_ONE, otherwise do nothing.
* Do the following when rl temp is less than or equal to ind\_var

1. Set reference of Array x to addition of reference of Array x and snum points
2. Set reference of Array y to addition of reference of Array y and snum points

* Do the following when rl temp is greater than to ind\_var

1. Set reference of Array x to difference of reference of Array x and snum points
2. Set reference of Array y to difference of reference of Array y and snum points

* Set rl temp to Array x.

When the following conditions are satisfied:

* Not of (pi32X of table is equal to M\_HW\_NULL logical OR with pi32y of table is equal to M\_HW\_NULL logical OR with stat is equal to M\_HW\_NULL) .
* rl temp is less than or equal to Slastx
* ind\_var is less than Slastx.
* ind\_var is greater than rl temp.

##### 10.11.3.7.5 dauanalookup-LookupTableLookup32-LLR-005

Requirement ID: H398-LLD-ANA-FNC-1921

The function shall perform the following

* Estatus is set to LS\_OUTSIDE
* Stat is set to Estatus
* return Array y from the function.

When the following conditions are satisfied:

* Not of (pi32X of table is equal to M\_HW\_NULL logical OR with pi32y of table is equal to M\_HW\_NULL logical OR with stat is equal to M\_HW\_NULL).
* rl temp is greater than Slastx
* ind\_var is greater than rl temp.

##### 10.11.3.7.6 dauanalookup-LookupTableLookup32-LLR-006

Requirement ID: H398-LLD-ANA-FNC-1922

The function shall perform

* reference of Array y is set to addition of reference of Array y and (snum points minus M\_ONE).
* Estatus is set to LS\_OUTSIDE when ind\_var is less than Slastx, otherwise do nothing.
* Stat is set to Estatus
* Return Array y from the function.

When

* Not of (pi32X of table is equal to M\_HW\_NULL logical OR with pi32y of table is equal to M\_HW\_NULL logical OR with stat is equal to M\_HW\_NULL) .
* rl temp is greater than Slastx
* ind variable is less than rl temp
* ind\_var is less than or equal to Slastx.

##### 10.11.3.7.7 dauanalookup-LookupTableLookup32-LLR-007

Requirement ID: H398-LLD-ANA-FNC-1923

The function shall loop until (NOT of (ind\_var is greater than rl temp) AND (ind\_var is less than or equal to value at (reference of Array x minus M\_ONE)) is TRUE and perform the following.

* Set snum points to snum points RIGHTSHIFT by M\_ONE when snum points is greater than M\_ONE, otherwise do nothing.
* Do the following when rl temp is greater than or equal to ind\_var
* Set reference of Array x to addition of reference of Array x and snum points
* Set reference of Array y to addition of reference of Array y and snum points.
* Do the following when rl temp is less than ind\_var
* Set reference of Array x to difference of reference of Array x and snum points
* Set reference of Array y to difference of reference of Array y and snum points.
* Set rl temp to Array x.

When the following conditions are satisfied:

* Not of (pi32X of table is equal to M\_HW\_NULL logical OR with pi32y of table is equal to M\_HW\_NULL logical OR with stat is equal to M\_HW\_NULL) .
* rl temp is greater than Slastx
* ind\_var is less than rl temp
* ind\_var is greater than Slastx.

##### 10.11.3.7.8 dauanalookup-LookupTableLookup32-LLR-008

Requirement ID: H398-LLD-ANA-FNC-1924

The function shall performs the following when Not of (pi32X of table is equal to M\_HW\_NULL logical OR with pi32y of table is equal to M\_HW\_NULL logical OR with stat is equal to M\_HW\_NULL) .

* Seg deltax is set to Array x
* Seg deltax is updated with difference of Seg deltax and value at (reference of Array x decremented by one).
* Seg deltay is set to Array y
* Seg deltay is updated with difference of Seg deltay and value at (reference of Array y decremented by one).
* Set stat to estatus.

##### 10.11.3.7.9 dauanalookup-LookupTableLookup32-LLR-009

Requirement ID: H398-LLD-ANA-FNC-1925

The function shall perform the following Not of (pi32X of table is equal to M\_HW\_NULL logical OR with pi32y of table is equal to M\_HW\_NULL logical OR with stat is equal to M\_HW\_NULL) .

* delta\_x is set to difference of ind\_var and Array x.
* results is set to product of delta x and Seg deltay and multiplied by M\_TWO.

##### 10.11.3.7.10 dauanalookup-LookupTableLookup32-LLR-010

Requirement ID: H398-LLD-ANA-FNC-1926

The function shall return M\_ZERO when Seg deltax is equal to M\_ZERO, otherwise do nothing

When below condition is satisfied:

* Not of (pi32X of table is equal to M\_HW\_NULL logical OR with pi32y of table is equal to M\_HW\_NULL logical OR with stat is equal to M\_HW\_NULL) .

##### 10.11.3.7.11 dauanalookup-LookupTableLookup32-LLR-011

Requirement ID: H398-LLD-ANA-FNC-1927

The function shall set results to division of results by Seg deltax

When below condition is satisfied:

* Not of (pi32X of table is equal to M\_HW\_NULL logical OR with pi32y of table is equal to M\_HW\_NULL logical OR with stat is equal to M\_HW\_NULL) .

##### 10.11.3.7.12 dauanalookup-LookupTableLookup32-LLR-012

Requirement ID: H398-LLD-ANA-FNC-1928

The function shall perform the following when results is less than M\_ZERO

1. set Nega to TRUE.
2. Set results to minus results.

Otherwise do nothing.

When below condition is satisfied:

Not of (pi32X of table is equal to M\_HW\_NULL logical OR with pi32y of table is equal to M\_HW\_NULL logical OR with stat is equal to M\_HW\_NULL) .

##### 10.11.3.7.13 dauanalookup-LookupTableLookup32-LLR-013

Requirement ID: H398-LLD-ANA-FNC-1929

The function shall performs the following:

1. results is set to addition of results and M\_ONE.
2. results is updated with results RIGHT SHIFT by M\_ONE.

When below condition is satisfied:

Not of (pi32X of table is equal to M\_HW\_NULL logical OR with pi32y of table is equal to M\_HW\_NULL logical OR with stat is equal to M\_HW\_NULL) .

##### 10.11.3.7.14 dauanalookup-LookupTableLookup32-LLR-014

Requirement ID: H398-LLD-ANA-FNC-1930

The function shall set results to minus results when Nega is equal to TRUE, otherwise do nothing.

When below condition is satisfied:

* Not of (pi32X of table is equal to M\_HW\_NULL logical OR with pi32y of table is equal to M\_HW\_NULL logical OR with stat is equal to M\_HW\_NULL) .

##### 10.11.3.7.15 dauanalookup-LookupTableLookup32-LLR-015

Requirement ID: H398-LLD-ANA-FNC-1931

The function shall set results to sum of results and Array y.

When below condition is satisfied:

Not of (pi32X of table is equal to M\_HW\_NULL logical OR with pi32y of table is equal to M\_HW\_NULL logical OR with stat is equal to M\_HW\_NULL) .

##### 10.11.3.7.16 dauanalookup-LookupTableLookup32-LLR-016

Requirement ID: H398-LLD-ANA-FNC-1932

The function shall return results from the function.

### 10.11.4 LookupTableInverse32

Low Level Design Details about CSU LookupTableInverse32 will follow in the sub sections.

#### 10.11.4.1 Brief Description

The function LookupTableInverse32 swaps the rows and columns of lookup table.

#### 10.11.4.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD(H398-003-012-ANA).

#### 10.11.4.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 10.11.4.4 Parameter list (Input/Output)

Inputs : T\_SENSOR\_TABLE\_32 \*table,

T\_SINT32 dep\_var,

T\_LOOKUP\_STAT \*stat

Outputs : None

Note:Reference to the Sensor table from module configuration data.

#### 10.11.4.5 Return Value

T\_SINT16 - Return the results of the Lookup of the swapped tables.

#### 10.11.4.6 Other CSUs called by this CSU

LookupTableLookup32

#### 10.11.4.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to LookupTableInverse32

##### 10.11.4.7.1 dauanalookup-LookupTableInverse32-LLR-001

Requirement ID: H398-LLD-ANA-FNC-1941

The function shall return M\_ZERO when (pi32X of table is equal to M\_HW\_NULL logical OR pi32Y of table is equal to M\_HW\_NULL logical OR stat is equal to M\_HW\_NULL) is equal to TRUE.

##### 10.11.4.7.2 dauanalookup-LookupTableInverse32-LLR-002

Requirement ID: H398-LLD-ANA-FNC-1942

The function shall do the following when (pi32X of table is equal to M\_HW\_NULL logical OR pi32Y of table is equal to M\_HW\_NULL logical OR stat is equal to M\_HW\_NULL) is equal to FALSE.

* Set pi32X of swap to pi32Y of table.
* Set pi32Y of swap to pi32X of table.

##### 10.11.4.7.3 dauanalookup-LookupTableInverse32-LLR-003

Requirement ID: H398-LLD-ANA-FNC-1943

The function shall return the value returned by LookupTableLookup32 function with parameters reference to swapped lookup table, dep\_var and stat.

## 10.12 dauanamain

This module contains main function to call other routines.

### 10.12.1 MainFunc

Low Level Design Details about CSU MainFunc will follow in the sub sections.

#### 10.12.1.1 Brief Description

The function MainFunc initializes all hardware, kernel, tasks and Start the multitasking process.

#### 10.12.1.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.12.1.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 10.12.1.4 Parameter list (Input/Output)

Inputs : None

Outputs : None

#### 10.12.1.5 Return Value

T\_SINT32 - Always return Zero

#### 10.12.1.6 Other CSUs called by this CSU

HwInit

PbitCheck

OsInit

InitInit

OsStart

#### 10.12.1.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to MainFunc.

##### 10.12.1.7.1 dauanamain-MainFunc-LLR-001

Requirement ID: H398-LLD-ANA-FNC-687

The function shall initialize System and Peripheral Clocks, hardware pins and interrupt controllers by calling HwInit.

##### 10.12.1.7.2 dauanamain-MainFunc-LLR-002

Requirement ID: H398-LLD-ANA-FNC-688

The function shall perform processor test by calling function PbitCheck.

##### 10.12.1.7.3 dauanamain-MainFunc-LLR-003

Requirement ID: H398-LLD-ANA-FNC-689

The function shall initialize the kernel by calling function OsInit.

##### 10.12.1.7.4 dauanamain-MainFunc-LLR-004

Requirement ID: H398-LLD-ANA-FNC-690

The function shall initialize the Init task by calling function InitInit.

##### 10.12.1.7.5 dauanamain-MainFunc-LLR-005

Requirement ID: H398-LLD-ANA-FNC-691

The function shall start the kernel by calling function OsStart.

##### 10.12.1.7.6 dauanamain-MainFunc-LLR-006

Requirement ID: H398-LLD-ANA-FNC-692

The function shall return M\_ZERO.

## 10.13 dauanaoscpu

The dauanaoscpu CSC defines the OS Task creation function.

### 10.13.1 OsTaskCreate

Low Level Design Details about CSU OsTaskCreate will follow in the sub sections.

#### 10.13.1.1 Brief Description

The OsTaskCreate function is used to create the OS task.

#### 10.13.1.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.13.1.3 List of global variables accessed and modified

Accessed : Os\_tcb\_prio\_tbl

Os\_running

U32\_critical\_sr

Modified : U32\_critical\_sr

#### 10.13.1.4 Parameter list (Input/Output)

Inputs : void (\*ptask) (void \*dptr) function pointer to the task

void \* p\_data pointer to task parameters

T\_UINT8 u8\_prio task priority

Outputs : void \*pt\_os pointer to top of stack

#### 10.13.1.5 Return Value

T\_UINT8 - Returns task creation status

M\_OS\_PRIO\_EXIST - Task already exists

M\_OS\_NO\_ERR - Task created successfully

M\_OS\_PRIO\_INVALID - Invalid priority

M\_OS\_NO\_MORE\_TCB - No more free TCB in the TCB list.

#### 10.13.1.6 Other CSUs called by this CSU

SaveStatusReg

RestoreStatusReg

OsTcbInit

OsSched

#### 10.13.1.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to OsTaskCreate.

##### 10.13.1.7.1 dauanaoscpu-OsTaskCreate-LLR-001

Requirement ID: H398-LLD-ANA-FNC-702

The function shall return M\_OS\_PRIO\_INVALID when priority specified (i.e. u8\_prio) is higher than the maximum value M\_OS\_LOWEST\_PRIO allowed.

##### 10.13.1.7.2 dauanaoscpu-OsTaskCreate-LLR-002

Requirement ID: H398-LLD-ANA-FNC-703

The function shall call ‘SaveStatusReg’ (M\_OS\_ENTER\_CRITICAL) return value gets stored in U32\_critical\_sr to disable interrupts before checking when priority specified (i.e. u8\_prio) is not higher than the maximum value M\_OS\_LOWEST\_PRIO allowed.

##### 10.13.1.7.3 dauanaoscpu-OsTaskCreate-LLR-003

Requirement ID: H398-LLD-ANA-FNC-704

The function shall perform the following operations when when priority specified (i.e. u8\_prio) is not higher than the maximum value M\_OS\_LOWEST\_PRIO allowed and the requested task priority u8\_prio is available in the TCB priority table 'Os\_tcb\_prio\_tbl'

* Call ‘RestoreStatusReg’ (M\_OS\_EXIT\_CRITICAL) with parameter U32\_critical\_sr to restore the priority mask register.
* Set pu32\_stk to pt\_os.
* Set (Stack top minus 1) with address of M\_FPCCR register.
* Set (Stack top minus 2) with address of M\_FPSCR register.
* Decrement the stack pointer Stack top further by M\_ESC\_16\_LOCATION.
* Set (Stack top minus 19) with address of M\_XPSR register .
* Set (Stack top minus 20) with the address of task code 'ptask' .
* Set (Stack top minus 21) with link register M\_R14\_LR.
* Decrement the stack pointer Stack top further by M\_ESC\_4\_LOCATION.
* Set (Stack top minus 26) with task parameters 'pdata'.
* Decrement the stack pointer Stack top further by M\_ESC\_24\_LOCATION to save the remaining registers on process stack.
* Call OsTcbInit with the parameters u8\_prio and Reference to process stack to initialize the TCB.
* Call function OsSched to find highest priority task when the call to function OsTcbInit returned M\_OS\_NO\_ERR and flag indicating that kernel is running (Os\_running) is TRUE, do nothing when flag indicating that kernel is running (Os\_running) is FALSE.
* Do nothing when function ‘OsTcbInit’ returns other than M\_OS\_NO\_ERR.
* Return the return variable of function ‘OsTcbInit’.

##### 10.13.1.7.4 dauanaoscpu-OsTaskCreate-LLR-004

Requirement ID: H398-LLD-ANA-FNC-705

The function shall call ‘RestoreStatusReg’ (M\_OS\_EXIT\_CRITICAL) with parameter U32\_critical\_sr to restore the priority mask register and return with M\_OS\_PRIO\_EXIST when the task priority to be created (i.e. Os\_tcb\_prio\_tbl) already exists and the priority specified (i.e. u8\_prio) is not higher than the maximum value M\_OS\_LOWEST\_PRIO allowed.

## 10.14 dauanaoscpua

The dauanaoscpua CSC defines asm routines for uCOS.

### 10.14.1 OsCtxSw

Low Level Design Details about CSU OsCtxSw will follow in the sub sections.

#### 10.14.1.1 Brief Description

The OsCtxSw function performs a task level context switch.

#### 10.14.1.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD(H398-003-012-ANA).

#### 10.14.1.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 10.14.1.4 Parameter list (Input/Output)

Inputs : None

Outputs :None

#### 10.14.1.5 Return Value

None

#### 10.14.1.6 Other CSUs called by this CSU

None

#### 10.14.1.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to OsCtxSw.

##### 10.14.1.7.1 dauanaoscpua-OsCtxSw-LLR-001

Requirement ID: H398-LLD-ANA-FNC-715

The function shall do the following to trigger the PendSV exception

- Load the Interrupt control state register(M\_NVIC\_INT\_CTRL) to R0

- Load the Value to trigger PendSV isr (M\_NVIC\_PENDSVSET) to R1

- Write R1 content to address of R0.

### 10.14.2 OsIntCtxSw

Low Level Design Details about CSU OSIntCtxSw will follow in the sub sections.

#### 10.14.2.1 Brief Description

The OSIntCtxSw function performs a task level context switch same as OsCtxSw.

#### 10.14.2.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD(H398-003-012-ANA).

#### 10.14.2.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 10.14.2.4 Parameter list (Input/Output)

Inputs : None

Outputs : None

#### 10.14.2.5 Return Value

None

#### 10.14.2.6 Other CSUs called by this CSU

None

#### 10.14.2.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to OsIntCtxSw

##### 10.14.2.7.1 dauanaoscpua-OsIntCtxSw-LLR-001

Requirement ID: H398-LLD-ANA-FNC-724

The function shall do the following to trigger the PendSV exception

- Load the Interrupt control state register(M\_NVIC\_INT\_CTRL) to R0

- Load the Value to trigger PendSV isr(M\_NVIC\_PENDSVSET) to R1

- write R1 content to address of R0.

### 10.14.3 SaveStatusReg

Low Level Design Details about CSU SaveStatusReg will follow in the sub sections.

#### 10.14.3.1 Brief Description

The SaveStatusReg function save the interrupt mask register and then disables interrupts to enter critical section.

#### 10.14.3.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD(H398-003-012-ANA).

#### 10.14.3.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 10.14.3.4 Parameter list (Input/Output)

Inputs : None

Outputs : None

#### 10.14.3.5 Return Value

T\_UINT32 return Critical status register value

#### 10.14.3.6 Other CSUs called by this CSU

None

#### 10.14.3.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to SaveStatusReg.

##### 10.14.3.7.1 dauanaoscpua-SaveStatusReg-LLR-001

Requirement ID: H398-LLD-ANA-FNC-733

The function shall do the following to save the interrupt mask register and then disables interrupts

- Read PRIMASK special register value and write it to R0

- Disable IRq by setting the PRIMASK special register value.

### 10.14.4 RestoreStatusReg

Low Level Design Details about CSU RestoreStatusReg will follow in the sub sections.

#### 10.14.4.1 Brief Description

The RestoreStatusReg function restores the interrupt disable mask to its original value.

#### 10.14.4.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD(H398-003-012-ANA).

#### 10.14.4.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 10.14.4.4 Parameter list (Input/Output)

Inputs : T\_UINT32 U32\_critical\_sr - Critical status register value

Outputs : None

#### 10.14.4.5 Return Value

None

#### 10.14.4.6 Other CSUs called by this CSU

None

#### 10.14.4.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to RestoreStatusReg.

##### 10.14.4.7.1 dauanaoscpua-RestoreStatusReg-LLR-001

Requirement ID: H398-LLD-ANA-FNC-742

The function shall write the contents of R0 into PRIMASK register to enable interrupts (Restore saved interrupts).

### 10.14.5 OsStartHighRdy

Low Level Design Details about CSU OsStartHighRdy will follow in the sub sections.

#### 10.14.5.1 Brief Description

The OsStartHighRdy function starts running the highest priority task.

#### 10.14.5.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD(H398-003-012-ANA).

#### 10.14.5.3 List of global variables accessed and modified

Accessed : Os\_running

Modified : None

#### 10.14.5.4 Parameter list (Input/Output)

Inputs : None

Outputs : None

#### 10.14.5.5 Return Value

None

#### 10.14.5.6 Other CSUs called by this CSU

None

#### 10.14.5.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to OsStartHighRdy.

##### 10.14.5.7.1 dauanaoscpua-OsStartHighRdy-LLR-001

Requirement ID: H398-LLD-ANA-FNC-751

The function shall do the following to set the PendSV exception priority

- Load the System prio register(M\_NVIC\_SYSPRI14) to R0

- Load the PendSV priority value(M\_NVIC\_PENDSV\_PRI) to R1

- Write R1 content as unsigned byte to address of R0.

##### 10.14.5.7.2 dauanaoscpua-OsStartHighRdy-LLR-002

Requirement ID: H398-LLD-ANA-FNC-752

The function shall do the following to Set the PSP to M\_DEFAULT\_VAL for initial context switch call

- Write value of M\_DEFAULT\_VAL to R0

- Writes the contents of R0 into PSP register.

##### 10.14.5.7.3 dauanaoscpua-OsStartHighRdy-LLR-003

Requirement ID: H398-LLD-ANA-FNC-753

The function shall do the following to Set the Os\_running flag to TRUE (indicate that multitasking will start)

- Load the Os\_running flag address to R0

- Write value of M\_DEFAULT\_TRUE\_VAL to R1

- Write R1 content as unsigned byte to address of R0.

##### 10.14.5.7.4 dauanaoscpua-OsStartHighRdy-LLR-004

Requirement ID: H398-LLD-ANA-FNC-754

The function shall do the following to trigger the PendSV exception

- Load the Interrupt control state register(M\_NVIC\_INT\_CTRL) to R0

- Load the Value to trigger PendSV isr(M\_NVIC\_PENDSVSET) to R1

- Write R1 content as unsigned byte to address of R0.

##### 10.14.5.7.5 dauanaoscpua-OsStartHighRdy-LLR-005

Requirement ID: H398-LLD-ANA-FNC-755

The function shall do the following to enable interrupts

- Enable Interrupt request by clearing the PRIMASK special register value.

##### 10.14.5.7.6 dauanaoscpua-OsStartHighRdy-LLR-006

Requirement ID: H398-LLD-ANA-FNC-756

The function shall enter infinite wait state when highest priority task could not be scheduled.

### 10.14.6 PendSvHandler

Low Level Design Details about CSU PendSvHandler will follow in the sub sections.

#### 10.14.6.1 Brief Description

The PendSvHandler function handles all context switching for uCOS.

#### 10.14.6.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD(H398-003-012-ANA).

#### 10.14.6.3 List of global variables accessed and modified

Accessed : Os\_tcb\_cur

Os\_tcb\_high\_rdy

Modified : None

#### 10.14.6.4 Parameter list (Input/Output)

Inputs : None

Outputs : None

#### 10.14.6.5 Return Value

None

#### 10.14.6.6 Other CSUs called by this CSU

None

#### 10.14.6.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to PendSvHandler.

##### 10.14.6.7.1 dauanaoscpua-PendSvHandler-LLR-001

Requirement ID: H398-LLD-ANA-FNC-765

The function shall do the following to Prevent interruption during context switch

- Disable Interrupt request by setting the I-bit in the CPSR.

##### 10.14.6.7.2 dauanaoscpua-PendSvHandler-LLR-002

Requirement ID: H398-LLD-ANA-FNC-766

The function shall do the following to save the process stack pointer (PSP)

- Read PSP register value and write it to R0.

##### 10.14.6.7.3 dauanaoscpua-PendSvHandler-LLR-003

Requirement ID: H398-LLD-ANA-FNC-767

The function shall do the following to skip register save the first time

- Forward branch to 'PendSvHandler\_NoSave' when R0 is zero.

##### 10.14.6.7.4 dauanaoscpua-PendSvHandler-LLR-004

Requirement ID: H398-LLD-ANA-FNC-768

The function shall do the following to save remaining register R4-R11 on process stack

- Decrement R0 Before each access and load the register R4-R11 to the Decremented address.

##### 10.14.6.7.5 dauanaoscpua-PendSvHandler-LLR-005

Requirement ID: H398-LLD-ANA-FNC-769

The function shall do the following to save remaining floating point register S16-S31

- Decrement R0 Before each access and store the register S16-S31 to the Decremented address.

##### 10.14.6.7.6 dauanaoscpua-PendSvHandler-LLR-006

Requirement ID: H398-LLD-ANA-FNC-770

The function shall do the following to load the Stack pointer to the TCB stack

- Load the current TCB address(Os\_tcb\_cur) to R1

- Load the address content in R1 to R1

- write R0 content to address of R1

##### 10.14.6.7.7 dauanaoscpua-PendSvHandler-LLR-007

Requirement ID: H398-LLD-ANA-FNC-771

The function shall do the following to set highest priority task ready to execute into the current TCB (i.e., Os\_tcb\_cur equal to Os\_tcb\_high\_rdy) when branch PendSvHandler\_NoSave is invoked

- Load the current TCB(Os\_tcb\_cur) address to R0

- Load the address of highest priority task ready to execute (Os\_tcb\_high\_rdy) to R1

- Load the content of address in R1 to R2

- write R2 content to address of R0

- Load the address content in R2 to R0

(i.e. - Set Stack Pointer (R0) to os\_tcb\_stkptr value of Os\_tcb\_high\_rdy)

- Increment R0 After each access and restore the register S16- S31

from the Incremented address

- Increment R0 After each access and load the register R4 - R11 from

the Incremented address

- Load PSP with the new stack pointer (R0)

- Load the link register to link register bitwise OR M\_EXCEPTION\_RETURN

- Enable Interrupt request by clearing the PRIMASK special register value.

## 10.15 dauanapbit

This module contains the routine to Implement the PBIT.

### 10.15.1 RamTest

Low Level Design Details about CSU RamTest will follow in the sub sections.

#### 10.15.1.1 Brief Description

This function does the Internal RAM test for the selected location by writing some set of data to each address in the memory device and verify the data by reading it back. When all the values read back are the same as those that were written, then the memory device is said to pass the test.

#### 10.15.1.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.15.1.3 List of global variables accessed and modified

Accessed : \_ram\_test\_start

\_ram\_test\_end

Modified : None

#### 10.15.1.4 Parameter list (Input/Output)

Inputs : None

Outputs : None

#### 10.15.1.5 Return Value

None

#### 10.15.1.6 Other CSUs called by this CSU

BitErrorHandler

#### 10.15.1.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to RamTest.

##### 10.15.1.7.1 dauanapbit-RamTest-LLR-001

Requirement ID: H398-LLD-ANA-FNC-781

The function shall do the following for all the location from start of ram test location (\_ram\_test\_start) to end of ram test location (\_ram\_test\_end) in InternalRAM:

a. Set the value M\_CHECK\_5A5A to the reference of Internal RAM location.

b. Call the function BitErrorHandler with parameter (RAM\_ERR) when the value at the reference of Internal RAM location is not equal to M\_CHECK\_5A5A.

c. Do the following when the value at the reference of Internal RAM location is equal to M\_CHECK\_5A5A:

- Set the value M\_CHECK\_A5A5 to the reference of Internal RAM location.

- Call the function BitErrorHandler with parameter (RAM\_ERR) when the value at the reference

of Internal RAM location is not M\_CHECK\_A5A5 otherwise do nothing.

### 10.15.2 Cputest

Low Level Design Details about CSU Cputest will follow in the sub sections.

#### 10.15.2.1 Brief Description

This function performs the CPU test by performing the Arithmetic and logical operation check.

#### 10.15.2.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.15.2.3 List of global variables accessed and modified

Accessed : Cpu\_test\_res

Modified : Cpu\_test\_res

#### 10.15.2.4 Parameter list (Input/Output)

Inputs : None

Outputs : None

#### 10.15.2.5 Return Value

None

#### 10.15.2.6 Other CSUs called by this CSU

AluTest,

BitErrorHandler

#### 10.15.2.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to Cputest.

##### 10.15.2.7.1 dauanapbit-CpuTest-LLR-001

Requirement ID: H398-LLD-ANA-FNC-790

The function shall set the global variable Cpu\_test\_res to FAILED.

##### 10.15.2.7.2 dauanapbit-CpuTest-LLR-002

Requirement ID: H398-LLD-ANA-FNC-791

The function shall call the function Alutest to perform the Arithmetic and logical operation check.

##### 10.15.2.7.3 dauanapbit-CpuTest-LLR-003

Requirement ID: H398-LLD-ANA-FNC-792

The function shall call the function BitErrorHandler with parameter (CPU\_ERR)

when the global variable Cpu\_test\_res is FAILED otherwise do nothing.

### 10.15.3 Mcdcheck

Low Level Design Details about CSU Mcdcheck will follow in the sub sections.

#### 10.15.3.1 Brief Description

This function does the validity check for the MCD entries.

#### 10.15.3.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.15.3.3 List of global variables accessed and modified

Accessed : Ptr\_sensor\_aas\_analog

Modified : None

#### 10.15.3.4 Parameter list (Input/Output)

Inputs : None

Outputs : None

#### 10.15.3.5 Return Value

None

#### 10.15.3.6 Other CSUs called by this CSU

BitErrorHandler

#### 10.15.3.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to Mcdcheck.

##### 10.15.3.7.1 dauanapbit-McdCheck-LLR-001

Requirement ID: H398-LLD-ANA-FNC-801

The function shall perform as mentioned below by looping through XADC\_CHAN\_1 to XADC\_CHANS-1 and XADC\_MUX\_1 to XADC\_MUX\_CHANS-1:

-call the function BitErrorHandler with parameter(MCD\_ERR) when External ADC channel state (i.e u8\_Chan\_State of (address of Sensor\_aas\_analog with indices channel and Mux of Ptr\_sensor\_aas\_analog)) is set to other than ENABLE AND DISABLE.

##### 10.15.3.7.2 dauanapbit-McdCheck-LLR-002

Requirement ID: H398-LLD-ANA-FNC-802

The function shall perform as mentioned below by looping through XADC\_CHAN\_1 to XADC\_CHANS-1 and XADC\_MUX\_1 to XADC\_MUX\_CHANS-1:

-do nothing when External ADC channel state (i.e u8\_Chan\_State of (address of Sensor\_aas\_analog with indices channel and Mux of Ptr\_sensor\_aas\_analog)) is set to DISABLE.

##### 10.15.3.7.3 dauanapbit-McdCheck-LLR-003

Requirement ID: H398-LLD-ANA-FNC-803

The function shall perform as mentioned below by looping through XADC\_CHAN\_1 to XADC\_CHANS-1 and XADC\_MUX\_1 to XADC\_MUX\_CHANS-1:

-call the function BitErrorHandler with parameter(MCD\_ERR) External ADC channel state (i.e u8\_Chan\_State of address of (Sensor\_aas\_analog with indices channel and Mux of Ptr\_sensor\_aas\_analog)) is set to ENABLE AND when External ADC default state (i.e u8\_Default\_State of (address of Sensor\_aas\_analog with indices channel and Mux of Ptr\_sensor\_aas\_analog) )is set to other than ENABLE AND DISABLE.

##### 10.15.3.7.4 dauanapbit-McdCheck-LLR-004

Requirement ID: H398-LLD-ANA-FNC-804

The function shall perform as mentioned below by looping through XADC\_CHAN\_1 to XADC\_CHANS-1 and XADC\_MUX\_1 to XADC\_MUX\_CHANS-1:

-call the function BitErrorHandler with parameter(MCD\_ERR) when External ADC default state (i.e u8\_Default\_State of Sensor\_aas\_analog with indices channel and Mux of Ptr\_sensor\_aas\_analog) is set to ENABLE OR DISABLE AND when External ADC default value (i.e i16\_Default\_Val of (address of Sensor\_aas\_analog with indices channel and Mux of Ptr\_sensor\_aas\_analog)) is not equal to M\_ZERO.

##### 10.15.3.7.5 dauanapbit-McdCheck-LLR-005

Requirement ID: H398-LLD-ANA-FNC-805

The function shall perform as mentioned below by looping through XADC\_CHAN\_1 to XADC\_CHANS-1 and XADC\_MUX\_1 to XADC\_MUX\_CHANS-1:

-call the function BitErrorHandler with parameter(MCD\_ERR) when External ADC default value (i.e i16\_Default\_Val of (address of Sensor\_aas\_analog with indices channel and Mux of Ptr\_sensor\_aas\_analog)) is equal to M\_ZERO AND when External ADC calibrate state (i.e b\_Calibrate of (address of Sensor\_aas\_analog with indices channel and Mux of Ptr\_sensor\_aas\_analog)) is set to other than TRUE AND FALSE.

##### 10.15.3.7.6 dauanapbit-McdCheck-LLR-006

Requirement ID: H398-LLD-ANA-FNC-806

The function shall perform as mentioned below by looping through XADC\_CHAN\_1 to XADC\_CHANS-1 and XADC\_MUX\_1 to XADC\_MUX\_CHANS-1:

-call the function BitErrorHandler with parameter(MCD\_ERR) when External ADC calibrate state (i.e b\_Calibrate of Sensor\_aas\_analog with indices channel and Mux of Ptr\_sensor\_aas\_analog) is set to TRUE OR FALSE AND when External ADC low calibration point (i.e i16\_Low\_Cal\_Point of (address of Sensor\_aas\_analog with indices channel and Mux of Ptr\_sensor\_aas\_analog)) is less than M\_LOW\_CAL\_POINT\_MIN OR when External ADC low calibration point (i.e i16\_Low\_Cal\_Point of (address of Sensor\_aas\_analog with indices channel and Mux of Ptr\_sensor\_aas\_analog)) is greater than M\_LOW\_CAL\_POINT\_MAX.

##### 10.15.3.7.7 dauanapbit-McdCheck-LLR-007

Requirement ID: H398-LLD-ANA-FNC-807

The function shall perform as mentioned below by looping through XADC\_CHAN\_1 to XADC\_CHANS-1 and XADC\_MUX\_1 to XADC\_MUX\_CHANS-1:

-call the function BitErrorHandler with parameter(MCD\_ERR) when External ADC low calibration point (i.e i16\_Low\_Cal\_Point of (address of Sensor\_aas\_analog with indices channel and Mux of Ptr\_sensor\_aas\_analog)) is greater than or equal to M\_LOW\_CAL\_POINT\_MIN AND when External ADC low calibration point (i.e i16\_Low\_Cal\_Point of (address of Sensor\_aas\_analog with indices channel and Mux of Ptr\_sensor\_aas\_analog)) is less than or equal to M\_LOW\_CAL\_POINT\_MAX AND when External ADC high calibration point (i.e i16\_High\_Cal\_Point of (address of Sensor\_aas\_analog with indices channel and Mux of Ptr\_sensor\_aas\_analog)) is less than M\_HIGH\_CAL\_POINT\_MIN OR when External ADC high calibration point (i.e i16\_High\_Cal\_Point of (address of Sensor\_aas\_analog with indices channel and Mux of Ptr\_sensor\_aas\_analog)) is greater than M\_HIGH\_CAL\_POINT\_MAX.

##### 10.15.3.7.8 dauanapbit-McdCheck-LLR-008

Requirement ID: H398-LLD-ANA-FNC-808

The function shall perform as mentioned below by looping through XADC\_CHAN\_1 to XADC\_CHANS-1 and XADC\_MUX\_1 to XADC\_MUX\_CHANS-1:

-call the function BitErrorHandler with parameter(MCD\_ERR) when External ADC high calibration point (i.e i16\_High\_Cal\_Point of (address of Sensor\_aas\_analog with indices channel and Mux of Ptr\_sensor\_aas\_analog)) is greater than or equal to M\_HIGH\_CAL\_POINT\_MIN AND when External ADC high calibration point (i.e i16\_High\_Cal\_Point of (address of Sensor\_aas\_analog with indices channel and Mux of Ptr\_sensor\_aas\_analog)) is less than or equal to M\_HIGH\_CAL\_POINT\_MAX AND when External ADC input type (i.e c\_Type of (address of Sensor\_aas\_analog with indices channel and Mux of Ptr\_sensor\_aas\_analog)) is less than CAL\_DIS\_MV OR when External ADC input type (i.e c\_Type of (address of Sensor\_aas\_analog with indices channel and Mux of Ptr\_sensor\_aas\_analog)) is greater than CAL\_DIS\_OHM.

##### 10.15.3.7.9 dauanapbit-McdCheck-LLR-009

Requirement ID: H398-LLD-ANA-FNC-809

The function shall perform as mentioned below by looping through XADC\_CHAN\_1 to XADC\_CHANS-1 and XADC\_MUX\_1 to XADC\_MUX\_CHANS-1:

-call the function BitErrorHandler with parameter(MCD\_ERR) when External ADC input type (i.e c\_Type of (address of Sensor\_aas\_analog with indices channel and Mux of Ptr\_sensor\_aas\_analog)) is greater than or equal to CAL\_DIS\_MV AND when External ADC input type (i.e c\_Type of (address of Sensor\_aas\_analog with indices channel and Mux of Ptr\_sensor\_aas\_analog)) is less than or equal to CAL\_DIS\_OHM AND when External ADC low calibration display value (i.e i16\_Low\_Cal\_Disp of (address of Sensor\_aas\_analog with indices channel and Mux of Ptr\_sensor\_aas\_analog)) is less than M\_LOW\_CAL\_DISP\_MIN OR when External ADC low calibration display value (i.e i16\_Low\_Cal\_Disp of (address of Sensor\_aas\_analog with indices channel and Mux of Ptr\_sensor\_aas\_analog)) is greater than M\_LOW\_CAL\_DISP\_MAX.

##### 10.15.3.7.10 dauanapbit-McdCheck-LLR-010

Requirement ID: H398-LLD-ANA-FNC-810

The function shall perform as mentioned below by looping through XADC\_CHAN\_1 to XADC\_CHANS-1 and XADC\_MUX\_1 to XADC\_MUX\_CHANS-1:

-call the function BitErrorHandler with parameter(MCD\_ERR) when External ADC low calibration display value (i.e i16\_Low\_Cal\_Disp of (address of Sensor\_aas\_analog with indices channel and Mux of Ptr\_sensor\_aas\_analog)) is greater than or equal to M\_LOW\_CAL\_DISP\_MIN AND when External ADC low calibration display value (i.e i16\_Low\_Cal\_Disp of (address of Sensor\_aas\_analog with indices channel and Mux of Ptr\_sensor\_aas\_analog) is less than or equal to M\_LOW\_CAL\_DISP\_MAX AND when External ADC high calibration display value (i.e i16\_High\_Cal\_Disp of (address of Sensor\_aas\_analog with indices channel and Mux of Ptr\_sensor\_aas\_analog)) is less than M\_HIGH\_CAL\_DISP\_MIN OR when External ADC high calibration display value (i.e i16\_High\_Cal\_Disp of (address of Sensor\_aas\_analog with indices channel and Mux of Ptr\_sensor\_aas\_analog)) is greater than M\_HIGH\_CAL\_DISP\_MAX.

##### 10.15.3.7.11 dauanapbit-McdCheck-LLR-011

Requirement ID: H398-LLD-ANA-FNC-811

The function shall perform as mentioned below by looping through XADC\_CHAN\_1 to XADC\_CHANS-1 and XADC\_MUX\_1 to XADC\_MUX\_CHANS-1:

-call the function BitErrorHandler with parameter(MCD\_ERR) when External ADC high calibration display value (i.e i16\_High\_Cal\_Disp of (address of Sensor\_aas\_analog with indices channel and Mux of Ptr\_sensor\_aas\_analog)) is greater than or equal to M\_HIGH\_CAL\_DISP\_MIN AND when External ADC high calibration display value (i.e i16\_High\_Cal\_Disp of (address of Sensor\_aas\_analog with indices channel and Mux of Ptr\_sensor\_aas\_analog)) is less than or equal to M\_HIGH\_CAL\_DISP\_MAX AND when External ADC resolution (i.e i16\_Resolution of (address of Sensor\_aas\_analog with indices channel and Mux of Ptr\_sensor\_aas\_analog)) is less than M\_RESOLUTION\_MIN OR when External ADC resolution (i.e i16\_Resolution of (address of Sensor\_aas\_analog with indices channel and Mux of Ptr\_sensor\_aas\_analog)) is greater than M\_RESOLUTION\_MAX.

##### 10.15.3.7.12 dauanapbit-McdCheck-LLR-012

Requirement ID: H398-LLD-ANA-FNC-812

The function shall perform as mentioned below by looping through XADC\_CHAN\_1 to XADC\_CHANS-1 and XADC\_MUX\_1 to XADC\_MUX\_CHANS-1:

-call the function BitErrorHandler with parameter(MCD\_ERR) when External ADC resolution (i.e i16\_Resolution of (address of Sensor\_aas\_analog with indices channel and Mux of Ptr\_sensor\_aas\_analog)) is greater than or equal to M\_RESOLUTION\_MIN AND when External ADC resolution (i.e i16\_Resolution of (address of Sensor\_aas\_analog with indices channel and Mux of Ptr\_sensor\_aas\_analog)) is less than or equal to M\_RESOLUTION\_MAX AND when External ADC excitation state (i.e b\_excite\_req of (address of Sensor\_aas\_analog with indices channel and Mux of Ptr\_sensor\_aas\_analog)) is set to value other than TRUE AND FALSE.

##### 10.15.3.7.13 dauanapbit-McdCheck-LLR-013

Requirement ID: H398-LLD-ANA-FNC-813

The function shall perform as mentioned below by looping through XADC\_CHAN\_1 to XADC\_CHANS-1 and XADC\_MUX\_1 to XADC\_MUX\_CHANS-1:

-call the function BitErrorHandler with parameter(MCD\_ERR) when External ADC excitation state (i.e b\_excite\_req of (address of Sensor\_aas\_analog with indices channel and Mux of Ptr\_sensor\_aas\_analog)) is set to TRUE OR FALSE AND when External ADC sensor table x value (i.e pi16X of s\_Table of (address of Sensor\_aas\_analog with indices channel and Mux of Ptr\_sensor\_aas\_analog)) is M\_NULL OR when External ADC sensor table y value (i.e pi16Y of s\_Table of (address of Sensor\_aas\_analog with indices channel and Mux of Ptr\_sensor\_aas\_analog)) is M\_NULL.

##### 10.15.3.7.14 dauanapbit-McdCheck-LLR-014

Requirement ID: H398-LLD-ANA-FNC-814

The function shall perform as mentioned below by looping through XADC\_CHAN\_1 to XADC\_CHANS-1 and XADC\_MUX\_1 to XADC\_MUX\_CHANS-1:

-call the function BitErrorHandler with parameter(MCD\_ERR) when External ADC sensor table x value (i.e pi16X of s\_Table of (address of Sensor\_aas\_analog with indices channel and Mux of Ptr\_sensor\_aas\_analog)) is not equal to M\_NULL AND when External ADC sensor table y value (i.e pi16Y of s\_Table of (address of Sensor\_aas\_analog with indices channel and Mux of Ptr\_sensor\_aas\_analog)) is not equal to M\_NULL AND when External ADC minimum range (i.e i16\_Min\_Range of (address of Sensor\_aas\_analog with indices channel and Mux of Ptr\_sensor\_aas\_analog)) is less than M\_MIN\_RANGE\_MIN OR when External ADC minimum range (i.e i16\_Min\_Range of (address of Sensor\_aas\_analog with indices channel and Mux of Ptr\_sensor\_aas\_analog)) is greater than M\_MIN\_RANGE\_MAX.

##### 10.15.3.7.15 dauanapbit-McdCheck-LLR-015

Requirement ID: H398-LLD-ANA-FNC-815

The function shall perform as mentioned below by looping through XADC\_CHAN\_1 to XADC\_CHANS-1 and XADC\_MUX\_1 to XADC\_MUX\_CHANS-1:

-call the function BitErrorHandler with parameter(MCD\_ERR) when External ADC minimum range (i.e i16\_Min\_Range of (address of Sensor\_aas\_analog with indices channel and Mux of Ptr\_sensor\_aas\_analog)) is greater than or equal to M\_MIN\_RANGE\_MIN AND when External ADC minimum range (i.e i16\_Min\_Range of (address of Sensor\_aas\_analog with indices channel and Mux of Ptr\_sensor\_aas\_analog)) is less than or equal to M\_MIN\_RANGE\_MAX AND when any entry for the External ADC maximum range (i.e i16\_Max\_Range of (address of Sensor\_aas\_analog with indices channel and Mux of Ptr\_sensor\_aas\_analog)) is less than M\_MAX\_RANGE\_MIN OR when any entry for the External ADC maximum range (i.e i16\_Max\_Range of (address of Sensor\_aas\_analog with indices channel and Mux of Ptr\_sensor\_aas\_analog)) is greater than M\_MAX\_RANGE\_MAX, otherwise do nothing.

##### 10.15.3.7.16 dauanapbit-McdCheck-LLR-016

Requirement ID: H398-LLD-ANA-FNC-816

The function shall perform as mentioned below by looping through IADC\_CHAN\_1 to IADC\_CHANS-1:

-call the function BitErrorHandler with parameter(MCD\_ERR) when Internal ADC channel state (i.e u8\_Chan\_State of Sensor\_as\_coldjunction with index as channel of Ptr\_sensor\_aas\_analog) is set to value other than ENABLE AND DISABLE.

##### 10.15.3.7.17 dauanapbit-McdCheck-LLR-017

Requirement ID: H398-LLD-ANA-FNC-1863

The function shall perform nothing by looping through IADC\_CHAN\_1 to IADC\_CHANS-1, when Internal ADC channel state (i.e u8\_Chan\_State of Sensor\_as\_coldjunction with index as channel of Ptr\_sensor\_aas\_analog) is equal to DISABLE.

##### 10.15.3.7.18 dauanapbit-McdCheck-LLR-018

Requirement ID: H398-LLD-ANA-FNC-1864

The function shall perform as mentioned below by looping through IADC\_CHAN\_1 to IADC\_CHANS-1:

-call the function BitErrorHandler with parameter(MCD\_ERR) when Internal ADC channel state (i.e u8\_Chan\_State of Sensor\_as\_coldjunction with index as channel of Ptr\_sensor\_aas\_analog) is equal to ENABLE AND when Internal ADC default state (i.e u8\_Default\_State of Sensor\_as\_coldjunction with index as channel of Ptr\_sensor\_aas\_analog) is set to value other than ENABLE AND DISABLE.

##### 10.15.3.7.19 dauanapbit-McdCheck-LLR-019

Requirement ID: H398-LLD-ANA-FNC-1865

The function shall perform as mentioned below by looping through IADC\_CHAN\_1 to IADC\_CHANS-1:

-call the function BitErrorHandler with parameter(MCD\_ERR) when Internal ADC default state (i.e u8\_Default\_State of Sensor\_as\_coldjunction with index as channel of Ptr\_sensor\_aas\_analog) is set to ENABLE OR DISABLE AND when Internal ADC default value (i.e i16\_Default\_Val of Sensor\_as\_coldjunction with index as channel of Ptr\_sensor\_aas\_analog) is not equal to M\_ZERO.

##### 10.15.3.7.20 dauanapbit-McdCheck-LLR-020

Requirement ID: H398-LLD-ANA-FNC-1866

The function shall perform as mentioned below by looping through IADC\_CHAN\_1 to IADC\_CHANS-1:

-call the function BitErrorHandler with parameter(MCD\_ERR) when Internal ADC default value (i.e i16\_Default\_Val of Sensor\_as\_coldjunction with index as channel of Ptr\_sensor\_aas\_analog) is equal to M\_ZERO AND when Internal ADC calibrate state (i.e b\_Calibrate of Sensor\_as\_coldjunction with index as channel of Ptr\_sensor\_aas\_analog) is set to value other than FALSE and TRUE.

##### 10.15.3.7.21 dauanapbit-McdCheck-LLR-021

Requirement ID: H398-LLD-ANA-FNC-1867

The function shall perform as mentioned below by looping through IADC\_CHAN\_1 to IADC\_CHANS-1:

-call the function BitErrorHandler with parameter(MCD\_ERR) when Internal ADC calibrate state (i.e b\_Calibrate of Sensor\_as\_coldjunction with index as channel of Ptr\_sensor\_aas\_analog) is set to FALSE OR TRUE AND when Internal ADC low calibration value (i.e i16\_Low\_Cal\_Point of Sensor\_as\_coldjunction with index as channel of Ptr\_sensor\_aas\_analog) is less than M\_LOW\_CAL\_POINT\_MIN OR when Internal ADC low calibration value (i.e i16\_Low\_Cal\_Point of Sensor\_as\_coldjunction with index as channel of Ptr\_sensor\_aas\_analog) is greater than M\_LOW\_CAL\_POINT\_MAX.

##### 10.15.3.7.22 dauanapbit-McdCheck-LLR-022

Requirement ID: H398-LLD-ANA-FNC-1868

The function shall perform as mentioned below by looping through IADC\_CHAN\_1 to IADC\_CHANS-1:

-call the function BitErrorHandler with parameter(MCD\_ERR) when Internal ADC low calibration value (i.e i16\_Low\_Cal\_Point of Sensor\_as\_coldjunction with index as channel of Ptr\_sensor\_aas\_analog) is greater than or equal to M\_LOW\_CAL\_POINT\_MIN AND when Internal ADC low calibration value (i.e i16\_Low\_Cal\_Point of Sensor\_as\_coldjunction with index as channel of Ptr\_sensor\_aas\_analog) is less than or equal to M\_LOW\_CAL\_POINT\_MAX AND when Internal ADC high calibration value (i.e i16\_High\_Cal\_Point of Sensor\_as\_coldjunction with index as channel of Ptr\_sensor\_aas\_analog) is less than M\_HIGH\_CAL\_POINT\_MIN OR when Internal ADC high calibration value (i.e i16\_High\_Cal\_Point of Sensor\_as\_coldjunction with index as channel of Ptr\_sensor\_aas\_analog) is greater than M\_HIGH\_CAL\_POINT\_MAX.

##### 10.15.3.7.23 dauanapbit-McdCheck-LLR-023

Requirement ID: H398-LLD-ANA-FNC-1869

The function shall perform as mentioned below by looping through IADC\_CHAN\_1 to IADC\_CHANS-1:

-call the function BitErrorHandler with parameter(MCD\_ERR) when Internal ADC high calibration value (i.e i16\_High\_Cal\_Point of Sensor\_as\_coldjunction with index as channel of Ptr\_sensor\_aas\_analog) is greater than or equal to M\_HIGH\_CAL\_POINT\_MIN AND when Internal ADC high calibration value (i.e i16\_High\_Cal\_Point of Sensor\_as\_coldjunction with index as channel of Ptr\_sensor\_aas\_analog) is less than or equal to M\_HIGH\_CAL\_POINT\_MAX AND when Internal ADC input type (i.e c\_Type of Sensor\_as\_coldjunction with index as channel of Ptr\_sensor\_aas\_analog) is less than CAL\_DIS\_MV OR when Internal ADC input type (i.e c\_Type of Sensor\_as\_coldjunction with index as channel of Ptr\_sensor\_aas\_analog) is greater than CAL\_DIS\_OHM.

##### 10.15.3.7.24 dauanapbit-McdCheck-LLR-024

Requirement ID: H398-LLD-ANA-FNC-1870

The function shall perform as mentioned below by looping through IADC\_CHAN\_1 to IADC\_CHANS-1:

-call the function BitErrorHandler with parameter(MCD\_ERR) when Internal ADC input type (i.e c\_Type of Sensor\_as\_coldjunction with index as channel of Ptr\_sensor\_aas\_analog) is greater than or equal to CAL\_DIS\_MV AND when Internal ADC input type (i.e c\_Type of Sensor\_as\_coldjunction with index as channel of Ptr\_sensor\_aas\_analog) is less than or equal to CAL\_DIS\_OHM AND when Internal ADC low calibration display value (i.e i16\_Low\_Cal\_Disp of Sensor\_as\_coldjunction with index as channel of Ptr\_sensor\_aas\_analog) is less than M\_LOW\_CAL\_DISP\_MIN OR when Internal ADC low calibration display value (i.e i16\_Low\_Cal\_Disp of Sensor\_as\_coldjunction with index as channel of Ptr\_sensor\_aas\_analog) is greater than M\_LOW\_CAL\_DISP\_MAX.

##### 10.15.3.7.25 dauanapbit-McdCheck-LLR-025

Requirement ID: H398-LLD-ANA-FNC-1871

The function shall perform as mentioned below by looping through IADC\_CHAN\_1 to IADC\_CHANS-1:

-call the function BitErrorHandler with parameter(MCD\_ERR) when Internal ADC low calibration display value (i.e i16\_Low\_Cal\_Disp of Sensor\_as\_coldjunction with index as channel of Ptr\_sensor\_aas\_analog) is gretaer than or equal to M\_LOW\_CAL\_DISP\_MIN AND when Internal ADC low calibration display value (i.e i16\_Low\_Cal\_Disp of Sensor\_as\_coldjunction with index as channel of Ptr\_sensor\_aas\_analog) is less than or equal to M\_LOW\_CAL\_DISP\_MAX AND when Internal ADC high calibration display value (i.e i16\_High\_Cal\_Disp of Sensor\_as\_coldjunction with index as channel of Ptr\_sensor\_aas\_analog) is less than M\_HIGH\_CAL\_DISP\_MIN OR when Internal ADC high calibration display value (i.e i16\_High\_Cal\_Disp of Sensor\_as\_coldjunction with index as channel of Ptr\_sensor\_aas\_analog) is greater than M\_HIGH\_CAL\_DISP\_MAX.

##### 10.15.3.7.26 dauanapbit-McdCheck-LLR-026

Requirement ID: H398-LLD-ANA-FNC-1872

The function shall perform as mentioned below by looping through IADC\_CHAN\_1 to IADC\_CHANS-1:

-call the function BitErrorHandler with parameter(MCD\_ERR) when Internal ADC high calibration display value (i.e i16\_High\_Cal\_Disp of Sensor\_as\_coldjunction with index as channel of Ptr\_sensor\_aas\_analog) is greater than or equal to M\_HIGH\_CAL\_DISP\_MIN AND when Internal ADC high calibration display value (i.e i16\_High\_Cal\_Disp of Sensor\_as\_coldjunction with index as channel of Ptr\_sensor\_aas\_analog) is less than or equal to M\_HIGH\_CAL\_DISP\_MAX AND when Internal ADC resolution (i.e i16\_Resolution of Sensor\_as\_coldjunction with index as channel of Ptr\_sensor\_aas\_analog) is less than M\_RESOLUTION\_MIN OR when Internal ADC resolution (i.e i16\_Resolution of Sensor\_as\_coldjunction with index as channel of Ptr\_sensor\_aas\_analog) is greater than M\_RESOLUTION\_MAX.

##### 10.15.3.7.27 dauanapbit-McdCheck-LLR-027

Requirement ID: H398-LLD-ANA-FNC-1873

The function shall perform as mentioned below by looping through IADC\_CHAN\_1 to IADC\_CHANS-1:

-call the function BitErrorHandler with parameter(MCD\_ERR) when Internal ADC resolution (i.e i16\_Resolution of Sensor\_as\_coldjunction with index as channel of Ptr\_sensor\_aas\_analog) is greater than or equal to M\_RESOLUTION\_MIN AND when Internal ADC resolution (i.e i16\_Resolution of Sensor\_as\_coldjunction with index as channel of Ptr\_sensor\_aas\_analog) is less than or equal to M\_RESOLUTION\_MAX AND when Internal ADC excitaion state (i.e b\_excite\_req of Sensor\_as\_coldjunction with index as channel of Ptr\_sensor\_aas\_analog) is set to value other than TRUE AND FALSE.

##### 10.15.3.7.28 dauanapbit-McdCheck-LLR-028

Requirement ID: H398-LLD-ANA-FNC-1874

The function shall perform as mentioned below by looping through IADC\_CHAN\_1 to IADC\_CHANS-1:

-call the function BitErrorHandler with parameter(MCD\_ERR) when Internal ADC excitaion state (i.e b\_excite\_req of Sensor\_as\_coldjunction with index as channel of Ptr\_sensor\_aas\_analog) is set to TRUE OR FALSE AND when Internal ADC sensor table x value (i.e pi16X of s\_Table of Sensor\_as\_coldjunction with index as channel of Ptr\_sensor\_aas\_analog) is equal to M\_NULL OR when Internal ADC sensor table y value (i.e pi16Y of s\_Table of Sensor\_as\_coldjunction with index as channel of Ptr\_sensor\_aas\_analog) is equal to M\_NULL.

##### 10.15.3.7.29 dauanapbit-McdCheck-LLR-029

Requirement ID: H398-LLD-ANA-FNC-1875

The function shall perform as mentioned below by looping through IADC\_CHAN\_1 to IADC\_CHANS-1:

-call the function BitErrorHandler with parameter(MCD\_ERR) when Internal ADC sensor table x value (i.e pi16X of s\_Table of Sensor\_as\_coldjunction with index as channel of Ptr\_sensor\_aas\_analog) is not equal to M\_NULL AND when Internal ADC sensor table y value (i.e pi16Y of s\_Table of Sensor\_as\_coldjunction with index as channel of Ptr\_sensor\_aas\_analog) is not equal to M\_NULL AND when Internal ADC minimum range (i.e i16\_Min\_Range of Sensor\_as\_coldjunction with index as channel of Ptr\_sensor\_aas\_analog) is less than M\_MIN\_RANGE\_MIN OR when Internal ADC minimum range (i.e i16\_Min\_Range of Sensor\_as\_coldjunction with index as channel of Ptr\_sensor\_aas\_analog) is greater than M\_MIN\_RANGE\_MAX.

##### 10.15.3.7.30 dauanapbit-McdCheck-LLR-030

Requirement ID: H398-LLD-ANA-FNC-1876

The function shall perform as mentioned below by looping through IADC\_CHAN\_1 to IADC\_CHANS-1:

-call the function BitErrorHandler with parameter(MCD\_ERR) when Internal ADC minimum range (i.e i16\_Min\_Range of Sensor\_as\_coldjunction with index as channel of Ptr\_sensor\_aas\_analog) is greater than or equal to M\_MIN\_RANGE\_MIN AND when Internal ADC minimum range (i.e i16\_Min\_Range of Sensor\_as\_coldjunction with index as channel of Ptr\_sensor\_aas\_analog) is less than or equal to M\_MIN\_RANGE\_MAX AND when Internal ADC maximum range (i.e i16\_Max\_Range of Sensor\_as\_coldjunction with index as channel of Ptr\_sensor\_aas\_analog) is greater than M\_MAX\_RANGE\_MAX OR when Internal ADC maximum range (i.e i16\_Max\_Range of Sensor\_as\_coldjunction with index as channel of Ptr\_sensor\_aas\_analog) is less than M\_MAX\_RANGE\_MIN, otherwise do nothing.

### 10.15.4 Crccheck

Low Level Design Details about CSU Crccheck will follow in the sub sections.

#### 10.15.4.1 Brief Description

This function does the CRC validity check.

#### 10.15.4.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.15.4.3 List of global variables accessed and modified

Accessed : U32\_checksum\_config

U32\_checksum\_flight

Modified : U32\_checksum\_config

U32\_checksum\_flight

#### 10.15.4.4 Parameter list (Input/Output)

Inputs : None

Outputs : None

#### 10.15.4.5 Return Value

None

#### 10.15.4.6 Other CSUs called by this CSU

BitErrorHandler,

CrcResetDr,

CrcCalcBlockCrc

#### 10.15.4.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to Crccheck.

##### 10.15.4.7.1 dauanapbit-CrcCheck-LLR-001

Requirement ID: H398-LLD-ANA-FNC-825

The function shall reset the CRC Data register by calling the function CrcResetDr and compute the CRC of MCD Software by calling the function CrcCalcBlockCrc with parameters M\_MEMMAP\_CONFIG\_DATA\_ADDR and M\_MEMMAP\_CONFIG\_DATA\_CRC\_CNT and set the computed MCD CRC to U32\_checksum\_config.

##### 10.15.4.7.2 dauanapbit-CrcCheck-LLR-002

Requirement ID: H398-LLD-ANA-FNC-826

The function shall call the function BitErrorHandler with parameter(CRC\_ERR) when the computed checksum for MCD Software (U32\_checksum\_config) is not equal to the CRC value (i.e.M\_MEMMAP\_CONFIG\_DATA\_CRC\_ADDR) otherwise do nothing.

##### 10.15.4.7.3 dauanapbit-CrcCheck-LLR-003

Requirement ID: H398-LLD-ANA-FNC-827

The function shall reset the CRC Data register by calling the function CrcResetDr and compute the checksum for the CRC of FLIGHT Software by calling the function CrcCalcBlockCrc with parameters M\_MEMMAP\_DATA\_1\_ADDR and M\_MEMMAP\_DATA\_1\_CRC\_CNT and set the computed Flight CRC to U32\_checksum\_flight.

##### 10.15.4.7.4 dauanapbit-CrcCheck-LLR-004

Requirement ID: H398-LLD-ANA-FNC-828

The function shall call the function BitErrorHandler with parameter(CRC\_ERR) when the computed checksum for FLIGHT Software(U32\_checksum\_flight) is not equal to the CRC value (M\_MEMMAP\_DATA\_1\_CRC\_ADDR) otherwise do nothing.

##### 10.15.4.7.5 dauanapbit-CrcCheck-LLR-005

Requirement ID: H398-LLD-ANA-FNC-829

The function shall reset the CRC Data register by calling the function CrcResetDr and compute the checksum for the CRC of Calibration Software by calling the function CrcCalcBlockCrc with parameters M\_MEMMAP\_CAL\_ADDR and M\_MEMMAP\_CAL\_CRC\_CNT.

##### 10.15.4.7.6 dauanapbit-CrcCheck-LLR-006

Requirement ID: H398-LLD-ANA-FNC-830

The function shall call the function BitErrorHandler with parameter(CRC\_ERR) when the computed checksum for Calibration Software is not equal to the CRC value (i.e.M\_MEMMAP\_CAL\_CRC\_ADDR) otherwise do nothing.

### 10.15.5 BitErrorHandler

Low Level Design Details about CSU BitErrorHandler will follow in the sub sections.

#### 10.15.5.1 Brief Description

This function handles the PBIT errors.

#### 10.15.5.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.15.5.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 10.15.5.4 Parameter list (Input/Output)

Inputs :

T\_UINT8 error\_code Error number

CRC\_ERR - if CRC check fails

CPU\_ERR - if CPU check fails

RAM\_ERR - if RAM check fails

MCD\_ERR - if MCD check fails

Outputs : None

#### 10.15.5.5 Return Value

None

#### 10.15.5.6 Other CSUs called by this CSU

WdogKickWatchDog,

GpioSetBits

#### 10.15.5.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to BitErrorHandler.

##### 

##### 10.15.5.7.1 dauanapbit-BitErrorHandler-LLR-001

Requirement ID: H398-LLD-ANA-FNC-839

The function shall disable all interrupts.

##### 10.15.5.7.2 dauanapbit-BitErrorHandler-LLR-002

Requirement ID: H398-LLD-ANA-FNC-840

The function shall

a) enter into an infinite loop to do the following when the error\_code received by the function is CRC\_ERR or CPU\_ERR or RAM\_ERR or MCD\_ERR:

- call the function WdogKickWatchDog () to reset the watchdog counter

- call the function GpioSetBits (M\_HW\_LED\_HB\_ON) with parameter (M\_GPIOB, M\_GPIOB\_LED\_HB) to make the heartbeat LED steady glow

b) do nothing when the error\_code received by the function is other than CRC\_ERR and CPU\_ERR and RAM\_ERR and MCD\_ERR

### 10.15.6 Pbitcheck

Low Level Design Details about CSU Pbitcheck will follow in the sub sections.

#### 10.15.6.1 Brief Description

This function does PBIT check.

#### 10.15.6.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.15.6.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 10.15.6.4 Parameter list (Input/Output)

Inputs : None

Outputs : None

#### 10.15.6.5 Return Value

None

#### 10.15.6.6 Other CSUs called by this CSU

CrcCheck,

CpuTest,

RamTest,

McdCheck

#### 10.15.6.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to Pbitcheck.

##### 10.15.6.7.1 dauanapbit-PbitCheck-LLR-001

Requirement ID: H398-LLD-ANA-FNC-849

The function shall call the function CrcCheck to do the crc test.

##### 10.15.6.7.2 dauanapbit-PbitCheck-LLR-002

Requirement ID: H398-LLD-ANA-FNC-850

The function shall call the function CpuTest to do the cpu test.

##### 10.15.6.7.3 dauanapbit-PbitCheck-LLR-003

Requirement ID: H398-LLD-ANA-FNC-851

The function shall call the function RamTest to do the internal ram test.

##### 10.15.6.7.4 dauanapbit-PbitCheck-LLR-004

Requirement ID: H398-LLD-ANA-FNC-852

The function shall call the function McdCheck to do the mcd test.

## 10.16 dauanarevno

The dauanarevno CSC defines the Top level software configuration number and Analog Software Application Part Number.

It does not contain any functions.

### 10.16.1 Brief Description

This file defines the Module DAU Application software part number and Top level software configuration number.

### 10.16.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD(H398-003-012-ANA).

### 10.16.3 List of global variables accessed and modified

Accessed : None

Modified : None

### 10.16.4 Parameter list (Input/Output)

None

### 10.16.5 Return Value

None

### 10.16.6 Other CSUs called by this CSU

None

### 10.16.7 Description of list of LLRs allocated

This Module does not provide any public operation as its functionality is to just provide the revision number. Hence, the LLR name does not contain a CSU name.

#### 10.16.7.1 dauanarevno-LLR-001

Requirement ID: H398-LLD-ANA-FNC-861

The CSC dauanarevno shall define the Top level software configuration number(SCN) and set the Analog Application Software Part Number to M\_REVNO\_REVISION.

## 10.17 dauanatbase

This module contains Implementation of timebase routines for signaling semaphores.

### 10.17.1 TbaseTaskSignaling

Low Level Design Details about CSU TbaseTaskSignaling will follow in the sub sections.

#### 10.17.1.1 Brief Description

This function installs the following task signaling parameters into the timebase to be serviced

- task ticks

- semaphore of the task

#### 10.17.1.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.17.1.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 10.17.1.4 Parameter list (Input/Output)

Inputs : T\_UINT16 u16\_task\_ticks - Task ticks to delay this thread

T\_OS\_EVENT \*ps\_semaphore - pointer to Semaphore to be posted for thread

Outputs : None

#### 10.17.1.5 Return Value

None

#### 10.17.1.6 Other CSUs called by this CSU

None

#### 10.17.1.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to TbaseTaskSignaling.

##### 10.17.1.7.1 dauanatbase-TbaseTaskSignaling-LLR-001

Requirement ID: H398-LLD-ANA-FNC-871

The function shall return when the total number of installed tasks is greater than and equal to M\_OS\_MAX\_TASKS.

##### 10.17.1.7.2 dauanatbase-TbaseTaskSignaling-LLR-002

Requirement ID: H398-LLD-ANA-FNC-872

The function shall reset the tick counter(i.e u16\_tick\_cntr) of the timebase task list to M\_ZERO when the total number of installed tasks is less than the M\_OS\_MAX\_TASKS.

##### 10.17.1.7.3 dauanatbase-TbaseTaskSignaling-LLR-003

Requirement ID: H398-LLD-ANA-FNC-873

The function shall save u16\_task\_ticks to the task ticks of the timebase task list when the total number of installed tasks is less than the M\_OS\_MAX\_TASKS.

##### 10.17.1.7.4 dauanatbase-TbaseTaskSignaling-LLR-004

Requirement ID: H398-LLD-ANA-FNC-874

The function shall save ps\_semaphore to the semaphore of the timebase task list when the total number of installed tasks is less than the M\_OS\_MAX\_TASKS.

##### 10.17.1.7.5 dauanatbase-TbaseTaskSignaling-LLR-005

Requirement ID: H398-LLD-ANA-FNC-875

The function shall increment number of installed tasks by M\_ONE when the total number of installed tasks is less than the M\_OS\_MAX\_TASKS.

### 10.17.2 TbaseIntrHandler

Low Level Design Details about CSU TbaseIntrHandler will follow in the sub sections.

#### 10.17.2.1 Brief Description

The function TbaseIntrHandler is interrupt handler for timebase interrupt.

#### 10.17.2.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.17.2.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 10.17.2.4 Parameter list (Input/Output)

Inputs : None

Outputs : None

#### 10.17.2.5 Return Value

None

#### 10.17.2.6 Other CSUs called by this CSU

OsIntEnter

OsTimeTick

OsSemPost

OsIntExit

ErrorHandler

#### 10.17.2.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to TbaseIntrHandler.

##### 10.17.2.7.1 dauanatbase-TbaseIntrHandler-LLR-001

Requirement ID: H398-LLD-ANA-FNC-884

The function shall inform the uC/OS that, it is entering the ISR by calling OsIntEnter.

##### 10.17.2.7.2 dauanatbase-TbaseIntrHandler-LLR-002

Requirement ID: H398-LLD-ANA-FNC-885

The function shall update the time tick by calling OsTimeTick.

##### 10.17.2.7.3 dauanatbase-TbaseIntrHandler-LLR-003

Requirement ID: H398-LLD-ANA-FNC-886

The function shall loop through the timebase task list and perform the following for each task in the list:

a) when the task tick counter (incremented by 1) is greater than or equal to the task ticks of the corresponding task.

i) Reset the task tick counter to M\_ZERO.

ii)Signal the semaphore of the task by calling 'OsSemPost' with semaphore of the corresponding task as parameter.

iii)Call the function 'ErrorHandler' when the function 'OsSemPost' returns other than M\_OS\_NO\_ERR otherwise do nothing.

b) do nothing when the task tick counter (incremented by 1) is less than the task ticks of the corresponding task.

##### 10.17.2.7.4 dauanatbase-TbaseIntrHandler-LLR-004

Requirement ID: H398-LLD-ANA-FNC-887

The function shall inform the uC/OS, that it is leaving the ISR by calling OsIntExit.

### 10.17.3 TbaseInit

Low Level Design Details about CSU TbaseInit will follow in the sub sections.

#### 10.17.3.1 Brief Description

The function TbaseInit initializes the system tick interrupt period.

#### 10.17.3.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.17.3.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 10.17.3.4 Parameter list (Input/Output)

Inputs : None

Outputs : None

#### 10.17.3.5 Return Value

None

#### 10.17.3.6 Other CSUs called by this CSU

SysTickConfig

#### 10.17.3.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to TbaseInit.

##### 10.17.3.7.1 dauanatbase-TbaseInit-LLR-001

Requirement ID: H398-LLD-ANA-FNC-896

The function shall initialize the system tick interrupt period by calling SysTickConfig with number of ticks between two interrupts as parameter. (M\_HW\_SYSTEM\_CLOCK / M\_OS\_TICKS\_PER\_SEC).

## 10.18 dauanatmr

This module contains routines to Implement timers for adc.

### 10.18.1 TMRInitTIM6

Low Level Design Details about CSU TMRInitTIM6 will follow in the sub sections.

#### 10.18.1.1 Brief Description

The function TMRInitTIM6 Initializes Timer 6 to trigger internal ADC conversion.

#### 10.18.1.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.18.1.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 10.18.1.4 Parameter list (Input/Output)

Inputs : T\_INTR\_FN pfnTriggerInternalADC -Reference of the function to Trigger Internal ADC

Outputs : None

#### 10.18.1.5 Return Value

None

#### 10.18.1.6 Other CSUs called by this CSU

RccApb1PeriphClockCmd,

IntrInstall,

NvicInit,

TimPrescalerConfig,

TimSetAutoReload,

TimCmd,

TimItConfig

#### 10.18.1.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to TMRInitTIM6.

##### 10.18.1.7.1 dauanatmr-TMRInitTIM6-LLR-001

Requirement ID: H398-LLD-ANA-FNC-906

The function shall call 'RccApb1PeriphClockCmd' with parameters as M\_RCC\_APB1PERIPH\_TIM6 and ENABLE to enable the TIM 6 clock.

##### 10.18.1.7.2 dauanatmr-TMRInitTIM6-LLR-002

Requirement ID: H398-LLD-ANA-FNC-907

The function shall call 'IntrInstall' with parameters as INTR\_TIM\_6\_ADC and pfnTriggerInternalADC to Install IADC period interrupt into vector table in RAM.

##### 10.18.1.7.3 dauanatmr-TMRInitTIM6-LLR-003

Requirement ID: H398-LLD-ANA-FNC-908

The function shall Call function NvicInit with parameter as reference to NVIC init structure to Initialize the NVIC peripheral (M\_TIM6 global Interrupt)to the value given below:

- Set nvic\_irq\_channel to TIM6\_DAC\_IRQN

- Set nvic\_irq\_channel\_preemption\_priority to NVIC\_PRIORITY\_LEVEL\_7

- Set nvic\_irq\_channel\_subpriority to NVIC\_PRIORITY\_LEVEL\_0

- Set nvic\_irq\_channel\_cmd to ENABLE

##### 10.18.1.7.4 dauanatmr-TMRInitTIM6-LLR-004

Requirement ID: H398-LLD-ANA-FNC-909

The function shall call 'TimPrescalerConfig' with parameters as M\_TIM6, M\_TMR\_6\_SCALE minus M\_ONE and M\_TIM\_PSCRELOADMODE\_UPDATE to Configure prescaler for timer 6.

##### 10.18.1.7.5 dauanatmr-TMRInitTIM6-LLR-005

Requirement ID: H398-LLD-ANA-FNC-910

The function shall call 'TimSetAutoReload' with parameters as M\_TIM6 and M\_TMR\_6\_PERIOD minus M\_ONE to set timer 6 counter reset value.

##### 10.18.1.7.6 dauanatmr-TMRInitTIM6-LLR-006

Requirement ID: H398-LLD-ANA-FNC-911

The function shall call 'TimCmd' with parameters as M\_TIM6 and ENABLE to enable timer 6 counter.

##### 10.18.1.7.7 dauanatmr-TMRInitTIM6-LLR-007

Requirement ID: H398-LLD-ANA-FNC-912

The function shall call 'TimItConfig' with parameters as M\_TIM6, M\_TIM\_IT\_UPDATE and ENABLE to enable the UIF Interrupt Request for timer 6.

### 10.18.2 TMRInitTIM7

Low Level Design Details about CSU TMRInitTIM7 will follow in the sub sections.

#### 10.18.2.1 Brief Description

The function TMRInitTIM7 initializes timer 7 to read external ADC.

#### 10.18.2.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.18.2.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 10.18.2.4 Parameter list (Input/Output)

Inputs : T\_INTR\_FN ReadExternalADC Reference of the function to Trigger External ADC

Outputs : None

#### 10.18.2.5 Return Value

None

#### 10.18.2.6 Other CSUs called by this CSU

RccApb1PeriphClockCmd,

IntrInstall,

NvicInit,

TimPrescalerConfig,

TimSetAutoReload,

TimCmd,

TimItConfig

#### 10.18.2.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to TMRInitTIM7.

##### 10.18.2.7.1 dauanatmr-TMRInitTIM7-LLR-001

Requirement ID: H398-LLD-ANA-FNC-921

The function shall call 'RccApb1PeriphClockCmd' with parameters as M\_RCC\_APB1PERIPH\_TIM7 and ENABLE to enable the TIM 7 clock.

##### 10.18.2.7.2 dauanatmr-TMRInitTIM7-LLR-002

Requirement ID: H398-LLD-ANA-FNC-922

The function shall call 'IntrInstall' with parameters as INTR\_TIM\_7 and ReadExternalADC to install XADC period interrupt into vector table in RAM.

##### 10.18.2.7.3 dauanatmr-TMRInitTIM7-LLR-003

Requirement ID: H398-LLD-ANA-FNC-923

The function shall call function NvicInit with parameters as reference to NVIC init structure to Initialize the NVIC peripheral (M\_TIM7 global Interrupt)to the value given below

- Set nvic\_irq\_channel to TIM7\_IRQN

- Set nvic\_irq\_channel\_preemption\_priority to NVIC\_PRIORITY\_LEVEL\_7

- Set nvic\_irq\_channel\_subpriority to NVIC\_PRIORITY\_LEVEL\_0

- Set nvic\_irq\_channel\_cmd to ENABLE

##### 10.18.2.7.4 dauanatmr-TMRInitTIM7-LLR-004

Requirement ID: H398-LLD-ANA-FNC-924

The function shall call 'TimPrescalerConfig' with parameters as M\_TIM7, M\_TMR\_7\_SCALE minus M\_ONE and M\_TIM\_PSCRELOADMODE\_UPDATE to Configure prescaler for timer 7.

##### 10.18.2.7.5 dauanatmr-TMRInitTIM7-LLR-005

Requirement ID: H398-LLD-ANA-FNC-925

The function shall call 'TimSetAutoReload' with parameters as M\_TIM7 and M\_TMR\_7\_PERIOD minus M\_ONE to set timer 7 counter reset value.

##### 10.18.2.7.6 dauanatmr-TMRInitTIM7-LLR-006

Requirement ID: H398-LLD-ANA-FNC-926

The function shall call 'TimCmd' with parameters as M\_TIM7 and ENABLE to enable timer 7 counter.

##### 10.18.2.7.7 dauanatmr-TMRInitTIM7-LLR-007

Requirement ID: H398-LLD-ANA-FNC-927

The function shall call 'TimItConfig' with parameters as M\_TIM7, M\_TIM\_IT\_UPDATE and ENABLE to enable the UIF Interrupt Request for timer 7.

### 10.18.3 TMRInitTIM12

Low Level Design Details about CSU TMRInitTIM12 will follow in the sub sections.

#### 10.18.3.1 Brief Description

The function TMRInitTIM12 initializes timer 12 for tach driver

#### 10.18.3.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.18.3.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 10.18.3.4 Parameter list (Input/Output)

Inputs: T\_INTR\_FN pfnTachDriver Reference of the function to Tach Driver

Outputs: None

#### 10.18.3.5 Return Value

None

#### 10.18.3.6 Other CSUs called by this CSU

RccApb1PeriphClockCmd,

IntrInstall,

NvicInit,

TimPrescalerConfig,

TimSetAutoReload,

TimCmd,

TimItConfig

#### 10.18.3.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to TMRInitTIM12.

##### 10.18.3.7.1 dauanatmr-TMRInitTIM12-LLR-001

Requirement ID: H398-LLD-ANA-FNC-2011

The function shall call 'RccApb1PeriphClockCmd' with parameters as M\_RCC\_APB1PERIPH\_TIM12 and ENABLE to enable the TIM 12 clock.

##### 10.18.3.7.2 dauanatmr-TMRInitTIM12-LLR-002

Requirement ID: H398-LLD-ANA-FNC-2012

The function shall call 'IntrInstall' with parameters as INTR\_TIM\_8\_BRK\_TIM\_12 and pfnTachDriver to Install tach driver interrupt into vector table in RAM.

##### 10.18.3.7.3 dauanatmr-TMRInitTIM12-LLR-003

Requirement ID: H398-LLD-ANA-FNC-2013

The function shall Call function NvicInit with parameter as reference to NVIC init structure to Initialize the NVIC peripheral (M\_TIM12 global Interrupt) to the value given below:

- Set nvic\_irq\_channel to TIM8\_BRK\_TIM12\_IRQN

- Set nvic\_irq\_channel\_preemption\_priority to NVIC\_PRIORITY\_LEVEL\_4

- Set nvic\_irq\_channel\_subpriority to NVIC\_PRIORITY\_LEVEL\_0

- Set nvic\_irq\_channel\_cmd to ENABLE

##### 10.18.3.7.4 dauanatmr-TMRInitTIM12-LLR-004

Requirement ID: H398-LLD-ANA-FNC-2014

The function shall call 'TimPrescalerConfig' with parameters as M\_TIM12, M\_TMR\_12\_SCALE minus M\_ONE and M\_TIM\_PSCRELOADMODE\_UPDATE to Configure prescaler for timer 12.

##### 10.18.3.7.5 dauanatmr-TMRInitTIM12-LLR-005

Requirement ID: H398-LLD-ANA-FNC-2015

The function shall call 'TimSetAutoReload' with parameters as M\_TIM12 and M\_TMR\_12\_PERIOD minus M\_ONE to set timer 12 counter reset value.

##### 10.18.3.7.6 dauanatmr-TMRInitTIM12-LLR-006

Requirement ID: H398-LLD-ANA-FNC-2016

The function shall call 'TimCmd' with parameters as M\_TIM12 and ENABLE to enable timer 12 counter.

##### 10.18.3.7.7 dauanatmr-TMRInitTIM12-LLR-007

Requirement ID: H398-LLD-ANA-FNC-2017

The function shall call 'TimItConfig' with parameters as M\_TIM12, M\_TIM\_IT\_UPDATE and ENABLE to enable the UIF Interrupt Request for timer 12.

### 10.18.4 GetDivConfig

Low Level Design Details about CSU GetDivConfig will follow in the sub sections.

#### 10.18.4.1 Brief Description

The function GetDivConfig Set the frequency to Capture Prescaler

#### 10.18.4.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.18.4.3 List of global variables accessed and modified.

Accessed : None

Modified : None

#### 10.18.4.4 Parameter list (Input/Output)

Inputs: T\_UINT8 u8FreqDiv -> frequency divider

Outputs: None

#### 10.18.4.5 Return Value

T\_UINT16 ->Return frequency to Capture Prescaler

Return value shall be one of the following:

-- M\_TIM\_ICPSC\_DIV1

-- M\_TIM\_ICPSC\_DIV2

-- M\_TIM\_ICPSC\_DIV4

-- M\_TIM\_ICPSC\_DIV8

#### 10.18.4.6 Other CSUs called by this CSU

None

#### 10.18.4.7 Description of list of LLRs allocated.

The following section will list the LLRs allocated to GetDivConfig.

##### 10.18.4.7.1 dauanatmr-GetDivConfig-LLR-001

Requirement ID: H398-LLD-ANA-FNC-2026

The function shall return M\_TIM\_ICPSC\_DIV2 when u8FreqDiv is equal to M\_TMR\_FREQ\_DIV\_2.

##### 10.18.4.7.2 dauanatmr-GetDivConfig-LLR-002

Requirement ID: H398-LLD-ANA-FNC-2027

The function shall return M\_TIM\_ICPSC\_DIV4 when u8FreqDiv is equal to M\_TMR\_FREQ\_DIV\_4.

##### 10.18.4.7.3 dauanatmr-GetDivConfig-LLR-003

Requirement ID: H398-LLD-ANA-FNC-2028

The function shall return M\_TIM\_ICPSC\_DIV8 when u8FreqDiv is equal to M\_TMR\_FREQ\_DIV\_8.

##### 10.18.4.7.4 dauanatmr-GetDivConfig-LLR-004

Requirement ID: H398-LLD-ANA-FNC-2029

The function shall return M\_TIM\_ICPSC\_DIV1 when u8FreqDiv is other than M\_TMR\_FREQ\_DIV\_2, M\_TMR\_FREQ\_DIV\_4, M\_TMR\_FREQ\_DIV\_8.

### 10.18.5 TMRInitTach1

Low Level Design Details about CSU TMRInitTach1 will follow in the sub sections.

#### 10.18.5.1 Brief Description

The function TMRInitTach1 Initialize timer 8 channel 1 for tach 1, DMA channel and Tach 1 GPIO pins for communication.

#### 10.18.5.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.18.5.3 List of global variables accessed and modified.

Accessed : None

Modified : None

#### 10.18.5.4 Parameter list (Input/Output)

Inputs: T\_UINT8 u8FreqDiv - Frequency divider of the Tach channel

T\_UINT16 \*pu16BufferDMA - Reference to the DMA buffer

T\_UINT16 u16BufferSize - Size of the buffer

Outputs: None

#### 10.18.5.5 Return Value

None

#### 10.18.5.6 Other CSUs called by this CSU

RccAhb1PeriphClockCmd

RccApb2PeriphClockCmd

GpioPinAFConfig

GpioInit

DmaDeInit

DmaInit

DmaCmd

TimICInit

TimDmaCmd

TimPrescalerConfig

TimCmd

#### 10.18.5.7 Description of list of LLRs allocated.

The following section will list the LLRs allocated to TMRInitTach1.

##### 10.18.5.7.1 dauanatmr-TMRInitTach1-LLR-001

Requirement ID: H398-LLD-ANA-FNC-2039

The function shall call RccAhb1PeriphClockCmd with parameters M\_RCC\_AHB1PERIPH\_DMA2 and ENABLE to Enable DMA2 clock.

##### 10.18.5.7.2 dauanatmr-TMRInitTach1-LLR-002

Requirement ID: H398-LLD-ANA-FNC-2040

The function shall call RccApb2PeriphClockCmd with parameters M\_RCC\_APB2PERIPH\_TIM8 and ENABLE to Enable TIM8 clock.

##### 10.18.5.7.3 dauanatmr-TMRInitTach1-LLR-003

Requirement ID: H398-LLD-ANA-FNC-2041

The function shall call RccAhb1PeriphClockCmd with parameters M\_RCC\_AHB1PERIPH\_GPIOC and ENABLE to Enable M\_GPIOC clock.

##### 10.18.5.7.4 dauanatmr-TMRInitTach1-LLR-004

Requirement ID: H398-LLD-ANA-FNC-2042

The function shall call GpioPinAFConfig with parameters M\_GPIOC, M\_GPIO\_PINSOURCE6 and M\_GPIO\_AF\_TIM8 to Configure pin function for tach 1.

##### 10.18.5.7.5 dauanatmr-TMRInitTach1-LLR-005

Requirement ID: H398-LLD-ANA-FNC-2043

The function shall call GpioInit with parameters as reference to GPIOC init structure and M\_GPIOC to Configure tach 1 input pin to the value given below:

* Set gpio pin to M\_GPIOC\_TACH\_1
* Set gpio mode to GPIO\_MODE\_AF
* Set gpio speed to GPIO\_SPEED\_50MHZ
* Set gpio pupd to GPIO\_PUPD\_NOPULL

##### 10.18.5.7.6 dauanatmr-TMRInitTach1-LLR-006

Requirement ID: H398-LLD-ANA-FNC-2044

The function shall call DmaDeInit with parameter M\_DMA2\_STREAM2 to Reset M\_DMA2, stream 2 to original values.

##### 10.18.5.7.7 dauanatmr-TMRInitTach1-LLR-007

Requirement ID: H398-LLD-ANA-FNC-2045

The function shall call DmaInit with parameter as reference to dma init structure and M\_DMA2\_STREAM2 to Configure DMA2, stream 2, channel 7 to the value given below:

* Set Dma channel to M\_DMA\_CHANNEL\_7
* Set Dma peripheral baseaddr to address of ccr1 of M\_TIM8
* Set Dma memory0 baseaddr to pu16BufferDMA.
* Set Dma dir to M\_DMA\_DIR\_PERIPHERAL\_TO\_MEMORY
* Set Dma buffersize to u16BufferSize.
* Set Dma peripheral inc to M\_DMA\_PERIPHERALINC\_DISABLE
* Set Dma memory inc to M\_DMA\_MEMORYINC\_ENABLE
* Set Dma peripheral datasize to M\_DMA\_PERIPH\_DATASIZE\_HALFWORD
* Set Dma memory datasize to M\_DMA\_MEMORY\_DATASIZE\_HALFWORD
* Set Dma mode to M\_DMA\_MODE\_CIRCULAR
* Set Dma priority to M\_DMA\_PRIORITY\_HIGH
* Set Dma fifo mode to M\_DMA\_FIFOMODE\_DISABLE
* Set Dma fifo threshold to M\_DMA\_FIFOTHRESHOLD\_FULL
* Set Dma memory burst to M\_DMA\_MEMORYBURST\_SINGLE

- Set Dma peripheral burst to M\_DMA\_PERIPHERALBURST\_SINGLE

##### 10.18.5.7.8 dauanatmr-TMRInitTach1-LLR-008

Requirement ID: H398-LLD-ANA-FNC-2046

The function shall call DmaCmd with parameters M\_DMA2\_STREAM2 and ENABLE to Enable M\_DMA2 stream 2.

##### 10.18.5.7.9 dauanatmr-TMRInitTach1-LLR-009

Requirement ID: H398-LLD-ANA-FNC-2047

The function shall call TimICInit with parameters as reference to tim icinit structure and M\_TIM8 to Configure channel 1 input of timer 8 to the value given below:

* Set Tim channel to M\_TIM\_CHANNEL\_1
* Set Tim ic polarity to M\_TIM\_ICPOLARITY\_RISING
* Set Tim ic selection to M\_TIM\_ICSELECTION\_DIRECTTI
* Set Tim ic prescaler by calling function GetDivConfig with parameter u8FreqDiv to set the frequency to Capture Prescaler
* Set Tim ic filter to M\_TIM\_ICFILTER\_15

##### 10.18.5.7.10 dauanatmr-TMRInitTach1-LLR-010

Requirement ID: H398-LLD-ANA-FNC-2048

The function shall call TimDmaCmd with parameters M\_TIM8, M\_TIM\_DMA\_CC1 and ENABLE to Enable DMA trigger for channel 1.

##### 10.18.5.7.11 dauanatmr-TMRInitTach1-LLR-011

Requirement ID: H398-LLD-ANA-FNC-2049

The function shall call TimPrescalerConfig with parameters M\_TIM8, M\_TMR\_8\_SCALE minus M\_ONE, and M\_TIM\_PSCRELOADMODE\_UPDATE to Set pre-scale for internal timer 8.

##### 10.18.5.7.12 dauanatmr-TMRInitTach1-LLR-012

Requirement ID: H398-LLD-ANA-FNC-2050

The function shall call TimCmd with parameters M\_TIM8 and ENABLE to Enable timer 8 to start the counter

### 10.18.6 TMRInitTach2

Low Level Design Details about CSU TMRInitTach2 will follow in the sub sections.

#### 10.18.6.1 Brief Description

The function TMRInitTach2 Initialize timer 8 channel 2 for tach 2, DMA channel and Tach 2 GPIO pins for communication.

#### 10.18.6.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.18.6.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 10.18.6.4 Parameter list (Input/Output)

Inputs: T\_UINT8 u8FreqDiv - Frequency divider of the Tach channel

T\_UINT16 \*pu16BufferDMA - Reference to the DMA buffer

T\_UINT16 u16BufferSize - Size of the buffer

Outputs: None

#### 10.18.6.5 Return Value

None

#### 10.18.6.6 Other CSUs called by this CSU

RccAhb1PeriphClockCmd

RccApb2PeriphClockCmd

GpioPinAFConfig

GpioInit

DmaDeInit

DmaInit

DmaCmd

TimICInit

TimDmaCmd

TimPrescalerConfig

TimCmd

#### 10.18.6.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to TMRInitTach2.

10.18.6.7.1 dauanatmr-TMRInitTach2-LLR-001

Requirement ID: H398-LLD-ANA-FNC-2059

The function shall call RccAhb1PeriphClockCmd with parameters M\_RCC\_AHB1PERIPH\_DMA2 and ENABLE to Enable DMA2 clock.

##### 10.18.6.7.2 dauanatmr-TMRInitTach2-LLR-002

Requirement ID: H398-LLD-ANA-FNC-2060

The function shall call RccApb2PeriphClockCmd with parameters M\_RCC\_APB2PERIPH\_TIM8 and ENABLE to Enable M\_TIM8 clock.

##### 10.18.6.7.3 dauanatmr-TMRInitTach2-LLR-003

Requirement ID: H398-LLD-ANA-FNC-2061

The function shall call RccAhb1PeriphClockCmd with parameters M\_RCC\_AHB1PERIPH\_GPIOC and ENABLE to Enable M\_GPIOC clock.

##### 10.18.6.7.4 dauanatmr-TMRInitTach2-LLR-004

Requirement ID: H398-LLD-ANA-FNC-2062

The function shall call GpioPinAFConfig with parameters M\_GPIOC, M\_GPIO\_PINSOURCE7 and M\_GPIO\_AF\_TIM8 to Configure pin function for tach 2.

##### 10.18.6.7.5 dauanatmr-TMRInitTach2-LLR-005

Requirement ID: H398-LLD-ANA-FNC-2063

The function shall call GpioInit with parameters as reference to GPIOC init structure and M\_GPIOC to Configure tach 2 input pin to the value given below:

* Set gpio pin to M\_GPIOC\_TACH\_2
* Set gpio mode to GPIO\_MODE\_AF
* Set gpio speed to GPIO\_SPEED\_50MHZ
* Set gpio pupd to GPIO\_PUPD\_NOPULL

##### 10.18.6.7.6 dauanatmr-TMRInitTach2-LLR-006

Requirement ID: H398-LLD-ANA-FNC-2064

The function shall call DmaDeInit with parameter M\_DMA2\_STREAM3 to Reset M\_DMA2, stream 3 to original values.

##### 10.18.6.7.7 dauanatmr-TMRInitTach2-LLR-007

Requirement ID: H398-LLD-ANA-FNC-2065

The function shall call DmaInit with parameter as reference to dma init structure and M\_DMA2\_STREAM3 to Configure M\_DMA2, stream 3, channel 7 to the value given below:

* Set Dma channel to M\_DMA\_CHANNEL\_7
* Set Dma peripheral baseaddr to address of ccr2 of M\_TIM8
* Set Dma memory0 baseaddr to pu16BufferDMA.
* Set Dma dir to M\_DMA\_DIR\_PERIPHERAL\_TO\_MEMORY
* Set Dma buffersize to u16BufferSize.
* Set Dma peripheral inc to M\_DMA\_PERIPHERALINC\_DISABLE
* Set Dma memory inc to M\_DMA\_MEMORYINC\_ENABLE
* Set Dma peripheral datasize to M\_DMA\_PERIPH\_DATASIZE\_HALFWORD
* Set Dma memory datasize to M\_DMA\_MEMORY\_DATASIZE\_HALFWORD
* Set Dma mode to M\_DMA\_MODE\_CIRCULAR
* Set Dma priority to M\_DMA\_PRIORITY\_HIGH
* Set Dma fifo mode to M\_DMA\_FIFOMODE\_DISABLE
* Set Dma fifo threshold to M\_DMA\_FIFOTHRESHOLD\_FULL
* Set Dma memory burst to M\_DMA\_MEMORYBURST\_SINGLE

- Set Dma peripheral burst to M\_DMA\_PERIPHERALBURST\_SINGLE

##### 10.18.6.7.8 dauanatmr-TMRInitTach2-LLR-008

Requirement ID: H398-LLD-ANA-FNC-2066

The function shall call DmaCmd with parameters M\_DMA2\_STREAM3 and ENABLE to Enable M\_DMA2 stream 3.

##### 10.18.6.7.9 dauanatmr-TMRInitTach2-LLR-009

Requirement ID: H398-LLD-ANA-FNC-2067

The function shall call TimICInit with parameters as reference to tim icinit structure and M\_TIM8 to Configure channel 2 input of timer 8 to the value given below:

* Set Tim channel to M\_TIM\_CHANNEL\_2
* Set Tim ic polarity to M\_TIM\_ICPOLARITY\_RISING
* Set Tim ic selection to M\_TIM\_ICSELECTION\_DIRECTTI
* Set Tim ic prescaler by calling function GetDivConfig with parameter u8FreqDiv to set the frequency to Capture Prescaler
* Set Tim ic filter to M\_TIM\_ICFILTER\_15

##### 10.18.6.7.10 dauanatmr-TMRInitTach2-LLR-010

Requirement ID: H398-LLD-ANA-FNC-2068

The function shall call TimDmaCmd with parameters M\_TIM8, M\_TIM\_DMA\_CC2 and ENABLE to Enable DMA trigger for channel 2.

##### 10.18.6.7.11 dauanatmr-TMRInitTach2-LLR-011

Requirement ID: H398-LLD-ANA-FNC-2069

The function shall call TimPrescalerConfig with parameters M\_TIM8, M\_TMR\_8\_SCALE minus M\_ONE, and M\_TIM\_PSCRELOADMODE\_UPDATE to Set pre-scale for internal timer 8.

##### 10.18.6.7.12 dauanatmr-TMRInitTach2-LLR-012

Requirement ID: H398-LLD-ANA-FNC-2070

The function shall call TimCmd with parameters M\_TIM8 and ENABLE to Enable timer 8 to start the counter.

### 10.18.7 TMRInitTach3

Low Level Design Details about CSU TMRInitTach3 will follow in the sub sections.

#### 10.18.7.1 Brief Description

The function TMRInitTach3 Initialize timer 8 channel 3 for tach 3, DMA channel and Tach 3 GPIO pins for communication.

#### 10.18.7.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.18.7.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 10.18.7.4 Parameter list (Input/Output)

Inputs: T\_UINT8 u8FreqDiv - Frequency divider of the Tach channel

T\_UINT16 \*pu16BufferDMA - Reference to the DMA buffer

T\_UINT16 u16BufferSize - Size of the buffer

Outputs: None

#### 10.18.7.5 Return Value

None

#### 10.18.7.6 Other CSUs called by this CSU

RccAhb1PeriphClockCmd

RccApb2PeriphClockCmd

GpioPinAFConfig

GpioInit

DmaDeInit

DmaInit

DmaCmd

TimICInit

TimDmaCmd

TimPrescalerConfig

TimCmd

#### 10.18.7.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to TMRInitTach3.

##### 10.18.7.7.1 dauanatmr-TMRInitTach3-LLR-001

Requirement ID: H398-LLD-ANA-FNC-2079

The function shall call RccAhb1PeriphClockCmd with parameters M\_RCC\_AHB1PERIPH\_DMA2 and ENABLE to Enable DMA2 clock.

##### 10.18.7.7.2 dauanatmr-TMRInitTach3-LLR-002

Requirement ID: H398-LLD-ANA-FNC-2080

The function shall call RccApb2PeriphClockCmd with parameters M\_RCC\_APB2PERIPH\_TIM8 and ENABLE to Enable M\_TIM8 clock.

##### 10.18.7.7.3 dauanatmr-TMRInitTach3-LLR-003

Requirement ID: H398-LLD-ANA-FNC-2081

The function shall call RccAhb1PeriphClockCmd with parameters M\_RCC\_AHB1PERIPH\_GPIOC and ENABLE to Enable M\_GPIOC clock.

##### 10.18.7.7.4 dauanatmr-TMRInitTach3-LLR-004

Requirement ID: H398-LLD-ANA-FNC-2082

The function shall call GpioPinAFConfig with parameters M\_GPIOC, M\_GPIO\_PINSOURCE8 and M\_GPIO\_AF\_TIM8 to Configure pin function for tach 3.

##### 10.18.7.7.5 dauanatmr-TMRInitTach3-LLR-005

Requirement ID: H398-LLD-ANA-FNC-2083

The function shall call GpioInit with parameters as reference to GPIOC init structure and M\_GPIOC to Configure tach 3 input pin to the value given below:

* Set gpio pin to M\_GPIOC\_TACH\_3
* Set gpio mode to GPIO\_MODE\_AF
* Set gpio speed to GPIO\_SPEED\_50MHZ
* Set gpio pupd to GPIO\_PUPD\_NOPULL

##### 10.18.7.7.6 dauanatmr-TMRInitTach3-LLR-006

Requirement ID: H398-LLD-ANA-FNC-2084

The function shall call DmaDeInit with parameter M\_DMA2\_STREAM4 to Reset M\_DMA2, stream 4 to original values.

##### 10.18.7.7.7 dauanatmr-TMRInitTach3-LLR-007

Requirement ID: H398-LLD-ANA-FNC-2085

The function shall call DmaInit with parameter as reference to dma init structure and M\_DMA2\_STREAM4 to Configure M\_DMA2, stream 4, channel 7 to the value given below:

* Set Dma channel to M\_DMA\_CHANNEL\_7
* Set Dma peripheral baseaddr to address of ccr3 of M\_TIM8
* Set Dma memory0 baseaddr to pu16BufferDMA.
* Set Dma dir to M\_DMA\_DIR\_PERIPHERAL\_TO\_MEMORY
* Set Dma buffersize to u16BufferSize.
* Set Dma peripheral inc to M\_DMA\_PERIPHERALINC\_DISABLE
* Set Dma memory inc to M\_DMA\_MEMORYINC\_ENABLE
* Set Dma peripheral datasize to M\_DMA\_PERIPH\_DATASIZE\_HALFWORD
* Set Dma memory datasize to M\_DMA\_MEMORY\_DATASIZE\_HALFWORD
* Set Dma mode to M\_DMA\_MODE\_CIRCULAR
* Set Dma priority to M\_DMA\_PRIORITY\_HIGH
* Set Dma fifo mode to M\_DMA\_FIFOMODE\_DISABLE
* Set Dma fifo threshold to M\_DMA\_FIFOTHRESHOLD\_FULL
* Set Dma memory burst to M\_DMA\_MEMORYBURST\_SINGLE

- Set Dma peripheral burst to M\_DMA\_PERIPHERALBURST\_SINGLE

##### 10.18.7.7.8 dauanatmr-TMRInitTach3-LLR-008

Requirement ID: H398-LLD-ANA-FNC-2086

The function shall call DmaCmd with parameters M\_DMA2\_STREAM4 and ENABLE to Enable M\_DMA2 stream 4.

##### 10.18.7.7.9 dauanatmr-TMRInitTach3-LLR-009

Requirement ID: H398-LLD-ANA-FNC-2087

The function shall call TimICInit with parameters as reference to tim icinit structure and M\_TIM8 to Configure channel 3 input of timer 8 to the value given below:

* Set Tim channel to M\_TIM\_CHANNEL\_3
* Set Tim ic polarity to M\_TIM\_ICPOLARITY\_RISING
* Set Tim ic selection to M\_TIM\_ICSELECTION\_DIRECTTI
* Set Tim ic prescaler by calling function GetDivConfig with parameter u8FreqDiv to set the frequency to Capture Prescaler
* Set Tim ic filter to M\_TIM\_ICFILTER\_15

##### 10.18.7.7.10 dauanatmr-TMRInitTach3-LLR-010

Requirement ID: H398-LLD-ANA-FNC-2088

The function shall call TimDmaCmd with parameters M\_TIM8, M\_TIM\_DMA\_CC3 and ENABLE to Enable DMA trigger for channel 3.

##### 10.18.7.7.11 dauanatmr-TMRInitTach3-LLR-011

Requirement ID: H398-LLD-ANA-FNC-2089

The function shall call TimPrescalerConfig with parameters M\_TIM8, M\_TMR\_8\_SCALE minus M\_ONE, and M\_TIM\_PSCRELOADMODE\_UPDATE to Set pre-scale for internal timer 8.

##### 10.18.7.7.12 dauanatmr-TMRInitTach3-LLR-012

Requirement ID: H398-LLD-ANA-FNC-2090

The function shall call TimCmd with parameters M\_TIM8 and ENABLE to Enable timer 8 to start the counter.

### 10.18.8 TMRInitTach4

Low Level Design Details about CSU TMRInitTach4 will follow in the sub sections.

#### 10.18.8.1 Brief Description

The function TMRInitTach4 Initialize timer 8 channel 4 for tach 4 DMA channel and Tach 4 GPIO pins for communication.

#### 10.18.8.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.18.8.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 10.18.8.4 Parameter list (Input/Output)

Inputs: T\_UINT8 u8FreqDiv - Frequency divider of the Tach channel

T\_UINT16 \*pu16BufferDMA - Reference to the DMA buffer

T\_UINT16 u16BufferSize - Size of the buffer

Outputs: None

#### 10.18.8.5 Return Value

None

#### 10.18.8.6 Other CSUs called by this CSU

RccAhb1PeriphClockCmd

RccApb2PeriphClockCmd

GpioPinAFConfig

GpioInit

DmaDeInit

DmaInit

DmaCmd

TimICInit

TimDmaCmd

TimPrescalerConfig

TimCmd

#### 10.18.8.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to TMRInitTach4.

##### 10.18.8.7.1 dauanatmr-TMRInitTach4-LLR-001

Requirement ID: H398-LLD-ANA-FNC-2099

The function shall call RccAhb1PeriphClockCmd with parameters M\_RCC\_AHB1PERIPH\_DMA2 and ENABLE to Enable DMA2 clock.

##### 10.18.8.7.2 dauanatmr-TMRInitTach4-LLR-002

Requirement ID: H398-LLD-ANA-FNC-2100

The function shall call RccApb2PeriphClockCmd with parameters M\_RCC\_APB2PERIPH\_TIM8 and ENABLE to Enable TIM8 clock.

##### 10.18.8.7.3 dauanatmr-TMRInitTach4-LLR-003

Requirement ID: H398-LLD-ANA-FNC-2101

The function shall call RccAhb1PeriphClockCmd with parameters M\_RCC\_AHB1PERIPH\_GPIOC and ENABLE to Enable M\_GPIOC clock.

##### 10.18.8.7.4 dauanatmr-TMRInitTach4-LLR-004

Requirement ID: H398-LLD-ANA-FNC-2102

The function shall call GpioPinAFConfig with parameters M\_GPIOC, M\_GPIO\_PINSOURCE9 and M\_GPIO\_AF\_TIM8 to Configure pin function for tach 4.

##### 10.18.8.7.5 dauanatmr-TMRInitTach4-LLR-005

Requirement ID: H398-LLD-ANA-FNC-2103

The function shall call GpioInit with parameters as reference to GPIOC init structure and M\_GPIOC to Configure tach 4 input pin to the value given below:

* Set gpio pin to M\_GPIOC\_TACH\_4
* Set gpio mode to GPIO\_MODE\_AF
* Set gpio speed to GPIO\_SPEED\_50MHZ
* Set gpio pupd to GPIO\_PUPD\_NOPULL

##### 10.18.8.7.6 dauanatmr-TMRInitTach4-LLR-006

Requirement ID: H398-LLD-ANA-FNC-2104

The function shall call DmaDeInit with parameter M\_DMA2\_STREAM7 to Reset M\_DMA2, stream 7 to original values.

##### 10.18.8.7.7 dauanatmr-TMRInitTach4-LLR-007

Requirement ID: H398-LLD-ANA-FNC-2105

The function shall call DmaInit with parameter as reference to dma init structure and M\_DMA2\_STREAM7 to Configure M\_DMA2, stream 7, channel 7 to the value given below:

* Set Dma channel to M\_DMA\_CHANNEL\_7
* Set Dma peripheral baseaddr to address of ccr4 of M\_TIM8
* Set Dma memory0 baseaddr to pu16BufferDMA.
* Set Dma dir to M\_DMA\_DIR\_PERIPHERAL\_TO\_MEMORY
* Set Dma buffersize to u16BufferSize.
* Set Dma peripheral inc to M\_DMA\_PERIPHERALINC\_DISABLE
* Set Dma memory inc to M\_DMA\_MEMORYINC\_ENABLE
* Set Dma peripheral datasize to M\_DMA\_PERIPH\_DATASIZE\_HALFWORD
* Set Dma memory datasize to M\_DMA\_MEMORY\_DATASIZE\_HALFWORD
* Set Dma mode to M\_DMA\_MODE\_CIRCULAR
* Set Dma priority to M\_DMA\_PRIORITY\_HIGH
* Set Dma fifo mode to M\_DMA\_FIFOMODE\_DISABLE
* Set Dma fifo threshold to M\_DMA\_FIFOTHRESHOLD\_FULL
* Set Dma memory burst to M\_DMA\_MEMORYBURST\_SINGLE

- Set Dma peripheral burst to M\_DMA\_PERIPHERALBURST\_SINGLE

##### 10.18.8.7.8 dauanatmr-TMRInitTach4-LLR-008

Requirement ID: H398-LLD-ANA-FNC-2106

The function shall call DmaCmd with parameters M\_DMA2\_STREAM7 and ENABLE to Enable M\_DMA2 stream 7.

##### 10.18.8.7.9 dauanatmr-TMRInitTach4-LLR-009

Requirement ID: H398-LLD-ANA-FNC-2107

The function shall call TimICInit with parameters as reference to tim icinit structure and M\_TIM8 to Configure channel 4 input of timer 8 to the value given below:

* Set Tim channel to M\_TIM\_CHANNEL\_4
* Set Tim ic polarity to M\_TIM\_ICPOLARITY\_RISING
* Set Tim ic selection to M\_TIM\_ICSELECTION\_DIRECTTI
* Set Tim ic prescaler by calling function GetDivConfig with parameter u8FreqDiv to set the frequency to Capture Prescaler
* Set Tim ic filter to M\_TIM\_ICFILTER\_15

##### 10.18.8.7.10 dauanatmr-TMRInitTach4-LLR-010

Requirement ID: H398-LLD-ANA-FNC-2108

The function shall call TimDmaCmd with parameters M\_TIM8, M\_TIM\_DMA\_CC4 and ENABLE to Enable DMA trigger for channel 4.

##### 10.18.8.7.11 dauanatmr-TMRInitTach4-LLR-011

Requirement ID: H398-LLD-ANA-FNC-2109

The function shall call TimPrescalerConfig with parameters M\_TIM8, M\_TMR\_8\_SCALE minus M\_ONE, and M\_TIM\_PSCRELOADMODE\_UPDATE to Set pre-scale for internal timer 8.

##### 10.18.8.7.12 dauanatmr-TMRInitTach4-LLR-012

Requirement ID: H398-LLD-ANA-FNC-2110

The function shall call TimCmd with parameters M\_TIM8 and ENABLE to Enable timer 8 to start the counter.

### 10.18.9 TMRInitTach5

Low Level Design Details about CSU TMRInitTach5 will follow in the sub sections.

#### 10.18.9.1 Brief Description

The function TMRInitTach5 Initialize timer 4 channel 1 for tach 5, DMA channel and Tach 5 GPIO pins for communication.

#### 10.18.9.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.18.9.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 10.18.9.4 Parameter list (Input/Output)

Inputs: T\_UINT8 u8FreqDiv - Frequency divider of the Tach channel

T\_UINT16 \*pu16BufferDMA - Reference to the DMA buffer

T\_UINT16 u16BufferSize - Size of the buffer

Outputs: None

#### 10.18.9.5 Return Value

None

#### 10.18.9.6 Other CSUs called by this CSU

RccAhb1PeriphClockCmd

RccApb2PeriphClockCmd

GpioPinAFConfig

GpioInit

DmaDeInit

DmaInit

DmaCmd

TimICInit

TimDmaCmd

TimPrescalerConfig

TimCmd

#### 10.18.9.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to TMRInitTach5.

##### 10.18.9.7.1 dauanatmr-TMRInitTach5-LLR-001

Requirement ID: H398-LLD-ANA-FNC-2119

The function shall call RccAhb1PeriphClockCmd with parameters M\_RCC\_AHB1PERIPH\_DMA1 and ENABLE to Enable DMA1 clock.

##### 10.18.9.7.2 dauanatmr-TMRInitTach5-LLR-002

Requirement ID: H398-LLD-ANA-FNC-2120

The function shall call RccApb1PeriphClockCmd with parameters M\_RCC\_APB1PERIPH\_TIM4 and ENABLE to Enable TIM4 clock.

##### 10.18.9.7.3 dauanatmr-TMRInitTach5-LLR-003

Requirement ID: H398-LLD-ANA-FNC-2121

The function shall call RccAhb1PeriphClockCmd with parameters M\_RCC\_AHB1PERIPH\_GPIOB and ENABLE to Enable GPIOB clock.

##### 10.18.9.7.4 dauanatmr-TMRInitTach5-LLR-004

Requirement ID: H398-LLD-ANA-FNC-2122

The function shall call GpioPinAFConfig with parameters M\_GPIOB, M\_GPIO\_PINSOURCE6 and M\_GPIO\_AF\_TIM4 to Configure pin function for tach 5.

##### 10.18.9.7.5 dauanatmr-TMRInitTach5-LLR-005

Requirement ID: H398-LLD-ANA-FNC-2123

The function shall call GpioInit with parameters as reference to GPIOB init structure and M\_GPIOB to Configure tach 5 input pin to the value given below:

* Set gpio pin to M\_GPIOB\_TACH\_5
* Set gpio mode to GPIO\_MODE\_AF
* Set gpio speed to GPIO\_SPEED\_50MHZ
* Set gpio pupd to GPIO\_PUPD\_NOPULL

##### 10.18.9.7.6 dauanatmr-TMRInitTach5-LLR-006

Requirement ID: H398-LLD-ANA-FNC-2124

The function shall call DmaDeInit with parameter M\_DMA1\_STREAM0 to Reset M\_DMA1, stream 0 to original values.

##### 10.18.9.7.7 dauanatmr-TMRInitTach5-LLR-007

Requirement ID: H398-LLD-ANA-FNC-2125

The function shall call DmaInit with parameter as reference to dma init structure and M\_DMA1\_STREAM0 to Configure DMA1, stream 0, channel 2 to the value given below:

* Set Dma channel to M\_DMA\_CHANNEL\_2
* Set Dma peripheral baseaddr to address of ccr1 of M\_TIM4
* Set Dma memory0 baseaddr to pu16BufferDMA.
* Set Dma dir to M\_DMA\_DIR\_PERIPHERAL\_TO\_MEMORY
* Set Dma buffersize to u16BufferSize.
* Set Dma peripheral inc to M\_DMA\_PERIPHERALINC\_DISABLE
* Set Dma memory inc to M\_DMA\_MEMORYINC\_ENABLE
* Set Dma peripheral datasize to M\_DMA\_PERIPH\_DATASIZE\_HALFWORD
* Set Dma memory datasize to M\_DMA\_MEMORY\_DATASIZE\_HALFWORD
* Set Dma mode to M\_DMA\_MODE\_CIRCULAR
* Set Dma priority to M\_DMA\_PRIORITY\_HIGH
* Set Dma fifo mode to M\_DMA\_FIFOMODE\_DISABLE
* Set Dma fifo threshold to M\_DMA\_FIFOTHRESHOLD\_FULL
* Set Dma memory burst to M\_DMA\_MEMORYBURST\_SINGLE

- Set Dma peripheral burst to M\_DMA\_PERIPHERALBURST\_SINGLE

##### 10.18.9.7.8 dauanatmr-TMRInitTach5-LLR-008

Requirement ID: H398-LLD-ANA-FNC-2126

The function shall call DmaCmd with parameters M\_DMA1\_STREAM0 and ENABLE to Enable M\_DMA1 stream 0.

##### 10.18.9.7.9 dauanatmr-TMRInitTach5-LLR-009

Requirement ID: H398-LLD-ANA-FNC-2127

The function shall call TimICInit with parameters as reference to tim icinit structure and M\_TIM4 to Configure channel 1 input of timer 4 to the value given below:

* Set Tim channel to M\_TIM\_CHANNEL\_1
* Set Tim ic polarity to M\_TIM\_ICPOLARITY\_RISING
* Set Tim ic selection to M\_TIM\_ICSELECTION\_DIRECTTI
* Set Tim ic prescaler by calling function GetDivConfig with parameter u8FreqDiv to set the frequency to Capture Prescaler
* Set Tim ic filter to M\_TIM\_ICFILTER\_15

##### 10.18.9.7.10 dauanatmr-TMRInitTach5-LLR-010

Requirement ID: H398-LLD-ANA-FNC-2128

The function shall call TimDmaCmd with parameters M\_TIM4, M\_TIM\_DMA\_CC1 and ENABLE to Enable DMA trigger for channel 1.

##### 10.18.9.7.11 dauanatmr-TMRInitTach5-LLR-011

Requirement ID: H398-LLD-ANA-FNC-2129

The function shall Call TimPrescalerConfig with parameters M\_TIM4, M\_TMR\_4\_SCALE minus M\_ONE, and M\_TIM\_PSCRELOADMODE\_UPDATE to Set pre-scale for internal timer 4.

##### 10.18.9.7.12 dauanatmr-TMRInitTach5-LLR-012

Requirement ID: H398-LLD-ANA-FNC-2130

The function shall Call TimCmd with parameters M\_TIM4 and ENABLE to Enable timer 4 to start the counter.

### 10.18.10 TMRInitTach6

Low Level Design Details about CSU TMRInitTach6 will follow in the sub sections.

#### 10.18.10.1 Brief Description

The function TMRInitTach5 Initialize timer 4 channel 2 for tach 6, DMA channel and Tach 6 GPIO pins for communication.

#### 10.18.10.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.18.10.3 3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 10.18.10.4 Parameter list (Input/Output)

Inputs: T\_UINT8 u8FreqDiv - Frequency divider of the Tach channel

T\_UINT16 \*pu16BufferDMA - Reference to the DMA buffer

T\_UINT16 u16BufferSize - Size of the buffer

Outputs: None

#### 10.18.10.5 Return Value

None

#### 10.18.10.6 Other CSUs called by this CSU

RccAhb1PeriphClockCmd

RccApb2PeriphClockCmd

GpioPinAFConfig

GpioInit

DmaDeInit

DmaInit

DmaCmd

TimICInit

TimDmaCmd

TimPrescalerConfig

TimCmd

#### 10.18.10.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to TMRInitTach6.

##### 10.18.10.7.1 dauanatmr-TMRInitTach6-LLR-001

Requirement ID: H398-LLD-ANA-FNC-2139

The function shall call RccAhb1PeriphClockCmd with parameters M\_RCC\_AHB1PERIPH\_DMA1 and ENABLE to Enable M\_DMA1 clock.

##### 10.18.10.7.2 dauanatmr-TMRInitTach6-LLR-002

Requirement ID: H398-LLD-ANA-FNC-2140

The function shall call RccApb1PeriphClockCmd with parameters M\_RCC\_APB1PERIPH\_TIM4 and ENABLE to Enable M\_TIM4 clock.

##### 10.18.10.7.3 dauanatmr-TMRInitTach6-LLR-003

Requirement ID: H398-LLD-ANA-FNC-2141

The function shall call RccAhb1PeriphClockCmd with parameters M\_RCC\_AHB1PERIPH\_GPIOB and ENABLE to Enable M\_GPIOB clock.

##### 10.18.10.7.4 dauanatmr-TMRInitTach6-LLR-004

Requirement ID: H398-LLD-ANA-FNC-2142

The function shall call GpioPinAFConfig with parameters M\_GPIOB, M\_GPIO\_PINSOURCE7 and M\_GPIO\_AF\_TIM4 to Configure pin function for tach 6.

##### 10.18.10.7.5 dauanatmr-TMRInitTach6-LLR-005

Requirement ID: H398-LLD-ANA-FNC-2143

The function shall call GpioInit with parameters as reference to GPIOB init structure and M\_GPIOB to Configure tach 6 input pin to the value given below:

* Set gpio pin to M\_GPIOB\_TACH\_6
* Set gpio mode to GPIO\_MODE\_AF
* Set gpio speed to GPIO\_SPEED\_50MHZ
* Set gpio pupd to GPIO\_PUPD\_NOPULL

##### 10.18.10.7.6 dauanatmr-TMRInitTach6-LLR-006

Requirement ID: H398-LLD-ANA-FNC-2144

The function shall call DmaDeInit with parameter M\_DMA1\_STREAM3 to Reset M\_DMA1, stream 3 to original values.

##### 10.18.10.7.7 dauanatmr-TMRInitTach6-LLR-007

Requirement ID: H398-LLD-ANA-FNC-2145

The function shall call DmaInit with parameter as reference to dma init structure and M\_DMA1\_STREAM3 to Configure M\_DMA1, stream 3, channel 2 to the value given below:

* Set Dma channel to M\_DMA\_CHANNEL\_2
* Set Dma peripheral baseaddr to address of ccr2 of M\_TIM4
* Set Dma memory0 baseaddr to pu16BufferDMA.
* Set Dma dir to M\_DMA\_DIR\_PERIPHERAL\_TO\_MEMORY
* Set Dma buffersize to u16BufferSize.
* Set Dma peripheral inc to M\_DMA\_PERIPHERALINC\_DISABLE
* Set Dma memory inc to M\_DMA\_MEMORYINC\_ENABLE
* Set Dma peripheral datasize to M\_DMA\_PERIPH\_DATASIZE\_HALFWORD
* Set Dma memory datasize to M\_DMA\_MEMORY\_DATASIZE\_HALFWORD
* Set Dma mode to M\_DMA\_MODE\_CIRCULAR
* Set Dma priority to M\_DMA\_PRIORITY\_HIGH
* Set Dma fifo mode to M\_DMA\_FIFOMODE\_DISABLE
* Set Dma fifo threshold to M\_DMA\_FIFOTHRESHOLD\_FULL
* Set Dma memory burst to M\_DMA\_MEMORYBURST\_SINGLE

- Set Dma peripheral burst to M\_DMA\_PERIPHERALBURST\_SINGLE

##### 10.18.10.7.8 dauanatmr-TMRInitTach6-LLR-008

Requirement ID: H398-LLD-ANA-FNC-2146

The function shall call DmaCmd with parameters M\_DMA1\_STREAM3 and ENABLE to Enable M\_DMA1 stream 3.

##### 10.18.10.7.9 dauanatmr-TMRInitTach6-LLR-009

Requirement ID: H398-LLD-ANA-FNC-2147

The function shall call TimICInit with parameters as reference to tim icinit structure and M\_TIM4 to Configure channel 2 input of timer 4 to the value given below:

* Set Tim channel to M\_TIM\_CHANNEL\_2
* Set Tim ic polarity to M\_TIM\_ICPOLARITY\_RISING
* Set Tim ic selection to M\_TIM\_ICSELECTION\_DIRECTTI
* Set Tim ic prescaler by calling function GetDivConfig with parameter u8FreqDiv to set the frequency to Capture Prescaler
* Set Tim ic filter to M\_TIM\_ICFILTER\_0

##### 10.18.10.7.10 dauanatmr-TMRInitTach6-LLR-010

Requirement ID: H398-LLD-ANA-FNC-2148

The function shall call TimDmaCmd with parameters M\_TIM4, M\_TIM\_DMA\_CC2 and ENABLE to Enable DMA trigger for channel 2.

##### 10.18.10.7.11 dauanatmr-TMRInitTach6-LLR-011

Requirement ID: H398-LLD-ANA-FNC-2149

The function shall call TimPrescalerConfig with parameters M\_TIM4, M\_TMR\_4\_SCALE minus M\_ONE, and M\_TIM\_PSCRELOADMODE\_UPDATE to Set pre-scale for internal timer 4.

##### 10.18.10.7.12 dauanatmr-TMRInitTach6-LLR-012

Requirement ID: H398-LLD-ANA-FNC-2150

The function shall call TimCmd with parameters M\_TIM4 and ENABLE to Enable timer 4 to start the counter.

### 10.18.11 TMRInitPwmc1

Low Level Design Details about CSU TMRInitPwmc1 will follow in the sub sections.

#### 10.18.11.1 Brief Description

This function initializes timer 5 channel 1 for tach 2, DMA channel and Tach 2 GPIO pins for communication.

#### 10.18.11.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.18.11.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 10.18.11.4 Parameter list (Input/Output)

Inputs: T\_UINT8 u8FreqDiv - Frequency divider of the Tach channel

T\_UINT16 \*pu16BufferDMA - Reference to the DMA buffer

T\_UINT16 u16BufferSize - Size of the buffer

Outputs: None

#### 10.18.11.5 Return Value

None

#### 10.18.11.6 Other CSUs called by this CSU

RccAhb1PeriphClockCmd

RccApb2PeriphClockCmd

GpioPinAFConfig

GpioInit

DmaDeInit

DmaInit

DmaCmd

TimICInit

TimDmaCmd

TimPrescalerConfig

TimCmd

#### 10.18.11.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to TMRInitPwmc1.

##### 10.18.11.7.1 dauanatmr-TMRInitPwmc1-LLR-001

Requirement ID: H398-LLD-ANA-FNC-2159

The function shall call RccAhb1PeriphClockCmd with parameters M\_RCC\_AHB1PERIPH\_DMA1 and ENABLE to Enable DMA1 clock.

##### 10.18.11.7.2 dauanatmr-TMRInitPwmc1-LLR-002

Requirement ID: H398-LLD-ANA-FNC-2160

The function shall call RccApb1PeriphClockCmd with parameters M\_RCC\_APB1PERIPH\_TIM5 and ENABLE to Enable M\_TIM5 clock.

##### 10.18.11.7.3 dauanatmr-TMRInitPwmc1-LLR-003

Requirement ID: H398-LLD-ANA-FNC-2161

The function shall call RccAhb1PeriphClockCmd with parameters M\_RCC\_AHB1PERIPH\_GPIOA and ENABLE to Enable M\_GPIOA clock.

##### 10.18.11.7.4 dauanatmr-TMRInitPwmc1-LLR-004

Requirement ID: H398-LLD-ANA-FNC-2162

The function shall call GpioPinAFConfig with parameters M\_GPIOA, M\_GPIO\_PINSOURCE0 and M\_GPIO\_AF\_TIM5 to Configure pin function for tach 2.

##### 10.18.11.7.5 dauanatmr-TMRInitPwmc1-LLR-005

Requirement ID: H398-LLD-ANA-FNC-2163

The function shall call GpioInit with parameters as reference to GPIOA init structure and M\_GPIOA to Configure tach 2 input pin to the value given below:

* Set gpio pin to M\_GPIOA\_PWMC1\_NE
* Set gpio mode to GPIO\_MODE\_AF
* Set gpio speed to GPIO\_SPEED\_50MHZ
* Set gpio pupd to GPIO\_PUPD\_NOPULL

##### 10.18.11.7.6 dauanatmr-TMRInitPwmc1-LLR-006

Requirement ID: H398-LLD-ANA-FNC-2164

The function shall call DmaDeInit with parameter M\_DMA1\_STREAM2 to Reset M\_DMA1, stream 2 to original values.

##### 10.18.11.7.7 dauanatmr-TMRInitPwmc1-LLR-007

Requirement ID: H398-LLD-ANA-FNC-2165

The function shall call DmaInit with parameter as reference to dma init structure and M\_DMA1\_STREAM2 to Configure M\_DMA1, stream 2, channel 6 to the value given below:

* Set Dma channel to M\_DMA\_CHANNEL\_6
* Set Dma peripheral baseaddr to address of ccr1 of M\_TIM5
* Set Dma memory0 baseaddr to pu16BufferDMA.
* Set Dma dir to M\_DMA\_DIR\_PERIPHERAL\_TO\_MEMORY
* Set Dma buffersize to u16BufferSize.
* Set Dma peripheral inc to M\_DMA\_PERIPHERALINC\_DISABLE
* Set Dma memory inc to M\_DMA\_MEMORYINC\_ENABLE
* Set Dma peripheral datasize to M\_DMA\_PERIPH\_DATASIZE\_HALFWORD
* Set Dma memory datasize to M\_DMA\_MEMORY\_DATASIZE\_HALFWORD
* Set Dma mode to M\_DMA\_MODE\_CIRCULAR
* Set Dma priority to M\_DMA\_PRIORITY\_HIGH
* Set Dma fifo mode to M\_DMA\_FIFOMODE\_DISABLE
* Set Dma fifo threshold to M\_DMA\_FIFOTHRESHOLD\_FULL
* Set Dma memory burst to M\_DMA\_MEMORYBURST\_SINGLE

- Set Dma peripheral burst to M\_DMA\_PERIPHERALBURST\_SINGLE

##### 10.18.11.7.8 dauanatmr-TMRInitPwmc1-LLR-008

Requirement ID: H398-LLD-ANA-FNC-2166

The function shall call DmaCmd with parameters M\_DMA1\_STREAM2 and ENABLE to Enable M\_DMA1 stream 2.

##### 10.18.11.7.9 dauanatmr-TMRInitPwmc1-LLR-009

Requirement ID: H398-LLD-ANA-FNC-2167

The function shall call TimICInit with parameters as reference to tim icinit structure and M\_TIM5 to Configure channel 1 input of timer 5 to the value given below:

* Set Tim channel to M\_TIM\_CHANNEL\_1
* Set Tim ic polarity to M\_TIM\_ICPOLARITY\_RISING
* Set Tim ic selection to M\_TIM\_ICSELECTION\_DIRECTTI
* Set Tim ic prescaler by calling function GetDivConfig with parameter u8FreqDiv to set the frequency to Capture Prescaler
* Set Tim ic filter to M\_TIM\_ICFILTER\_15

##### 10.18.11.7.10 dauanatmr-TMRInitPwmc1-LLR-010

Requirement ID: H398-LLD-ANA-FNC-2168

The function shall call TimDmaCmd with parameters M\_TIM5, M\_TIM\_DMA\_CC1 and ENABLE to Enable DMA trigger for channel 1.

##### 10.18.11.7.11 dauanatmr-TMRInitPwmc1-LLR-011

Requirement ID: H398-LLD-ANA-FNC-2169

The function shall call TimPrescalerConfig with parameters M\_TIM5, M\_TMR\_5\_SCALE minus M\_ONE, and M\_TIM\_PSCRELOADMODE\_IMMEDIATE to Set pre-scale for internal timer 5.

##### 10.18.11.7.12 dauanatmr-TMRInitPwmc1-LLR-012

Requirement ID: H398-LLD-ANA-FNC-2170

The function shall call TimCmd with parameters M\_TIM5 and ENABLE to Enable timer 5 to start the counter.

### 10.18.12 TMRInitPwmc2

Low Level Design Details about CSU TMRInitPwmc2 will follow in the sub sections.

#### 10.18.12.1 Brief Description

This function initializes timer 5 channel 2 for tach 3,DMA channel and Tach 3 GPIO pins for communication.

#### 10.18.12.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.18.12.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 10.18.12.4 Parameter list (Input/Output)

Inputs: T\_UINT8 u8FreqDiv - Frequency divider of the Tach channel

T\_UINT16 \*pu16BufferDMA - Reference to the DMA buffer

T\_UINT16 u16BufferSize - Size of the buffer

Outputs: None

#### 10.18.12.5 Return Value

None

#### 10.18.12.6 Other CSUs called by this CSU

RccAhb1PeriphClockCmd

RccApb2PeriphClockCmd

GpioPinAFConfig

GpioInit

DmaDeInit

DmaInit

DmaCmd

TimICInit

TimDmaCmd

TimPrescalerConfig

TimCmd

#### 10.18.12.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to TMRInitPwmc2.

##### 10.18.12.7.1 dauanatmr-TMRInitPwmc2-LLR-001

Requirement ID: H398-LLD-ANA-FNC-2179

The function shall call RccAhb1PeriphClockCmd with parameters M\_RCC\_AHB1PERIPH\_DMA1 and ENABLE to Enable DMA1 clock.

##### 10.18.12.7.2 dauanatmr-TMRInitPwmc2-LLR-002

Requirement ID: H398-LLD-ANA-FNC-2180

The function shall call RccApb1PeriphClockCmd with parameters M\_RCC\_APB1PERIPH\_TIM5 and ENABLE to Enable M\_TIM5 clock.

##### 10.18.12.7.3 dauanatmr-TMRInitPwmc2-LLR-003

Requirement ID: H398-LLD-ANA-FNC-2181

The function shall call RccAhb1PeriphClockCmd with parameters M\_RCC\_AHB1PERIPH\_GPIOA and ENABLE to Enable M\_GPIOA clock.

##### 10.18.12.7.4 dauanatmr-TMRInitPwmc2-LLR-004

Requirement ID: H398-LLD-ANA-FNC-2182

The function shall call GpioPinAFConfig with parameters M\_GPIOA, M\_GPIO\_PINSOURCE1 and M\_GPIO\_AF\_TIM5 to Configure pin function for tach 3.

##### 10.18.12.7.5 dauanatmr-TMRInitPwmc2-LLR-005

Requirement ID: H398-LLD-ANA-FNC-2183

The function shall call GpioInit with parameters as reference to GPIOA init structure and M\_GPIOA to Configure tach 3 input pin to the value given below:

* Set gpio pin to M\_GPIOA\_PWMC1\_PE
* Set gpio mode to GPIO\_MODE\_AF
* Set gpio speed to GPIO\_SPEED\_50MHZ
* Set gpio pupd to GPIO\_PUPD\_NOPULL

##### 10.18.12.7.6 dauanatmr-TMRInitPwmc2-LLR-006

Requirement ID: H398-LLD-ANA-FNC-2184

The function shall call DmaDeInit with parameter M\_DMA1\_STREAM4 to Reset M\_DMA1, stream 4 to original values.

##### 10.18.12.7.7 dauanatmr-TMRInitPwmc2-LLR-007

Requirement ID: H398-LLD-ANA-FNC-2185

The function shall call DmaInit with parameter as reference to dma init structure and M\_DMA1\_STREAM4 to Configure M\_DMA1, stream 4, channel 6 to the value given below:

* Set Dma channel to M\_DMA\_CHANNEL\_6
* Set Dma peripheral baseaddr to address of ccr2 of M\_TIM5
* Set Dma memory0 baseaddr to pu16BufferDMA.
* Set Dma dir to M\_DMA\_DIR\_PERIPHERAL\_TO\_MEMORY
* Set Dma buffersize to u16BufferSize.
* Set Dma peripheral inc to M\_DMA\_PERIPHERALINC\_DISABLE
* Set Dma memory inc to M\_DMA\_MEMORYINC\_ENABLE
* Set Dma peripheral datasize to M\_DMA\_PERIPH\_DATASIZE\_HALFWORD
* Set Dma memory datasize to M\_DMA\_MEMORY\_DATASIZE\_HALFWORD
* Set Dma mode to M\_DMA\_MODE\_CIRCULAR
* Set Dma priority to M\_DMA\_PRIORITY\_HIGH
* Set Dma fifo mode to M\_DMA\_FIFOMODE\_DISABLE
* Set Dma fifo threshold to M\_DMA\_FIFOTHRESHOLD\_FULL
* Set Dma memory burst to M\_DMA\_MEMORYBURST\_SINGLE

- Set Dma peripheral burst to M\_DMA\_PERIPHERALBURST\_SINGLE

##### 10.18.12.7.8 dauanatmr-TMRInitPwmc2-LLR-008

Requirement ID: H398-LLD-ANA-FNC-2186

The function shall call DmaCmd with parameters M\_DMA1\_STREAM4 and ENABLE to Enable M\_DMA1 stream 4.

##### 10.18.12.7.9 dauanatmr-TMRInitPwmc2-LLR-009

Requirement ID: H398-LLD-ANA-FNC-2187

The function shall call TimICInit with parameters as reference to tim icinit structure and M\_TIM5 to Configure channel 2 input of timer 5 to the value given below:

* Set Tim channel to M\_TIM\_CHANNEL\_2
* Set Tim ic polarity to M\_TIM\_ICPOLARITY\_FALLING
* Set Tim ic selection to M\_TIM\_ICSELECTION\_DIRECTTI
* Set Tim ic prescaler by calling function GetDivConfig with parameter u8FreqDiv to set the frequency to Capture Prescaler
* Set Tim ic filter to M\_TIM\_ICFILTER\_15

##### 10.18.12.7.10 dauanatmr-TMRInitPwmc2-LLR-010

Requirement ID: H398-LLD-ANA-FNC-2188

The function shall call TimDmaCmd with parameters M\_TIM5, M\_TIM\_DMA\_CC2 and ENABLE to Enable DMA trigger for channel 2.

##### 10.18.12.7.11 dauanatmr-TMRInitPwmc2-LLR-011

Requirement ID: H398-LLD-ANA-FNC-2189

The function shall call TimPrescalerConfig with parameters M\_TIM5, M\_TMR\_5\_SCALE minus M\_ONE, and M\_TIM\_PSCRELOADMODE\_IMMEDIATE to Set pre-scale for internal timer 5.

##### 10.18.12.7.12 dauanatmr-TMRInitPwmc2-LLR-012

Requirement ID: H398-LLD-ANA-FNC-2190

The function shall call TimCmd with parameters M\_TIM5 and ENABLE to Enable timer 5 to start the counter.

### 10.18.13 TMRInitPwmc3

Low Level Design Details about CSU TMRInitPwmc3 will follow in the sub sections.

#### 10.18.13.1 Brief Description

This function initializes timer 5 channel 2 for tach 4,DMA channel and Tach 4 GPIO pins for communication.

#### 10.18.13.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.18.13.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 10.18.13.4 Parameter list (Input/Output)

Inputs: T\_UINT8 u8FreqDiv - Frequency divider of the Tach channel

T\_UINT16 \*pu16BufferDMA - Reference to the DMA buffer

T\_UINT16 u16BufferSize - Size of the buffer

Outputs: None

#### 10.18.13.5 Return Value

None

#### 10.18.13.6 Other CSUs called by this CSU

RccAhb1PeriphClockCmd

RccApb2PeriphClockCmd

GpioPinAFConfig

GpioInit

DmaDeInit

DmaInit

DmaCmd

TimICInit

TimDmaCmd

TimPrescalerConfig

TimCmd

#### 10.18.13.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to TMRInitPwmc3.

##### 10.18.13.7.1 dauanatmr-TMRInitPwmc3-LLR-001

Requirement ID: H398-LLD-ANA-FNC-2199

The function shall call RccAhb1PeriphClockCmd with parameters M\_RCC\_AHB1PERIPH\_DMA1 and ENABLE to Enable DMA1 clock.

##### 10.18.13.7.2 dauanatmr-TMRInitPwmc3-LLR-002

Requirement ID: H398-LLD-ANA-FNC-2200

The function shall call RccApb1PeriphClockCmd with parameters M\_RCC\_APB1PERIPH\_TIM2 and ENABLE to Enable M\_TIM2 clock.

##### 10.18.13.7.3 dauanatmr-TMRInitPwmc3-LLR-003

Requirement ID: H398-LLD-ANA-FNC-2201

The function shall call RccAhb1PeriphClockCmd with parameters M\_RCC\_AHB1PERIPH\_GPIOA and ENABLE to Enable M\_GPIOA clock.

##### 10.18.13.7.4 dauanatmr-TMRInitPwmc3-LLR-004

Requirement ID: H398-LLD-ANA-FNC-2202

The function shall call GpioPinAFConfig with parameters M\_GPIOA, M\_GPIO\_PINSOURCE2 and M\_GPIO\_AF\_TIM2 to Configure pin function for tach 4.

##### 10.18.13.7.5 dauanatmr-TMRInitPwmc3-LLR-005

Requirement ID: H398-LLD-ANA-FNC-2203

The function shall call GpioInit with parameters as reference to GPIOA init structure and M\_GPIOA to Configure tach 4 input pin to the value given below:

* Set gpio pin to M\_GPIOA\_PWMC2\_NE
* Set gpio mode to GPIO\_MODE\_AF
* Set gpio speed to GPIO\_SPEED\_50MHZ
* Set gpio pupd to GPIO\_PUPD\_NOPULL

##### 10.18.13.7.6 dauanatmr-TMRInitPwmc3-LLR-006

Requirement ID: H398-LLD-ANA-FNC-2204

The function shall call DmaDeInit with parameter M\_DMA1\_STREAM1 to Reset M\_DMA1, stream 1 to original values.

##### 10.18.13.7.7 dauanatmr-TMRInitPwmc3-LLR-007

Requirement ID: H398-LLD-ANA-FNC-2205

The function shall call DmaInit with parameter as reference to dma init structure and M\_DMA1\_STREAM1 to Configure M\_DMA1, stream 1, channel 3 to the value given below:

* Set Dma channel to M\_DMA\_CHANNEL\_3
* Set Dma peripheral baseaddr to address of ccr3 of M\_TIM2
* Set Dma memory0 baseaddr to pu16BufferDMA.
* Set Dma dir to M\_DMA\_DIR\_PERIPHERAL\_TO\_MEMORY
* Set Dma buffersize to u16BufferSize.
* Set Dma peripheral inc to M\_DMA\_PERIPHERALINC\_DISABLE
* Set Dma memory inc to M\_DMA\_MEMORYINC\_ENABLE
* Set Dma peripheral datasize to M\_DMA\_PERIPH\_DATASIZE\_HALFWORD
* Set Dma memory datasize to M\_DMA\_MEMORY\_DATASIZE\_HALFWORD
* Set Dma mode to M\_DMA\_MODE\_CIRCULAR
* Set Dma priority to M\_DMA\_PRIORITY\_HIGH
* Set Dma fifo mode to M\_DMA\_FIFOMODE\_DISABLE
* Set Dma fifo threshold to M\_DMA\_FIFOTHRESHOLD\_FULL
* Set Dma memory burst to M\_DMA\_MEMORYBURST\_SINGLE

- Set Dma peripheral burst to M\_DMA\_PERIPHERALBURST\_SINGLE

##### 10.18.13.7.8 dauanatmr-TMRInitPwmc3-LLR-008

Requirement ID: H398-LLD-ANA-FNC-2206

The function shall call DmaCmd with parameters M\_DMA1\_STREAM1 and ENABLE to Enable M\_DMA1 stream 1.

##### 10.18.13.7.9 dauanatmr-TMRInitPwmc3-LLR-009

Requirement ID: H398-LLD-ANA-FNC-2207

The function shall call TimICInit with parameters as reference to tim icinit structure and M\_TIM2 to Configure channel 3 input of timer 2 to the value given below:

* Set Tim channel to M\_TIM\_CHANNEL\_3
* Set Tim ic polarity to M\_TIM\_ICPOLARITY\_RISING
* Set Tim ic selection to M\_TIM\_ICSELECTION\_DIRECTTI
* Set Tim ic prescaler by calling function GetDivConfig with parameter u8FreqDiv to set the frequency to Capture Prescaler
* Set Tim ic filter to M\_TIM\_ICFILTER\_15

##### 10.18.13.7.10 dauanatmr-TMRInitPwmc3-LLR-010

Requirement ID: H398-LLD-ANA-FNC-2208

The function shall call TimDmaCmd with parameters M\_TIM2, M\_TIM\_DMA\_CC3 and ENABLE to Enable DMA trigger for channel 3.

##### 10.18.13.7.11 dauanatmr-TMRInitPwmc3-LLR-011

Requirement ID: H398-LLD-ANA-FNC-2209

The function shall call TimPrescalerConfig with parameters M\_TIM2, M\_TMR\_5\_SCALE minus M\_ONE, and M\_TIM\_PSCRELOADMODE\_IMMEDIATE to Set pre-scale for internal timer 2.

##### 10.18.13.7.12 dauanatmr-TMRInitPwmc3-LLR-012

Requirement ID: H398-LLD-ANA-FNC-2210

The function shall call TimCmd with parameters M\_TIM2 and ENABLE to Enable timer 2 to start the counter.

### 10.18.14 TMRInitPwmc4

Low Level Design Details about CSU TMRInitPwmc4 will follow in the sub sections.

#### 10.18.14.1 Brief Description

This function initializes timer 5 channel 2 for tach 5,DMA channel and Tach 3 GPIO pins for communication.

#### 10.18.14.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.18.14.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 10.18.14.4 Parameter list (Input/Output)

Inputs: T\_UINT8 u8FreqDiv - Frequency divider of the Tach channel

T\_UINT16 \*pu16BufferDMA - Reference to the DMA buffer

T\_UINT16 u16BufferSize - Size of the buffer

Outputs: None

#### 10.18.14.5 Return Value

None

#### 10.18.14.6 Other CSUs called by this CSU

RccAhb1PeriphClockCmd

RccApb2PeriphClockCmd

GpioPinAFConfig

GpioInit

DmaDeInit

DmaInit

DmaCmd

TimICInit

TimDmaCmd

TimPrescalerConfig

TimCmd

#### 10.18.14.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to TMRInitPwmc4.

##### 10.18.14.7.1 dauanatmr-TMRInitPwmc4-LLR-001

Requirement ID: H398-LLD-ANA-FNC-2219

The function shall call RccAhb1PeriphClockCmd with parameters M\_RCC\_AHB1PERIPH\_DMA1 and ENABLE to Enable DMA1 clock.

##### 10.18.14.7.2 dauanatmr-TMRInitPwmc4-LLR-002

Requirement ID: H398-LLD-ANA-FNC-2220

The function shall call RccApb1PeriphClockCmd with parameters M\_RCC\_APB1PERIPH\_TIM2 and ENABLE to Enable M\_TIM2 clock.

##### 10.18.14.7.3 dauanatmr-TMRInitPwmc4-LLR-003

Requirement ID: H398-LLD-ANA-FNC-2221

The function shall call RccAhb1PeriphClockCmd with parameters M\_RCC\_AHB1PERIPH\_GPIOA and ENABLE to Enable M\_GPIOA clock.

##### 10.18.14.7.4 dauanatmr-TMRInitPwmc4-LLR-004

Requirement ID: H398-LLD-ANA-FNC-2222

The function shall call GpioPinAFConfig with parameters M\_GPIOA, M\_GPIO\_PINSOURCE3 and M\_GPIO\_AF\_TIM2 to Configure pin function for tach 4.

##### 10.18.14.7.5 dauanatmr-TMRInitPwmc4-LLR-005

Requirement ID: H398-LLD-ANA-FNC-2223

The function shall call GpioInit with parameters as reference to GPIOA init structure and M\_GPIOA to Configure tach 4 input pin to the value given below:

* Set gpio pin to M\_GPIOA\_PWMC2\_PE
* Set gpio mode to GPIO\_MODE\_AF
* Set gpio speed to GPIO\_SPEED\_50MHZ
* Set gpio pupd to GPIO\_PUPD\_NOPULL

##### 10.18.14.7.6 dauanatmr-TMRInitPwmc4-LLR-006

Requirement ID: H398-LLD-ANA-FNC-2224

The function shall call DmaDeInit with parameter M\_DMA1\_STREAM7 to Reset M\_DMA1, stream 7 to original values.

##### 10.18.14.7.7 dauanatmr-TMRInitPwmc4-LLR-007

Requirement ID: H398-LLD-ANA-FNC-2225

The function shall call DmaInit with parameter as reference to dma init structure and M\_DMA1\_STREAM7 to Configure M\_DMA1, stream 7, channel 3 to the value given below:

* Set Dma channel to M\_DMA\_CHANNEL\_3
* Set Dma peripheral baseaddr to address of ccr4 of M\_TIM2
* Set Dma memory0 baseaddr to pu16BufferDMA.
* Set Dma dir to M\_DMA\_DIR\_PERIPHERAL\_TO\_MEMORY
* Set Dma buffersize to u16BufferSize.
* Set Dma peripheral inc to M\_DMA\_PERIPHERALINC\_DISABLE
* Set Dma memory inc to M\_DMA\_MEMORYINC\_ENABLE
* Set Dma peripheral datasize to M\_DMA\_PERIPH\_DATASIZE\_HALFWORD
* Set Dma memory datasize to M\_DMA\_MEMORY\_DATASIZE\_HALFWORD
* Set Dma mode to M\_DMA\_MODE\_CIRCULAR
* Set Dma priority to M\_DMA\_PRIORITY\_HIGH
* Set Dma fifo mode to M\_DMA\_FIFOMODE\_DISABLE
* Set Dma fifo threshold to M\_DMA\_FIFOTHRESHOLD\_FULL
* Set Dma memory burst to M\_DMA\_MEMORYBURST\_SINGLE

- Set Dma peripheral burst to M\_DMA\_PERIPHERALBURST\_SINGLE

##### 10.18.14.7.8 dauanatmr-TMRInitPwmc4-LLR-008

Requirement ID: H398-LLD-ANA-FNC-2226

The function shall call DmaCmd with parameters M\_DMA1\_STREAM7 and ENABLE to Enable M\_DMA1 stream 7.

##### 10.18.14.7.9 dauanatmr-TMRInitPwmc4-LLR-009

Requirement ID: H398-LLD-ANA-FNC-2227

The function shall call TimICInit with parameters as reference to tim icinit structure and M\_TIM2 to Configure channel 4 input of timer 2 to the value given below:

* Set Tim channel to M\_TIM\_CHANNEL\_4
* Set Tim ic polarity to M\_TIM\_ICPOLARITY\_FALLING
* Set Tim ic selection to M\_TIM\_ICSELECTION\_DIRECTTI
* Set Tim ic prescaler by calling function GetDivConfig with parameter u8FreqDiv to set the frequency to Capture Prescaler
* Set Tim ic filter to M\_TIM\_ICFILTER\_15

##### 10.18.14.7.10 dauanatmr-TMRInitPwmc4-LLR-010

Requirement ID: H398-LLD-ANA-FNC-2228

The function shall call TimDmaCmd with parameters M\_TIM2, M\_TIM\_DMA\_CC4 and ENABLE to Enable DMA trigger for channel 4.

##### 10.18.14.7.11 dauanatmr-TMRInitPwmc4-LLR-011

Requirement ID: H398-LLD-ANA-FNC-2229

The function shall call TimPrescalerConfig with parameters M\_TIM2, M\_TMR\_5\_SCALE minus M\_ONE, and M\_TIM\_PSCRELOADMODE\_IMMEDIATE to Set pre-scale for internal timer 5.

##### 10.18.14.7.12 dauanatmr-TMRInitPwmc4-LLR-012

Requirement ID: H398-LLD-ANA-FNC-2230

The function shall call TimCmd with parameters M\_TIM2 and ENABLE to Enable timer 2 to start the counter.

## 10.19 dauanaucos

This module defines the implementation of uC/OS routines for the real-time kernel.

### 10.19.1 OsInit

Low Level Design Details about CSU OsInit will follow in the sub sections.

#### 10.19.1.1 Brief Description

This is the uC/OS initialization function.

The OsInit function initializes the Os ready to run task list, TCB priority table list, list of free TCBs, list of free Event Control Blocks to default values. The function creates the OsTaskIdle function.

(Note: OsInit must be called before OsStart).

#### 10.19.1.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.19.1.3 List of global variables accessed and modified

Accessed : Os\_task\_idle\_stk

Modified : Os\_tcb\_high\_rdy,

Os\_tcb\_cur,

Os\_running,

Os\_tcb\_prio\_tbl

#### 10.19.1.4 Parameter list (Input/Output)

Inputs : None

Outputs : None

#### 10.19.1.5 Return Value

None

#### 10.19.1.6 Other CSUs called by this CSU

OsTaskCreate

OsTaskIdle

#### 10.19.1.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to OsInit.

##### 10.19.1.7.1 dauanaucos-OsInit-LLR-001

Requirement ID: H398-LLD-ANA-FNC-937

The function shall initialize the following:

- Current value of System Time to M\_ZERO

- Pointer to highest priority TCB ready to run (Os\_tcb\_high\_rdy) to M\_NULL.

- Pointer to currently running TCB (Os\_tcb\_cur) to M\_NULL.

- Pointer to doubly linked list of TCBs to M\_NULL

- Interrupt nesting level, Multitasking lock nesting level to M\_ZERO.

- Flag indicating that Kernel is running(Os\_running) to false.

- Idle counter(Os\_idle\_ctr) is set to M\_ZERO.

- Counter of number of context switches to M\_ZERO.

-Ready list group to M\_ZERO.

- All the indices (until index is M\_RDY\_LST\_SIZE) of the Os ready to run list to M\_ZERO.

- all the indices (until index is M\_MAX\_TCB\_PRIO) of the TCB priority table list to M\_NULL.

- os\_tcb\_next of each index of OS TCB list to the next element of OS TCB list (except the last element M\_OS\_MAX\_TASKS).

- os\_tcb\_next of last index of OS TCB list to M\_NULL.

- OS TCB free list to the base address of OS TCB list.

- Set the os\_eventptr of each index of 'free Event Control Blocks list' to the next element of free Event Control Blocks list (except the last element (M\_OS\_MAX\_EVENTS minus M\_ONE)).

- os\_eventptr of last index of free Event Control Blocks list to NULL.

- OS EVENT free list to the base address of free Event Control Blocks list.

##### 10.19.1.7.2 dauanaucos-OsInit-LLR-002

Requirement ID: H398-LLD-ANA-FNC-938

The function shall call ErrorHandler when M\_OS\_NO\_ERR is not equal to return value of OsTaskCreate with parameters

a) Pointer to function OsTaskIdle,

b) M\_NULL as task entry point,

c) Reference to top of stack of Os task idle stack (i.e. Os\_task\_idle\_stk)

d) M\_OS\_LOWEST\_PRIO as task priority.

otherwise do nothing

### 10.19.2 OsTaskIdle

Low Level Design Details about CSU OsTaskIdle will follow in the sub sections.

#### 10.19.2.1 Brief Description

The function OsTaskIdle keeps track of CPU idle time and reset the watch dog counter. This function is executed in the cpu when no other task is running in the cpu.

#### 10.19.2.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.19.2.3 List of global variables accessed and modified

Accessed : U32\_critical\_sr

Modified : U32\_critical\_sr

#### 10.19.2.4 Parameter list (Input/Output)

Inputs : void \*p\_data - Not used always executed with null reference

Outputs : None

#### 10.19.2.5 Return Value

None

#### 10.19.2.6 Other CSUs called by this CSU

WdogKickWatchDog

SaveStatusReg

RestoreStatusReg

#### 10.19.2.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to OsTaskIdle.

##### 10.19.2.7.1 dauanaucos-OsTaskIdle-LLR-001

Requirement ID: H398-LLD-ANA-FNC-947

The function shall do the following:

1. Set p\_data to M\_NULL.
2. infinitely loop and perform the following operations

- call ‘SaveStatusReg’ (M\_OS\_ENTER\_CRITICAL) to disable interrupts before incrementing the idle and return value gets stored in U32\_critical\_sr.

- Call WdogKickWatchDog to reset the watchdog timer.

- Call ‘RestoreStatusReg’ (M\_OS\_EXIT\_CRITICAL) with parameter U32\_critical\_sr to restore the priority mask register after the idle counter increment is completed.

### 10.19.3 OsStart

Low Level Design Details about CSU OsStart will follow in the sub sections.

#### 10.19.3.1 Brief Description

The function OsStart starts the multitasking process, allowing uC/OS to manage

the tasks that have been created.

NOTE:

a) Before OsStart () is called, OsInit () has to be called and at least one task has to be created.

b) OSStart sets Os\_tcb\_high\_rdy to point to the Os\_tcb of the highest priority task.

#### 10.19.3.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.19.3.3 List of global variables accessed and modified

Accessed : Os\_tcb\_high\_rdy, Os\_tcb\_prio\_tbl

Modified : Os\_tcb\_high\_rdy, Os\_tcb\_cur, Os\_running

#### 10.19.3.4 Parameter list (Input/Output)

Inputs : None

Outputs : None

#### 10.19.3.5 Return Value

None

#### 10.19.3.6 Other CSUs called by this CSU

OsStartHighRdy

#### 10.19.3.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to OsStart.

##### 10.19.3.7.1 dauanaucos-OsStart-LLR-001

Requirement ID: H398-LLD-ANA-FNC-956

The function shall

a) Sets second set of highest priority task priority number to priority resolution table with index as ready list group.

b) Sets first set of highest priority task priority number to priority resolution table with index as table of tasks which are ready to run.

c)Calculates priority value as ((second set of highest priority task priority number bit shifted to left by M\_THREE) added to the first set of highest priority task priority number)

d)Sets the Os\_tcb\_high\_rdy to point to the highest priority task ready to run from the TCB priority table (i.e. Os\_tcb\_prio\_tbl).

##### 10.19.3.7.2 dauanaucos-OsStart-LLR-002

Requirement ID: H398-LLD-ANA-FNC-957

The function shall set the pointer Os\_tcb\_cur (currently running TCB) to the highest priority TCB ready to run (Os\_tcb\_high\_rdy).

##### 10.19.3.7.3 dauanaucos-OsStart-LLR-003

Requirement ID: H398-LLD-ANA-FNC-958

The function shall set the Os\_running to TRUE.

##### 10.19.3.7.4 dauanaucos-OsStart-LLR-004

Requirement ID: H398-LLD-ANA-FNC-959

The function shall call the function 'OsStartHighRdy' to start running the highest priority task.

### 10.19.4 OsSched

Low Level Design Details about CSU OsSched will follow in the sub sections.

#### 10.19.4.1 Brief Description

The function OsSched does the Task-level scheduling.

#### 10.19.4.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.19.4.3 List of global variables accessed and modified

Accessed : Os\_tcb\_high\_rdy, Os\_tcb\_cur,U32\_critical\_sr, Os\_tcb\_prio\_tbl,

Modified : Os\_tcb\_high\_rdy,U32\_critical\_sr

#### 10.19.4.4 Parameter list (Input/Output)

Inputs : None

Outputs : None

#### 10.19.4.5 Return Value

None

#### 10.19.4.6 Other CSUs called by this CSU

SaveStatusReg,

RestoreStatusReg,

OSCtxSw

#### 10.19.4.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to OsSched.

##### 10.19.4.7.1 dauanaucos-OsSched-LLR-001

Requirement ID: H398-LLD-ANA-FNC-968

The function shall call ‘SaveStatusReg’ (M\_OS\_ENTER\_CRITICAL) to disable interrupts before execution of the scheduler begins and stores return value of function SaveStatusReg’ in U32\_critical\_sr.

##### 10.19.4.7.2 dauanaucos-OsSched-LLR-002

Requirement ID: H398-LLD-ANA-FNC-969

The function shall do the following when the task scheduling is enabled and not the ISR level i.e. check when the Interrupt nesting level and Multitasking lock nesting levels are zero.

- Set the high priority task index to the priority resolution table with index ready list of task

- Sets the 'Os\_tcb\_high\_rdy' to the highest priority task ready to run from the Os tcb priority table (i.e. Os\_tcb\_prio\_tbl with index as (highest priority task bit shifted to left by M\_THREE) added to the priority resolution table with index (table of tasks which are ready to run with index (highest priority task))).

##### 10.19.4.7.3 dauanaucos-OsSched-LLR-003

Requirement ID: H398-LLD-ANA-FNC-970

The function shall do the following when the Interrupt nesting level, Multitasking lock nesting level are zero:

a) When the task to be scheduled next 'Os\_tcb\_high\_rdy' is not the current running task 'Os\_tcb\_cur' do the following:

i) Increment context switch counter by one

ii) call OsCtxSw (by using M\_OS\_TASK\_SW) to perform context switch.

b) Otherwise do nothing.

##### 10.19.4.7.4 dauanaucos-OsSched-LLR-004

Requirement ID: H398-LLD-ANA-FNC-971

The function shall do nothing when the Interrupt nesting level and Multitasking lock nesting level are not equal to zero.

##### 10.19.4.7.5 dauanaucos-OsSched-LLR-005

Requirement ID: H398-LLD-ANA-FNC-972

The function shall call the ‘RestoreStatusReg’ (M\_OS\_EXIT\_CRITICAL) with parameter U32\_critical\_sr to restore the priority mask register.

### 10.19.5 OsTcbInit

Low Level Design Details about CSU OsTcbInit will follow in the sub sections.

#### 10.19.5.1 Brief Description

This function Initializes the OS\_TCB - Each time a task is created, OsTcbInit ()

is called by OSTaskCreate ().

#### 10.19.5.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.19.5.3 List of global variables accessed and modified

Accessed : U32\_critical\_sr

Modified : U32\_critical\_sr

#### 10.19.5.4 Parameter list (Input/Output)

Inputs : T\_UBYTE prio Task priority

void \* stck Pointer to the top of stack

Outputs : None

#### 10.19.5.5 Return Value

T\_UBYTE - Return the status

M\_OS\_NO\_ERR - TCB created successfully.

M\_OS\_NO\_MORE\_TCB - No more free TCB in the TCB list

#### 10.19.5.6 Other CSUs called by this CSU

SaveStatusReg,

RestoreStatusReg

#### 10.19.5.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to OsTcbInit.

##### 10.19.5.7.1 dauanaucos-OsTcbInit-LLR-001

Requirement ID: H398-LLD-ANA-FNC-981

The function shall call ‘SaveStatusReg’ (M\_OS\_ENTER\_CRITICAL), return value gets stored in U32\_critical\_sr to disable interrupts before creation of TCB.

##### 10.19.5.7.2 dauanaucos-OsTcbInit-LLR-002

Requirement ID: H398-LLD-ANA-FNC-982

The function shall set a free TCB from the free TCB list (Os tcb free list).

##### 10.19.5.7.3 dauanaucos-OsTcbInit-LLR-003

Requirement ID: H398-LLD-ANA-FNC-983

The function shall perform following operations when free TCB is obtained successfully from the free TCB list Os tcb free list (i.e not equal to M\_NULL).

a) Update the free TCB list with the pointer to the next TCB (os\_tcb\_next) in the free TCB list

b) Call ‘RestoreStatusReg’ (M\_OS\_EXIT\_CRITICAL) to restore the priority mask register with parameter U32\_critical\_sr

c) set the following members of the free TCB list

- pointer to current stack(os\_tcb\_stkptr) to stack pointer 'stck'.

- task priority (os\_tcb\_prio) to the current task prioirty 'prio'.

- task status (os\_tcb\_stat) to M\_OS\_STAT\_RDY.

- task delay (os\_tcb\_dly) to M\_ZERO.

- os\_tcb\_y to current task priority 'prio' right shifted by M\_THREE.

- os\_tcb\_bity to value of Mapping table with index as os\_tcb\_y of free TCB list structure.

- os\_tcb\_x to current task priority 'prio' bitwise AND with M\_HEX\_SEVEN.

- os\_tcb\_bitx to value of Mapping table with index as os\_tcb\_x of free TCB list structure.

- os\_tcb\_eventptr to M\_NULL.

- call ‘SaveStatusReg’ (M\_OS\_ENTER\_CRITICAL) to disable interrupts, return valueof ‘SaveStatusReg’ gets stored in U32\_critical\_sr.

d) set Os\_tcb\_prio\_tbl with index 'prio' to free TCB

e) set the free TCB member

- os\_tcb\_next to pointer to next TCB in the Pointer to list of TCBs.

- os\_tcb\_prev to M\_NULL

f) Set os\_tcb\_prev of Pointer to list of TCBs to free TCB list when Pointer to list of TCBs has valid data(i.e not equal to M\_NULL), otherwise do nothing.

g) Set the Pointer to list of TCBs to free TCB list.

h) Make the task ready to run by updating the following values

- Ready list group Bitwise OR with os\_tcb\_bity of free TCB

- Table of tasks which are ready to run with index os\_tcb\_y of free TCB to Table of tasks which are ready to run Bitwise OR with os\_tcb\_bitx of free TCB

##### 10.19.5.7.4 dauanaucos-OsTcbInit-LLR-004

Requirement ID: H398-LLD-ANA-FNC-984

The function shall call ‘RestoreStatusReg’ (M\_OS\_EXIT\_CRITICAL) with parameter U32\_critical\_sr to restore the priority mask register after the successful creation of TCB is completed.

##### 10.19.5.7.5 dauanaucos-OsTcbInit-LLR-005

Requirement ID: H398-LLD-ANA-FNC-985

The function shall return M\_OS\_NO\_ERR on successful creation of TCB.

##### 10.19.5.7.6 dauanaucos-OsTcbInit-LLR-006

Requirement ID: H398-LLD-ANA-FNC-986

The function shall call ‘RestoreStatusReg’ (M\_OS\_EXIT\_CRITICAL) with parameter U32\_critical\_sr to restore the priority mask register when the list of free TCB is equal to M\_NULL.

##### 10.19.5.7.7 dauanaucos-OsTcbInit-LLR-007

Requirement ID: H398-LLD-ANA-FNC-987

The function shall return M\_OS\_NO\_MORE\_TCB when the list of free TCB is equal to M\_NULL.

### 10.19.6 OsIntEnter

Low Level Design Details about CSU OsIntEnter will follow in the sub sections.

#### 10.19.6.1 Brief Description

The function OsIntEnter increments ISR nesting level.

#### 10.19.6.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.19.6.3 List of global variables accessed and modified

Accessed : U32\_critical\_sr

Modified : U32\_critical\_sr

#### 10.19.6.4 Parameter list (Input/Output)

Inputs : None

Outputs : None

#### 10.19.6.5 Return Value

None

#### 10.19.6.6 Other CSUs called by this CSU

SaveStatusReg,

RestoreStatusReg

#### 10.19.6.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to OsIntEnter.

##### 10.19.6.7.1 dauanaucos-OsIntEnter-LLR-001

Requirement ID: H398-LLD-ANA-FNC-996

The function shall do the following on entering an ISR

- Send a request to uC/OS to disable all the interrupts by calling the function ‘SaveStatusReg’ (M\_OS\_ENTER\_CRITICAL) and return value gets stored in U32\_critical\_sr.

- Increment ISR nesting level counter by one.

- Send a request to uC/OS to restore the priority mask register by calling ‘RestoreStatusReg’ (M\_OS\_EXIT\_CRITICAL) with parameter U32\_critical\_sr.

### 10.19.7 OsIntExit

Low Level Design Details about CSU OsIntExit will follow in the sub sections.

#### 10.19.7.1 Brief Description

The function OsIntExit decrements ISR nesting level and perform interrupt level context switch.

#### 10.19.7.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.19.7.3 List of global variables accessed and modified

Accessed : Os\_tcb\_prio\_tbl, Os\_tcb\_high\_rdy, Os\_tcb\_cur,U32\_critical\_sr

Modified : Os\_tcb\_high\_rdy,U32\_critical\_sr

#### 10.19.7.4 Parameter list (Input/Output)

Inputs : None

Outputs : None

#### 10.19.7.5 Return Value

None

#### 10.19.7.6 Other CSUs called by this CSU

OsIntCtxSw,

SaveStatusReg,

RestoreStatusReg

#### 10.19.7.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to OsIntExit.

##### 10.19.7.7.1 dauanaucos-OsIntExit-LLR-001

Requirement ID: H398-LLD-ANA-FNC-1005

The function shall call ‘SaveStatusReg’ (M\_OS\_ENTER\_CRITICAL) to disable interrupts before execution of the function and return value of ‘SaveStatusReg’ gets stored in U32\_critical\_sr.

##### 10.19.7.7.2 dauanaucos-OsIntExit-LLR-002

Requirement ID: H398-LLD-ANA-FNC-1006

The function shall do the following when the ISR is complete and not locked i.e. check when the (pre decremented Interrupt nesting level bitwise OR with Multitasking lock nesting level ) is M\_ZERO.

- Set OS init exit to the priority of the highest priority task that is ready to run from the priority resolution table

- Set the 'Os\_tcb\_high\_rdy' to the highest priority task ready to run from the 'Os\_tcb\_prio\_tbl' with index ((OS init exit bit shifted to left by M\_THREE) added to priority resolution table with index (table of task which are ready to run with index (OS init exit)))

##### 10.19.7.7.3 dauanaucos-OsIntExit-LLR-003

Requirement ID: H398-LLD-ANA-FNC-1007

The function shall perform the following operations when the ISR is complete and not locked (i.e pre decremented Interrupt nesting level bitwise OR with Multitasking lock nesting level) is equal to M\_ZERO:

a) When the task to be run next 'Os\_tcb\_high\_rdy' is not the current running task 'Os\_tcb\_cur’.

i) Increment context switch counter by one.

ii) call ‘OsIntCtxSw’ to perform interrupt level context switch.

b) Otherwise do nothing.

##### 10.19.7.7.4 dauanaucos-OsIntExit-LLR-004

Requirement ID: H398-LLD-ANA-FNC-1008

The function shall do nothing when pre decremented Interrupt nesting level bitwise OR with Multitasking lock nesting level is not equal to M\_ZERO.

##### 10.19.7.7.5 dauanaucos-OsIntExit-LLR-005

Requirement ID: H398-LLD-ANA-FNC-1009

The function shall call the ‘RestoreStatusReg’ (M\_OS\_EXIT\_CRITICAL) with parameter U32\_critical\_sr to restore the priority mask register.

### 10.19.8 OsTimeTick

Low Level Design Details about CSU OsTimeTick will follow in the sub sections.

#### 10.19.8.1 Brief Description

The function OsTimeTick processes the clock tick for the task or the ISR. Check all the task to see when they are either waiting for time to expire (Call to OsTimeDly ()) or waiting for events to occur until they timeout.

#### 10.19.8.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.19.8.3 List of global variables accessed and modified

Accessed : U32\_critical\_sr

Modified : U32\_critical\_sr

#### 10.19.8.4 Parameter list (Input/Output)

Inputs : None

Outputs : None

#### 10.19.8.5 Return Value

None

#### 10.19.8.6 Other CSUs called by this CSU

SaveStatusReg,

RestoreStatusReg

#### 10.19.8.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to OsTimeTick.

##### 10.19.8.7.1 dauanaucos-OsTimeTick-LLR-001

Requirement ID: H398-LLD-ANA-FNC-1028

The function shall loop through all TCBs in TCB list till the task priority for the TCB is M\_OS\_LOWEST\_PRIO and does the following:

* Send a request to uC/OS to disable all the interrupts by calling ‘SaveStatusReg’ (M\_OS\_ENTER\_CRITICAL) and return value of ‘SaveStatusReg’ gets stored in U32\_critical\_sr.
* Do the following when number of ticks to os\_tcb\_dly for the TCB is not Zero:

1. Decrement the number of ticks os\_tcb\_dly for the TCB
2. Make task Ready to Run (i.e Set Os ready group to (os ready group bitwise OR with os\_tcb\_bity of TCB) and set table of task which are ready to run with index (os\_tcb\_y of TCB) to (table of task which are ready to run with index (os\_tcb\_y of TCB) bitwise OR with os\_tcb\_bitx of TCB)) when number of ticks os\_tcb\_dly for the TCB is Zero and (Negation of (task status os\_tcb\_stat for the TCB Bitwise AND with (M\_OS\_STAT\_SUSPEND))) is not equal to FALSE.
3. Set the number of ticks os\_tcb\_dly for the TCB to M\_ONE when number of ticks os\_tcb\_dly for the TCB is Zero and (Negation of (task status os\_tcb\_stat for the TCB Bitwise AND with (M\_OS\_STAT\_SUSPEND))) is equal to FALSE.
4. Do nothing when decremented number of ticks os\_tcb\_dly for the TCB is not equal to M\_ZERO.

* Do nothing when number of ticks to os\_tcb\_dly for the TCB is M\_ZERO.
* Send a request to uC/OS to enable all the interrupts by calling ‘RestoreStatusReg’ (M\_OS\_EXIT\_CRITICAL) with parameter U32\_critical\_sr.
* Point the current TCB to next TCB os\_tcb\_next.

##### 10.19.8.7.2 dauanaucos-OsTimeTick-LLR-002

Requirement ID: H398-LLD-ANA-FNC-1029

The function shall send a request to uC/OS to disable all the interrupts by calling ‘SaveStatusReg’ (M\_OS\_ENTER\_CRITICAL) and return value gets stored in U32\_critical\_sr.

##### 10.19.8.7.3 dauanaucos-OsTimeTick-LLR-003

Requirement ID: H398-LLD-ANA-FNC-1030

The function shall increment the Current value of system time by one.

##### 10.19.8.7.4 dauanaucos-OsTimeTick-LLR-004

Requirement ID: H398-LLD-ANA-FNC-1031

The function shall send a request to uC/OS to enable all the interrupts by calling ‘RestoreStatusReg’ (M\_OS\_EXIT\_CRITICAL) with parameter U32\_critical\_sr.

### 10.19.9 OsSemCreate

Low Level Design Details about CSU OsSemCreate will follow in the sub sections.

#### 10.19.9.1 Brief Description

The function OsSemCreate creates and initialize Semaphore.

#### 10.19.9.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

10.19.9.3 List of global variables accessed and modified

Accessed : U32\_critical\_sr

Modified : U32\_critical\_sr

#### 10.19.9.4 Parameter list (Input/Output)

Inputs : T\_UWORD count The initial value of the semaphore

Outputs : None

#### 10.19.9.5 Return Value

T\_OS\_EVENT\* Return Pointer to the event control block (ECB) associated with

the created semaphore.

#### 10.19.9.6 Other CSUs called by this CSU

SaveStatusReg,

RestoreStatusReg

#### 10.19.9.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to OsSemCreate.

##### 10.19.9.7.1 dauanaucos-OsSemCreate-LLR-001

Requirement ID: H398-LLD-ANA-FNC-1040

The function shall call ‘SaveStatusReg’ (M\_OS\_ENTER\_CRITICAL), return value gets stored in U32\_critical\_sr to disable interrupts.

##### 10.19.9.7.2 dauanaucos-OsSemCreate-LLR-002

Requirement ID: H398-LLD-ANA-FNC-1041

The function shall store an ECB from free list of ECBs.

##### 10.19.9.7.3 dauanaucos-OsSemCreate-LLR-003

Requirement ID: H398-LLD-ANA-FNC-1042

The function shall set the list of free ECB to next free list of ECB (os\_eventptr) when the the list of free ECB is not M\_NULL, otherwise do nothing.

##### 10.19.9.7.4 dauanaucos-OsSemCreate-LLR-004

Requirement ID: H398-LLD-ANA-FNC-1043

The function shall enable all the interrupts by calling ‘RestoreStatusReg’ (M\_OS\_EXIT\_CRITICAL) with parameter U32\_critical\_sr.

##### 10.19.9.7.5 dauanaucos-OsSemCreate-LLR-005

Requirement ID: H398-LLD-ANA-FNC-1044

The function shall do the following when the reference to the selected ECB is not equal to M\_NULL, otherwise return M\_NULL.

- Set the desired initial count (count) for the Semaphore into os\_eventcnt of obtained ECB.

- Clear the os\_eventgrp by setting it to M\_HEX2\_ZERO

-Loop through (until M\_MAX\_TSK\_WAIT\_EVNT) to clear os\_event\_tbl of the obtained ECB by setting it to M\_HEX2\_ZERO.

- Return with the reference to the obtained ECB.

### 10.19.10 OsSemPend

Low Level Design Details about CSU OsSemPend will follow in the sub sections.

#### 10.19.10.1 Brief Description

The function OsSemPend waits on a semaphore.

#### 10.19.10.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.19.10.3 List of global variables accessed and modified

Accessed : U32\_critical\_sr, Os\_tcb\_cur

Modified : U32\_critical\_sr, Os\_tcb\_cur

#### 10.19.10.4 Parameter list (Input/Output)

Inputs: T\_OS\_EVENT \*p\_event Pointer to the desired semaphore's ECB

T\_UWORD timeout Time in clock ticks to wait for the resource.

If 0, the task will wait until the resource becomes available or the

event occurs.

Outputs: T\_OS\_EVENT \*p\_event Pointer to the desired semaphore's ECB

T\_UBYTE \*error Pointer to error message.

- Set to M\_OS\_NO\_ERR if the semaphore was available.

- Set to M\_OS\_TIMEOUT if the semaphore was not signaled

within the specified timeout.

#### 10.19.10.5 Return Value

None

#### 10.19.10.6 Other CSUs called by this CSU

SaveStatusReg,

RestoreStatusReg,

OsSched

#### 10.19.10.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to OsSemPend.

##### 10.19.10.7.1 dauanaucos-OsSemPend-LLR-001

Requirement ID: H398-LLD-ANA-FNC-1053

The function shall call ‘SaveStatusReg’ (M\_OS\_ENTER\_CRITICAL) to disable interrupts and return value gets stored in U32\_critical\_sr.

##### 10.19.10.7.2 dauanaucos-OsSemPend-LLR-002

Requirement ID: H398-LLD-ANA-FNC-1054

The function shall do the following when the event counter of the current TCB (os\_eventcnt of p\_event) is greater than Zero (when semaphore is positive, resource is available).

- Decrement the Semaphore counter os\_eventcnt of the current TCB(p\_event) by one

- Send a request to uC/OS to enable all the interrupts by calling ‘RestoreStatusReg’ (M\_OS\_EXIT\_CRITICAL) with parameter U32\_critical\_sr.

- Set the error message to M\_OS\_NO\_ERR.

##### 10.19.10.7.3 dauanaucos-OsSemPend-LLR-003

Requirement ID: H398-LLD-ANA-FNC-1055

The function shall do the following when the event counter (os\_eventcnt) of the current TCB p\_event is less than or equal to Zero (when semaphore is negative, resource is not available):

* Set os\_tcb\_stat of Os\_tcb\_cur to os\_tcb\_stat Os\_tcb\_cur bitwise OR with M\_OS\_STAT\_SEM.
* Set the number of ticks os\_tcb\_dly of currently running TCB Os\_tcb\_cur to value of timeout.
* Set TCB event pointer os\_tcb\_eventptr of currently running TCB Os\_tcb\_cur to ECB location of the Semaphore(i.e.p\_event)
* Set Table of task which are ready to run with index os\_tcb\_y of Os\_tcb\_cur to (Table of task which are ready to run with index os\_tcb\_y of Os\_tcb\_cur bitwise AND with (Negation of (os\_tcb\_bitx of Os\_tcb\_cur))).
* Set Os ready group to (Os ready group bitwise AND (Negation of os\_tcb\_bity of Os\_tcb\_cur)) when table of task which are ready to run with index os\_tcb\_y of Os\_tcb\_cur is M\_ZERO, otherwise do nothing.
* Put the task in waiting list by doing the following:

1. Set os\_event\_tbl of ECB p\_event with index (os\_tcb\_y of Os\_tcb\_cur) to (os\_event\_tbl of ECB p\_event with index (os\_tcb\_y of Os\_tcb\_cur) bitwise OR with os\_tcb\_bitx of Os\_tcb\_cur).
2. Set os\_eventgrpof ECB p\_event to (os\_eventgrp of ECB p\_event bitwise OR with os\_tcb\_bity of Os\_tcb\_cur).

* Send a request to uC/OS to enable all the interrupts by calling ‘RestoreStatusReg’ (M\_OS\_EXIT\_CRITICAL) with parameter U32\_critical\_sr.
* Call the function 'OsSched' to schedule the next highest priority task.
* Send a request to uC/OS to disable all the interrupts by calling ‘SaveStatusReg’ (M\_OS\_ENTER\_CRITICAL) and return value of ‘SaveStatusReg’ gets stored in U32\_critical\_sr.
* Do the following when os\_tcb\_stat of Os\_tcb\_cur bitwise AND with M\_OS\_STAT\_SEM is not equal to FALSE:

1. Set os\_event\_tbl of p\_event with index os\_tcb\_y of Os\_tcb\_cur to (os\_event\_tbl of p\_event with index os\_tcb\_y of Os\_tcb\_cur bitwise AND with (Negation of (os\_tcb\_bitx of Os\_tcb\_cur))).
2. Set os\_eventgrp of p\_event to (os\_eventgrp of p\_event Bitwise AND with (Negation of (os\_tcb\_bity of Os\_tcb\_cur))) when os\_event\_tbl of p\_event is M\_ZERO, otherwise do nothing.
3. Set Task status os\_tcb\_stat of current running TCB Os\_tcb\_cur to 'M\_OS\_STAT\_RDY'.
4. Set the Event pointer os\_tcb\_eventptr of currently running TCB Os\_tcb\_cur to M\_NULL.
5. Send a request to uC/OS to enable all the interrupts by calling ‘RestoreStatusReg’ (M\_OS\_EXIT\_CRITICAL) with parameter U32\_critical\_sr.
6. Set the error message pointer (i.e.- \*error) to M\_OS\_TIMEOUT

* Do the following when os\_tcb\_stat of Os\_tcb\_cur bitwise AND with M\_OS\_STAT\_SEM is equal to FALSE:

1. Set the Event pointer os\_tcb\_eventptr of current running TCB Os\_tcb\_cur to M\_NULL.
2. Send a request to uC/OS to enable all the interrupts by calling ‘RestoreStatusReg’ (M\_OS\_EXIT\_CRITICAL) with parameter U32\_critical\_sr.
3. Set the error message to M\_OS\_NO\_ERR.

### 10.19.11 OsSemPost

Low Level Design Details about CSU OsSemPost will follow in the sub sections.

#### 10.19.11.1 Brief Description

The function OsSemPost posts to a Semaphore: Release the resource.

#### 10.19.11.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.19.11.3 List of global variables accessed and modified

Accessed : U32\_critical\_sr, Os\_tcb\_prio\_tbl

Modified : U32\_critical\_sr

#### 10.19.11.4 Parameter list (Input/Output)

Inputs : T\_OS\_EVENT \*p\_event Pointer to the desired semaphore's ECB

Outputs : T\_OS\_EVENT \*p\_event Pointer to the desired semaphore's ECB

#### 10.19.11.5 Return Value

T\_UBYTE - returns the error code

M\_OS\_NO\_ERR -a) Semaphore posted successfully.

b) Semaphore value within limit and no task waiting for Semaphore.

M\_OS\_SEM\_OVF -overflow of Semaphore value.

#### 10.19.11.6 Other CSUs called by this CSU

OsSched,

SaveStatusReg,

RestoreStatusReg

#### 10.19.11.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to OsSemPost.

##### 10.19.11.7.1 dauanaucos-OsSemPost-LLR-001

Requirement ID: H398-LLD-ANA-FNC-1064

The function shall call ‘SaveStatusReg’ (M\_OS\_ENTER\_CRITICAL) to disable interrupts and return value of ‘SaveStatusReg’ gets stored in U32\_critical\_sr.

##### 10.19.11.7.2 dauanaucos-OsSemPost-LLR-002

Requirement ID: H398-LLD-ANA-FNC-1065

The function shall do the following when any of the task is waiting on Semaphore (i.e. os\_eventgrp of ECB p\_event is not equal to FALSE):

* Set second set of priority index to priority resolution table with index os\_eventgrp of p\_event.
* Set bit position for second set of priority index to mapping table to map bit position to bit mask with index second set of priority index.
* Set first set of priority index to priority resolution table with index os\_event\_tbl with index as second set of priority index of p\_event.
* Set bit position for first set of priority index to mapping table to map bit position to bit mask with index as first set of priority index.
* Get the index of highest priority task pending on event as ((second set of priority index bit shifted to left by M\_THREE) added to first set of priority index).
* Set os\_event\_tbl of p\_event with index second set of priority index to (os\_event\_tbl of p\_event with index second set of priority index bitwise AND with (Negation of (bit position for first set of priority index))).
* Set os\_eventgrp of p\_event to (os\_eventgrp of p\_event Bitwise AND with (Negation of (bit position for second set of priority index))) when os\_event\_tbl of p\_event is M\_ZERO, otherwise do nothing.
* Point the current TCB to Os\_tcb\_prio\_tbl with index as highest priority task index.
* Set os\_tcb\_dly of current task TCB to M\_ZERO.
* Set Event pointer os\_tcb\_eventptr of the current task TCB to M\_NULL.
* Set os\_tcb\_stat of the current task TCB to (os\_tcb\_stat of the current task TCB bitwise AND with (Negation of (M\_OS\_STAT\_SEM))).
* Set Ready list of group to (Ready list of group bitwise OR with bit position for second set of priority index) and table of task which are ready to run with index second set of priority index to (table of task which are ready to run with index second set of priority index bitwise OR with bit position for first set of priority index) when os\_tcb\_stat of current TCB is equal to M\_OS\_STAT\_RDY, otherwise do nothing.
* Send a request to uC/OS to enable all the interrupts by calling ‘RestoreStatusReg’ (M\_OS\_EXIT\_CRITICAL) with parameter U32\_critical\_sr.
* Call the function 'OsSched' to schedule the next highest priority task.
* Return with M\_OS\_NO\_ERR.

##### 10.19.11.7.3 dauanaucos-OsSemPost-LLR-003

Requirement ID: H398-LLD-ANA-FNC-1066

The function shall do the following when no task is waiting for Semaphore and the Semaphore value is within the limit. (i.e. os\_eventgrp of ECB p\_event equal to Zero AND os\_eventcnt of ECB p\_event is less than M\_MAX\_SEMPHORE)

a) Increment the Semaphore count to register event os\_eventcnt of ECB p\_event by one

b) Send a request to uC/OS to enable all the interrupts by calling ‘RestoreStatusReg’ (M\_OS\_EXIT\_CRITICAL) with parameter U32\_critical\_sr.

c) Return with M\_OS\_NO\_ERR

##### 10.19.11.7.4 dauanaucos-OsSemPost-LLR-004

Requirement ID: H398-LLD-ANA-FNC-1067

The function shall do the following when no task is waiting for Semaphore and the Semaphore value is over the limit. (i.e. os\_eventgrp of ECB p\_event equal to Zero AND os\_eventcnt of ECB p\_event is greater than M\_MAX\_SEMPHORE)

a) Send a request to uC/OS to enable all the interrupts by calling ‘RestoreStatusReg’ (M\_OS\_EXIT\_CRITICAL) with parameter U32\_critical\_sr.

b) Return with M\_OS\_SEM\_OVF

## 10.20 dauanautils

This module contains Implementation of all the utility routines

### 10.20.1 UtilsCopy

Low Level Design Details about CSU UtilsCopy will follow in the sub sections.

#### 10.20.1.1 Brief Description

The function UtilsCopy copies the character string from the source reference to the destination as per the length supplied.

#### 10.20.1.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.20.1.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 10.20.1.4 Parameter list (Input/Output)

Inputs : T\_UINT8 \*par\_str\_src Pointer to the src address

T\_UINT32 Size\_in\_byte Length of the string to be copied

T\_UINT8 \*par\_str\_dest Pointer to the dest address

Outputs : T\_UINT8 \*par\_str\_dest Pointer to the dest address

#### 10.20.1.5 Return Value

None

#### 10.20.1.6 Other CSUs called by this CSU

None

#### 10.20.1.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to UtilsCopy.

##### 10.20.1.7.1 dauanautils-UtilsCopy-LLR-001

Requirement ID: H398-LLD-ANA-DRQ-1077

The function shall return from the function when the size of the string(Size\_in\_byte) is zero Logically OR with the Reference to the source address(par\_str\_src) is null Logically OR with the Reference to the destination address(par\_str\_dest) is null, otherwise do nothing.

##### 10.20.1.7.2 dauanautils-UtilsCopy-LLR-002

Requirement ID: H398-LLD-ANA-DRQ-1078

Rationale: This is a generic utility function to copy data from one area in memory location to another and not specific to any functionality.document HE0398AHA and to execute the application software in normal mode.

The function shall copy the input string from the given source address (i.e: par\_str\_src) to given destination address (i.e: par\_str\_dest) as per the length specified (Size\_in\_byte) and return from the function.

## 10.21 dauanawdog

This module contains Implementation of watchdog timer routines

### 10.21.1 WdogInit

Low Level Design Details about CSU WdogInit will follow in the sub sections.

#### 10.21.1.1 Brief Description

The function WdogInit calculates the reload value for the independent watchdog timer,initialize the prescaler divider to 64,write reload value to the reload register,enable the watchdog register and reload watchdog counter.

#### 10.21.1.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.21.1.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 10.21.1.4 Parameter list (Input/Output)

Inputs : None

Outputs : None

#### 10.21.1.5 Return Value

None

#### 10.21.1.6 Other CSUs called by this CSU

IwdgWriteAccessCmd,

IwdgSetPrescaler,

IwdgSetReload,

IwdgEnable,

IwdgReloadCounter

#### 10.21.1.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to WdogInit.

##### 10.21.1.7.1 dauanawdog-WdogInit-LLR-001

Requirement ID: H398-LLD-ANA-FNC-1088

The function shall calculate the reload value for the independent watchdog timer as the product of the watchdog timeout (M\_WDOG\_TIMEOUT\_MS) and watchdog frequency (M\_WDOG\_FREQ) divided by (M\_WATCHDOG\_FREQ\_KHZ\_SCALE multiplied into (M\_GET\_PRESCALER\_4\_FOR\_WD bit shift to left by M\_WDOG\_PRESCALER))

##### 10.21.1.7.2 dauanawdog-WdogInit-LLR-002

Requirement ID: H398-LLD-ANA-FNC-1089

The function shall set the calculated reload value to M\_MAX\_RELOAD\_VALUE when reload value is greater than M\_MAX\_RELOAD\_VALUE, otherwise do nothing.

##### 10.21.1.7.3 dauanawdog-WdogInit-LLR-003

Requirement ID: H398-LLD-ANA-FNC-1090

The function shall do the call IwdgWriteAccessCmd with parameter M\_IWDG\_WRITEACCESS\_ENABLE to enable write access to the prescaler and reload registers before writing into the registers.

(Note : first write 0x5555 in the kr register to have wr access to prescaler and reload registers of independent watchdog)

##### 10.21.1.7.4 dauanawdog-WdogInit-LLR-004

Requirement ID: H398-LLD-ANA-FNC-1091

The function shall call IwdgSetPrescaler with parameter M\_WDOG\_PRESCALER to set the prescaler divider to 64.

##### 10.21.1.7.5 dauanawdog-WdogInit-LLR-005

Requirement ID: H398-LLD-ANA-FNC-1092

The function shall call IwdgSetReload with the calculated reload value as parameter to write reload value to the reload register .

##### 10.21.1.7.6 dauanawdog-WdogInit-LLR-006

Requirement ID: H398-LLD-ANA-FNC-1093

The function shall call IwdgWriteAccessCmd with parameter M\_IWDG\_WRITEACCESS\_DISABLE to disable write access of prescaler and reload registers after write has been completed.

(Note : first write 0x5555 in the kr register to have wr access to prescaler and reload registers of independent watchdog )

##### 10.21.1.7.7 dauanawdog-WdogInit-LLR-007

Requirement ID: H398-LLD-ANA-FNC-1094

The function shall call IwdgEnable to enable watchdog timer.

##### 10.21.1.7.8 dauanawdog-WdogInit-LLR-008

Requirement ID: H398-LLD-ANA-FNC-1095

The function shall call IwdgReloadCounter to reload watchdog counter.

### 10.21.2 WdogKickWatchDog

Low Level Design Details about CSU WdogKickWatchDog will follow in the sub sections.

#### 10.21.2.1 Brief Description

The function WdogKickWatchDog reloads the Watch dog counter.

#### 10.21.2.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.21.2.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 10.21.2.4 Parameter list (Input/Output)

Inputs : None

Outputs : None

#### 10.21.2.5 Return Value

None

#### 10.21.2.6 Other CSUs called by this CSU

IwdgReloadCounter

#### 10.21.2.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to WdogKickWatchDog.

##### 10.21.2.7.1 dauanawdog-WdogKickWatchDog-LLR-001

Requirement ID: H398-LLD-ANA-FNC-1104

The function shall call IwdgReloadCounter to reload the watchdog counter.

## 10.22 dauanaxadc

This module contains implementation of Interface to external M\_ADC.

### 10.22.1 XADCRead

Low Level Design Details about CSU XADCRead will follow in the sub sections.

#### 10.22.1.1 Brief Description

The function XADCRead calculates and return the average of ADC reading.

#### 10.22.1.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.22.1.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 10.22.1.4 Parameter list (Input/Output)

Inputs : T\_UINT8 u8\_Chan Channel no of External ADC

T\_UINT8 u8\_Mux Mux no of External ADC

Outputs : None

#### 10.22.1.5 Return Value

T\_SINT16 - The average of ADC reading

#### 10.22.1.6 Other CSUs called by this CSU

None

#### 10.22.1.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to XADCRead.

##### 10.22.1.7.1 dauanaxadc-XADCRead-LLR-001

Requirement ID: H398-LLD-ANA-FNC-1114

The function shall Calculate and return the average ADC reading as follows

Average ADC reading bit shifted to right by M\_XADC\_POWER.

### 10.22.2 SetMultiplexer

Low Level Design Details about CSU SetMultiplexer will follow in the sub sections.

#### 10.22.2.1 Brief Description

The function SetMultiplexer sets the Multiplexer for XADC Gpio pins.

#### 10.22.2.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.22.2.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 10.22.2.4 Parameter list (Input/Output)

Inputs : T\_UINT8 u8\_Mux Mux no of External ADC

Outputs : T\_UINT8 u8\_Mux Mux no of External ADC

#### 10.22.2.5 Return Value

None

#### 10.22.2.6 Other CSUs called by this CSU

GpioSetBits,

GpioResetBits

#### 10.22.2.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to SetMultiplexer.

##### 10.22.2.7.1 dauanaxadc-SetMultiplexer-LLR-001

Requirement ID: H398-LLD-ANA-FNC-1123

The function shall Loop through all the multiplexer control pins (until M\_MULTIXER\_CTRL\_PIN\_NUM) and do the following

- Call the function 'GpioSetBits' with parameters M\_GPIOC and (M\_GPIOC\_XADC\_MUX\_0 bit shifted to left by loop counter) when u8\_Mux bitwise AND with TRUE is equal to TRUE

- Call the function 'GpioResetBits' with parameter M\_GPIOC and (M\_GPIOC\_XADC\_MUX\_0 bit shifted to left by loop counter) when u8\_Mux bitwise AND with TRUE is not equal to TRUE

### 10.22.3 ReadExternalADC

Low Level Design Details about CSU ReadExternalADC will follow in the sub sections.

#### 10.22.3.1 Brief Description

The function ReadExternalADC starts reading of external ADC.

#### 10.22.3.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.22.3.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 10.22.3.4 Parameter list (Input/Output)

Inputs : None

Outputs : None

#### 10.22.3.5 Return Value

None

#### 10.22.3.6 Other CSUs called by this CSU

TimClearITPendingBit,

GpioSetBits,

GpioResetBits,

SetMultiplexer

#### 10.22.3.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to ReadExternalADC.

##### 10.22.3.7.1 dauanaxadc-ReadExternalADC-LLR-001

Requirement ID: H398-LLD-ANA-FNC-1132

The function shall Call 'TimClearITPendingBit' with parameter (M\_TIM7, M\_TIM\_IT\_UPDATE) to Clear Timer 7 update interrupt pending bit.

##### 10.22.3.7.2 dauanaxadc-ReadExternalADC-LLR-002

Requirement ID: H398-LLD-ANA-FNC-1133

The function shall do the following when the selected ADC state is XADC\_CONVERT\_STATE

- Call the function GpioSetBits(M\_XADC\_CONVERT) with parameter (M\_GPIOC, M\_GPIOC\_XADC\_CONVST) to start new ADC conversion

- Set the ADC state to XADC\_READ\_STATE

(i.e - The function called first time to initiate ADC reading)

##### 10.22.3.7.3 dauanaxadc-ReadExternalADC-LLR-003

Requirement ID: H398-LLD-ANA-FNC-1134

The function shall do the following when the selected ADC state is XADC\_READ\_STATE

- Call the function 'GpioResetBits' (M\_XADC\_READY\_TO\_READ) with parameter (M\_GPIOC, M\_GPIOC\_XADC\_CONVST) to Bring CONVST line low while reading ADC

- Loop through all the ADC channels (XADC\_CHAN\_1 to XADC\_CHANS) and do the following

a. Set the ADC reading to xadc data for all channel and mux.

b. Apply filter to the reading by doing the following for all channel and mux.:

Actual ADC reading is subtracted by (Average ADC reading bit shifted to right by M\_XADC\_POWER)

c. Update average reading by adding it with filtered adc readings for all channel and mux.

- Set the Mux no to 0 when the Mux no is greater than equals to XADC\_MUX\_CHANS

otherwise do nothing

- Call the function 'SetMultiplexer' with parameter (Mux no) to set the Multiplexer

- Set the ADC state to XADC\_WAIT\_STATE

Note: XADC\_CHANS = Number XADC channels (6)

##### 10.22.3.7.4 dauanaxadc-ReadExternalADC-LLR-004

Requirement ID: H398-LLD-ANA-FNC-1135

The function shall Set the ADC state to XADC\_CONVERT\_STATE when the selected ADC state is XADC\_WAIT\_STATE.

##### 10.22.3.7.5 dauanaxadc-ReadExternalADC-LLR-005

Requirement ID: H398-LLD-ANA-FNC-1136

The function shall do nothing when the selected ADC state is not equal to (XADC\_CONVERT\_STATE and XADC\_READ\_STATE and XADC\_WAIT\_STATE).

### 10.22.4 XADCInit

Low Level Design Details about CSU XADCInit will follow in the sub sections.

#### 10.22.4.1 Brief Description

The function XADCInit initializes MAX12045 ADC.

#### 10.22.4.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.22.4.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 10.22.4.4 Parameter list (Input/Output)

Inputs : None

Outputs : None

#### 10.22.4.5 Return Value

None

#### 10.22.4.6 Other CSUs called by this CSU

FsmcNorSramInit,

FsmcNorSramCmd,

TMRInitTIM7

#### 10.22.4.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to XADCInit.

##### 10.22.4.7.1 dauanaxadc-XADCInit-LLR-001

Requirement ID: H398-LLD-ANA-FNC-1145

The function shall do the following to Initialize the FSMC NOR/SRAM Banks

- Set the FSMC config struct for XADC to the following setting

- fsmc\_address\_setuptime of FSMC config struct to M\_ZERO

- fsmc\_address\_holdtime of FSMC config struct to M\_ZERO

- fsmc\_data\_setuptime of FSMC config struct to M\_FSMC\_HCLK\_CYCLES

- fsmc\_bus\_turnaround\_duration of FSMC config struct to M\_FSMC\_BUS\_TURN\_AROUND

- fsmc\_clk\_division of FSMC config struct to M\_ZERO

- fsmc\_data\_latency of FSMC config struct to M\_ZERO

- fsmc\_access\_mode of FSMC config struct to M\_FSMC\_ACCESSMODE\_A

- Set the SRAM init struct to the following setting

- fsmc\_bank of SRAM init struct to M\_FSMC\_BANK1\_NORSRAM1

- fsmc\_data\_address\_mux of SRAM init struct to M\_FSMC\_DATAADDRESSMUX\_DISABLE

- fsmc\_memory\_type of SRAM init struct to M\_FSMC\_MEMORYTYPE\_PSRAM

- fsmc\_memory\_datawidth of SRAM init struct to M\_FSMC\_MEMORYDATAWIDTH\_16B

- fsmc\_burst\_accessmode of SRAM init struct to M\_FSMC\_BURSTACCESSMODE\_DISABLE

- fsmc\_asynchronous\_wait of SRAM init struct to M\_FSMC\_ASYNCHRONOUSWAIT\_DISABLE

- fsmc\_waitsignal\_polarity of SRAM init struct to M\_FSMC\_WAITSIGNALPOLARITY\_LOW

- fsmc\_wrap\_mode of SRAM init struct to M\_FSMC\_WRAPMODE\_DISABLE

- fsmc\_waitsignal\_active of SRAM init struct to M\_FSMC\_WAIT\_SIG\_ACTIVE\_BEF\_WAIT

- fsmc\_write\_operation of SRAM init struct to M\_FSMC\_WRITEOPERATION\_ENABLE

- fsmc\_waitsignal of SRAM init struct to M\_FSMC\_WAITSIGNAL\_DISABLE

- fsmc\_extended\_mode of SRAM init struct to M\_FSMC\_EXTENDEDMODE\_DISABLE

- fsmc\_write\_burst of SRAM init struct to M\_FSMC\_WRITEBURST\_DISABLE

- fsmc\_readwrite\_timing\_struct of SRAM init struct to Reference of FSMC config struct

- fsmc\_write\_timing\_struct of SRAM init struct to Reference of FSMC config struct

- Call the function FsmcNorSramInit with parameter as reference to the SRAM init struct

##### 10.22.4.7.2 dauanaxadc-XADCInit-LLR-002

Requirement ID: H398-LLD-ANA-FNC-1146

The function shall Call the function 'FsmcNorSramCmd' with Parameter (M\_FSMC\_BANK1\_NORSRAM1, ENABLE) to Enable the NOR/SRAM Memory Bank 1.

##### 10.22.4.7.3 dauanaxadc-XADCInit-LLR-003

Requirement ID: H398-LLD-ANA-FNC-1147

The function shall Configure the external ADC by setting xadc data to M\_XADC\_REF\_SEL Bitwise OR M\_XADC\_DATA\_FORMAT Bitwise OR M\_XADC\_CONVST\_MODE.

##### 10.22.4.7.4 dauanaxadc-XADCInit-LLR-004

Requirement ID: H398-LLD-ANA-FNC-1148

The function shall Call the function 'TMRInitTIM7' with parameter as function ‘ReadExternalADC’ to Initialize timer 7 to Start ADC reading.

## 10.23 dauanasspi

This module contains Provides routines to implement and work with the SPI bus.

### 10.23.1 SSPIclaimBusSPI2

Low Level Design Details about CSU SSPIclaimBusSPI2 will follow in the sub sections.

#### 10.23.1.1 Brief Description

This function requests access to the SPI2 bus by pending a semaphore.

#### 10.23.1.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.23.1.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 10.23.1.4 Parameter list (Input/Output)

Inputs : None

Outputs : None

#### 10.23.1.5 Return Value

None

#### 10.23.1.6 Other CSUs called by this CSU

OsSemPend

#### 10.23.1.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to SSPIclaimBusSPI2.

##### 10.23.1.7.1 dauanasspi-SSPIclaimBusSPI2-LLR-001

Requirement ID: H398-LLD-ANA-FNC-1953

The function shall Call the function OsSemPend with parameters l\_semKeySPI2, zero and reference to u8Err to pend the semaphore.

### 10.23.2 SSPIReleaseBusSPI2

Low Level Design Details about CSU SSPIreleaseBusSPI2 will follow in the sub sections.

#### 10.23.2.1 Brief Description

This function releases control of the SPI2 bus by posting a semaphore.

#### 10.23.2.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.23.2.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 10.23.2.4 Parameter list (Input/Output)

Inputs: None

Outputs: None

#### 10.23.2.5 Return Value

None

#### 10.23.2.6 Other CSUs called by this CSU

OsSemPost

#### 10.23.2.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to SSPIreleaseBusSPI2.

##### 10.23.2.7.1 dauanasspi-SSPIReleaseBusSPI2-LLR-001

Requirement ID: H398-LLD-ANA-FNC-1962

The function shall Call the function OsSemPost with parameters l\_semKeySPI2 to post the semaphore.

### 10.23.3 SSPIWaitSPI2

Low Level Design Details about CSU SSPIwaitSPI2will follow in the sub sections.

#### 10.23.3.1 Brief Description

This function waits for the semaphore associated with SPI2 communication to become available

#### 10.23.3.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.23.3.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 10.23.3.4 Parameter list (Input/Output)

Inputs: None

Outputs: None

#### 10.23.3.5 Return Value

None

#### 10.23.3.6 Other CSUs called by this CSU

OsSemPend

#### 10.23.3.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to SSPIwaitSPI2.

##### 10.23.3.7.1 dauanasspi-SSPIWaitSPI2-LLR-001

Requirement ID: H398-LLD-ANA-FNC-1971

The function shall Call the function OsSemPend with parameters l\_semWaitSPI2, zero and reference to u8Err to pend the semaphore.

### 10.23.4 SSPIFinishSPI2

Low Level Design Details about CSU SSPIfinishSPI2will follow in the sub sections.

#### 10.23.4.1 Brief Description

This function shall Posts the Semaphore indicating the completion of SPI2 communication.

#### 10.23.4.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.23.4.3 List of global variables accessed and modified

None

#### 10.23.4.4 Parameter list (Input/Output)

Inputs: None

Outputs: None

#### 10.23.4.5 Return Value

None

#### 10.23.4.6 Other CSUs called by this CSU

OsSemPost

#### 10.23.4.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to SSPIfinishSPI2.

##### 10.23.4.7.1 dauanasspi-SSPIFinishSPI2-LLR-001

Requirement ID: H398-LLD-ANA-FNC-1980

The function shall Call the function OsSemPost with parameters l\_semWaitSPI2 to post the semaphore.

### 10.23.5 SSPIInitSPI2

Low Level Design Details about CSU SSPIinitSPI2will follow in the sub sections.

#### 10.23.5.1 Brief Description

This function shall create a key semaphore, initialize semaphore to wait for transfer to finish,pins, NVIC channel, hardware and enables bus for SPI2.

#### 10.23.5.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.23.5.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 10.23.5.4 Parameter list (Input/Output)

Inputs: None

Outputs: None

#### 10.23.5.5 Return Value

None

#### 10.23.5.6 Other CSUs called by this CSU

OsSemCreate

RccAhb1PeriphClockCmd

RccApb1PeriphClockCmd

GpioPinAFConfig

GpioInit

NvicInit

SPIInit

SPII2SITConfig

SPICmd

#### 10.23.5.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to SSPIinitSPI2.

##### 10.23.5.7.1 dauanasspi-SSPIInitSPI2-LLR-001

Requirement ID: H398-LLD-ANA-FNC-1989

The function shall perform the following:

1. Call the function OsSemCreate with parameter M\_ONE and set the return value to l\_semKeySPI2.
2. Call the function OsSemCreate with parameter M\_ZERO and set the return value to l\_semWaitSPI2.
3. Call the function RccAhb1PeriphClockCmd with parameters M\_RCC\_AHB1PERIPH\_GPIOB and ENABLE.
4. Call the function RccApb1PeriphClockCmd with parameters M\_RCC\_APB1PERIPH\_SPI2 and ENABLE.
5. Call the function GpioPinAFConfig with parameters M\_GPIOB, M\_GPIO\_PINSOURCE13 and GPIO\_AF\_SPI2.
6. Call the function GpioPinAFConfig with parameters M\_GPIOB, M\_GPIO\_PINSOURCE14 and GPIO\_AF\_SPI2
7. Call the function GpioPinAFConfig with parameters M\_GPIOB, M\_GPIO\_PINSOURCE15 and GPIO\_AF\_SPI2
8. Set gpio\_pin of gpio\_init\_structure to (M\_GPIOB\_SPI2\_SCK BITWISE OR M\_GPIOB\_SPI2\_MISO BITWISE OR M\_GPIOB\_SPI2\_MOSI).
9. Set gpio\_mode of gpio\_init\_structure to GPIO\_MODE\_AF.
10. Set gpio\_speed of gpio\_init\_structure to GPIO\_SPEED\_50MHZ.
11. Set gpio\_otype of gpio\_init\_structure to GPIO\_OTYPE\_PP.
12. gpio\_pupd of gpio\_init\_structure to GPIO\_PUPD\_NOPULL.
13. Call the function GpioInit with parameters M\_GPIOB and reference to gpio\_init\_structure.

##### 10.23.5.7.2 dauanasspi-SSPIInitSPI2-LLR-002

Requirement ID: H398-LLD-ANA-FNC-1990

The function shall perform the following:

1. Set nvic\_irq\_channel of NVIC\_InitStructure to SPI2\_IRQN.
2. Set nvic\_irq\_channel\_preemption\_priority of NVIC\_InitStructure to one.
3. Set nvic\_irq\_channel\_subpriority of NVIC\_InitStructure to two.
4. Set nvic\_irq\_channel\_cmd of NVIC\_InitStructure to ENABLE.
5. Call the function NvicInit with parameter reference to NVIC\_InitStructure.
6. Set SPI\_Direction of SPI\_InitStructure to SPI\_Direction\_2Lines\_FullDuplex.
7. Set SPI\_Mode of SPI\_InitStructure to SPI\_Mode\_Master.
8. Set SPI\_DataSize of SPI\_InitStructure to SPI\_DataSize\_8b.
9. Set SPI\_CPOL of SPI\_InitStructure to SPI\_CPOL\_Low.
10. Set SPI\_CPHA of SPI\_InitStructure to SPI\_CPHA\_1Edge.
11. Set SPI\_NSS of SPI\_InitStructure to SPI\_NSS\_Soft.
12. Set SPI\_BaudRatePrescaler of SPI\_InitStructure to SPI\_BaudRatePrescaler\_16.
13. Set SPI\_FirstBit of SPI\_InitStructure to SPI\_FirstBit\_MSB.
14. Set SPI\_CRCPolynomial of SPI\_InitStructure to seven.
15. Call the function SPI\_Init with parameters M\_SPI2 and reference to SPI\_InitStructure.
16. Call the function SPI\_I2S\_ITConfig wit parameters (M\_SPI2, SPI\_I2S\_IT\_RXNE and ENABLE).
17. Call the function SPI\_Cmd with parameters M\_SPI2 and ENABLE.

## 10.24 dauanapwmc

This module is meant for Implementation of function to read tachometer.

### 10.24.1 PWMCRead

Low Level Design Details about CSU PWMCRead will follow in the sub sections.

#### 10.24.1.1 Brief Description

The function PWMCRead Calculate the sampled tach reading for the selected channel.

#### 10.24.1.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.24.1.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 10.24.1.4 Parameter list (Input/Output)

Inputs : T\_UINT8 u8Chan Tach channel no (TACH\_CHAN\_1 to TACH\_CHAN\_8)

Outputs : None

#### 10.24.1.5 Return Value

T\_FLOAT32 - Return the tach channel sampling reading value

#### 10.24.1.6 Other CSUs called by this CSU

RestoreStatusReg

SaveStatusReg

#### 10.24.1.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to PWMCRead.

##### 10.24.1.7.1 dauanapwmc-PWMCRead-LLR-001

Requirement ID: H398-LLD-ANA-FNC-2000

The function shall call SaveStatusReg[M\_OS\_ENTER\_CRITICAL] to save the status of register and set Buffer pointer to Pwm\_big\_buffer\_ptrs with index u8Chan.

##### 10.24.1.7.2 dauanapwmc-PWMCRead-LLR-002

Requirement ID: H398-LLD-ANA-FNC-2001

The function shall perform the following until M\_TACH\_BIG\_BUFFER\_SIZE minus one

* Set counts to counts plus Buffer pointer and increments valid samples when Buffer pointer is less than Pwm\_timer\_count\_limit.
* Set Buffer pointer to address of Pwm\_big\_buffer with index u8Chan and (M\_TACH\_BIG\_BUFFER\_SIZE minus M\_ONE) when

pre decrement of Buffer pointer is less than pwm big buffer with index u8Chan.

##### 10.24.1.7.3 dauanapwmc-PWMCRead-LLR-004

Requirement ID: H398-LLD-ANA-FNC-2439

The function shall perform the following

* Set average counts to counts divide by valid samples when valid samples is not equal to M\_ZERO.
* Set pwm value to multiplication of average counts and ( M\_FP\_THOUSAND divided by M\_TMR\_5\_CNTS).

##### 10.24.1.7.4 dauanapwmc-PWMCRead-LLR-003

Requirement ID: H398-LLD-ANA-FNC-2002

The function shall Call M\_OS\_EXIT\_CRITICAL and returns pwm\_value.

## 10.25 dauanatach

This module contains Implementation of dauanatach routines.

### 10.25.1 TACHRead

Low Level Design Details about CSU TACHRead will follow in the sub sections.

#### 10.25.1.1 Brief Description

Calculate the sampled tach reading for the selected channel.

#### 10.25.1.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.25.1.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 10.25.1.4 Parameter list (Input/Output)

Inputs: u8Chan

Outputs: none

#### 10.25.1.5 Return Value

tach\_value

#### 10.25.1.6 Other CSUs called by this CSU

SaveStatusReg

RestoreStatusReg

#### 10.25.1.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to TACHRead.

##### 10.25.1.7.1 dauanatach-TACHRead-LLR-001

Requirement ID: H398-LLD-ANA-FNC-2240

The function shall call the function M\_OS\_ENTER\_CRITICAL.

##### 10.25.1.7.2 dauanatach-TACHRead-LLR-002

Requirement ID: H398-LLD-ANA-FNC-2241

The function shall do the following:

1. Set ps big buffer to psBigBuffer of As\_tach\_chanwith index u8Chan

2. The loop counter is looping from M\_ZERO to M\_TACH\_BIG\_BUFFER\_SIZE minus one

3. Set ps big buffer to address of Aas\_big\_buffer with indexes channel, M\_TACH\_BIG\_BUFFER\_SIZE minus M\_ONE when

- Decremented ps big buffer is less than Aas\_big\_buffer with index channel other wise do nothing

4. Set timer counts to timer counts plus u16Counts of ps big buffer

5. Set samples to samples plus u16Samples of ps big buffer

6. Check the minimum resolution i.e stop the functionality when u32counts is grater than or equal to u32Resolution of As\_tach\_chan with index u8Chan Logical AND with u16loop ctr is greater than minbb, other wise do nothing.

##### 10.25.1.7.3 dauanatach-TACHRead-LLR-003

Requirement ID: H398-LLD-ANA-FNC-2242

1.The function shall call the M\_OS\_EXIT\_CRITICAL

- Set tach value to M\_ZERO when timer counts is less than u32Resolution of As\_tach\_chan with index channel other wise set tach value to (samples multiply with f32Scaler of As\_tach\_chan with index channel) divided by ( counts multiply with M\_TACH\_US\_PER\_COUNT)

-Return the tach value

### 10.25.2 TachDriver

Low Level Design Details about CSU TachDriver will follow in the sub sections.

#### 10.25.2.1 Brief Description

Install the tach driver (Load the data from DMA buffer to Tach buffer)

#### 10.25.2.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.25.2.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 10.25.2.4 Parameter list (Input/Output)

Inputs: None

Outputs: None

#### 10.25.2.5 Return Value

None

#### 10.25.2.6 Other CSUs called by this CSU

TimClearITPendingBit

#### 10.25.2.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to TachDriver

##### 10.25.2.7.1 dauanatach-TachDriver-LLR-001

Requirement ID: H398-LLD-ANA-FNC-2251

The function shall call the TimClearITPendingBit with parameters M\_TIM12 and M\_TIM\_IT\_UPDATE.

##### 10.25.2.7.2 dauanatach-TachDriver-LLR-002

Requirement ID: H398-LLD-ANA-FNC-2252

The function shall do nothing when u8ChanState of Sensor\_as\_tach with index channel of Ptr\_sensor\_aas\_analog is not equal to ENABLE until the channel from TACH\_CHAN\_1 to TACH\_CHANS minus one.

##### 10.25.2.7.3 dauanatach-TachDriver-LLR-003

Requirement ID: H398-LLD-ANA-FNC-2596

The function shall do the following until the channel from TACH\_CHAN\_1 to TACH\_CHANS minus one when u8ChanState of Sensor\_as\_tach with index channel of Ptr\_sensor\_aas\_analog is equal to ENABLE :

- Set pu16 last to (m0ar of psDMA of As\_tach\_chan with index channel plus (M\_TACH\_DMA\_BUFFER\_SIZE minus ndtr of psDMA of As\_tach\_chan with index channel

##### 10.25.2.7.4 dauanatach-TachDriver-LLR-004

Requirement ID: H398-LLD-ANA-FNC-2597

The function shall perform the following until the channel from TACH\_CHAN\_1 to TACH\_CHANS minus one when u8ChanState of Sensor\_as\_tach with index channel of Ptr\_sensor\_aas\_analog is equal to ENABLE:

- Set pu16 last to Aau16\_buffer\_dma with indexes channel M\_TACH\_DMA\_BUFFER\_SIZE minus M\_ONE when decremented pu16 last is less than Aau16\_buffer\_dma with index channel other wise do nothing.

##### 10.25.2.7.5 dauanatach-TachDriver-LLR-005

Requirement ID: H398-LLD-ANA-FNC-2598

The function shall perform the following until the channel from TACH\_CHAN\_1 to TACH\_CHANS minus one :

- Set reference pu16 last to addition of (m0ar of psDMA of As\_tach\_chan with index (u8 channel plus M\_ONE) and (M\_TACH\_DMA\_BUFFER\_SIZE minus ndtr of psDMA of As\_tach\_chan with index (u8 channel plus M\_ONE)))

- Set reference pu16 last to address of As\_tach\_chan with index (u8 channel plus M\_ONE) and (M\_TACH\_DMA\_BUFFER\_SIZE minus M\_ONE) when pre decreement of reference pu16 last is less than As\_tach\_chan with index (u8 channel plus M\_ONE) other wise do nothing.

when below conditions are satisfied :

- u8ChanState of Sensor\_as\_tach with index channel of Ptr\_sensor\_aas\_analog is equal to ENABLE

- PWM\_CHAN\_1 is equal to u8 channel logical OR with PWM\_CHAN\_3 is equal to u8 channel.

##### 10.25.2.7.6 dauanatach-TachDriver-LLR-006

Requirement ID: H398-LLD-ANA-FNC-2599

The function shall perform the following until the channel from TACH\_CHAN\_1 to TACH\_CHANS minus one :

- Set pwm length to reference pu16 last minus pu16 last

- Set pwm length to Pwm\_timer\_count\_limit plus pwm\_length when pwm length is less than 0

- Set pwm index to M\_ONE when PWM\_CHAN\_3 is equal to u8 channel

- Set pwm\_big\_buffer\_ptrs with index pwm index to pwm length

- Set pwm\_big\_buffer\_ptrs with index pwm index to pwm\_big\_buffer with index pwm index when pre increment of pwm\_big\_buffer\_ptrs with index pwm index is greater than or equal to address of pwm\_big\_buffer with index (pwm index and M\_TACH\_BIG\_BUFFER\_SIZE) otherwise do nothing.

when below conditions are satisfied :

- u8ChanState of Sensor\_as\_tach with index channel of Ptr\_sensor\_aas\_analog is equal to ENABLE

- PWM\_CHAN\_1 is equal to u8 channel logical OR with PWM\_CHAN\_3 is equal to u8 channel

- u8 Channel Status of As\_tach\_chan with index u8 channel is equal to ENABLE.

##### 10.25.2.7.7 dauanatach-TachDriver-LLR-007

Requirement ID: H398-LLD-ANA-FNC-2600

The function shall perform the following until the channel from TACH\_CHAN\_1 to TACH\_CHANS minus one :

- Set test to pu16\_last minus pu16First of As\_tach\_chan with index u8 channel and u16Samples of psBigBuffer of As\_tach\_chan with index u8 channel to pu16\_last minus pu16First of As\_tach\_chan with u8 index channel when pu16 last is grater than or equal to pu16First of As\_tach\_chan with index u8 channel Other wise

- Set u16Samples of psBigBuffer of As\_tach\_chan with index channel to

M\_TACH\_DMA\_BUFFER\_SIZE minus ( addition of pu16First of As\_tach\_chan with index u8 channel ,pu16 last and M\_ONE).

when below conditions are satisfied :

- u8ChanState of Sensor\_as\_tach with index channel of Ptr\_sensor\_aas\_analog is equal to ENABLE

- u8 Channel Status of As\_tach\_chan with index u8 channel is equal to ENABLE.

##### 10.25.2.7.8 dauanatach-TachDriver-LLR-008

Requirement ID: H398-LLD-ANA-FNC-2601

The function shall perform the following until the channel from TACH\_CHAN\_1 to TACH\_CHANS minus one :

- Set u16Samples of psBigBuffer of As\_tach\_chan with index u8 channel to pu16 last minus pu16First of As\_tach\_chan with index u8 channel when pu16 last is greater than or equal to pu16First of As\_tach\_chan with index u8 channel.

- Set u16Samples of psBigBuffer of As\_tach\_chan with index u8 channel to pu16 last plus M\_TACH\_DMA\_BUFFER\_SIZE minus pu16First of As\_tach\_chan with index u8 channel when pu16 last is less than or equal to pu16First of As\_tach\_chan with index u8 channel.

when below conditions are satisfied :

- u8ChanState of Sensor\_as\_tach with index channel of Ptr\_sensor\_aas\_analog is equal to ENABLE

- u8 Channel Status of As\_tach\_chan with index u8 channel is equal to ENABLE.

##### 10.25.2.7.9 dauanatach-TachDriver-LLR-009

Requirement ID: H398-LLD-ANA-FNC-2602

The function shall perform the following until the channel from TACH\_CHAN\_1 to TACH\_CHANS minus one :

- Set u16Counts of psBigBuffer of As\_tach\_chan with index u8 channel to M\_ZERO

- Set u16Samples of psBigBuffer of As\_tach\_chan with index u8 channel to M\_ZERO

when below conditions are satisfied :

- u8ChanState of Sensor\_as\_tach with index channel of Ptr\_sensor\_aas\_analog is equal to ENABLE

- u8 Channel Status of As\_tach\_chan with index u8 channel is not equal to ENABLE.

##### 10.25.2.7.10 dauanatach-TachDriver-LLR-010

Requirement ID: H398-LLD-ANA-FNC-2603

The function shall perform the following until the channel from TACH\_CHAN\_1 to TACH\_CHANS minus one :

- Set u8ChanStatus of As\_tach\_chan with index u8 channel to FALSE when pre increment of u16Timeout of As\_tach\_chan with index u8 channel is greater than or equal to M\_TACH\_DRIVER\_RATE otherwise do nothing.

when below conditions are satisfied :

- u8ChanState of Sensor\_as\_tach with index channel of Ptr\_sensor\_aas\_analog is equal to ENABLE

- pu16 last is equal to pu16First of As\_tach\_chan with index u8 channel

##### 10.25.2.7.11 dauanatach-TachDriver-LLR-011

Requirement ID: H398-LLD-ANA-FNC-2604

The function shall perform the following until the channel from TACH\_CHAN\_1 to TACH\_CHANS minus one :

- Set u16Timeout of As\_tach\_chan with index u8 channel to M\_ZERO

- Set u8ChanStatus of As\_tach\_chan with index u8 channel to TRUE

when below conditions are satisfied :

- u8ChanState of Sensor\_as\_tach with index channel of Ptr\_sensor\_aas\_analog is equal to ENABLE

- pu16 last is not equal to pu16First of As\_tach\_chan with index u8 channel

##### 10.25.2.7.12 dauanatach-TachDriver-LLR-012

Requirement ID: H398-LLD-ANA-FNC-2605

The function shall perform the following until the channel from TACH\_CHAN\_1 to TACH\_CHANS minus one when u8ChanState of Sensor\_as\_tach with index channel of Ptr\_sensor\_aas\_analog is equal to ENABLE :

- Set pu16First of As\_tach\_chan with index u8 channel to pu16 last.

##### 10.25.2.7.13 dauanatach-TachDriver-LLR-013

Requirement ID: H398-LLD-ANA-FNC-2606

The function shall perform the following until the channel from TACH\_CHAN\_1 to TACH\_CHANS minus one :

- Set psBigBuffer of As\_tach\_chan with index u8 channel to Aas\_big\_buffer with index u8 channel when pre increment of psBigBuffer of As\_tach\_chan with index u8 channel is greater than or equal to address of Aas\_big\_buffer with index u8 channel and M\_TACH\_BIG\_BUFFER\_SIZE.

when below conditions are satisfied :

- u8ChanState of Sensor\_as\_tach with index channel of Ptr\_sensor\_aas\_analog is equal to ENABLE

### 10.25.3 TACHInit

Low Level Design Details about CSU TACHInit will follow in the sub sections.

#### 10.25.3.1 Brief Description

Initialize the tach channel and DMA buffer for the channel to start communication

#### 10.25.3.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.25.3.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 10.25.3.4 Parameter list (Input/Output)

Inputs: None

Outputs: None

#### 10.25.3.5 Return Value

None

#### 10.25.3.6 Other CSUs called by this CSU

InitTachChan

TMRInitTIM12

#### 10.25.3.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to TACHInit

##### 10.25.3.7.1 dauanatach-TACHInit-LLR-001

Requirement ID: H398-LLD-ANA-FNC-2261

The function shall Do nothing when u8ChanState of Sensor\_as\_tach with index loop counter of Ptr\_sensor\_aas\_analog set to not equal to ENABLE until the channel from TACH\_CHAN\_1 to TACH\_CHANS minus one.

##### 10.25.3.7.2 dauanatach-TACHInit-LLR-002

Requirement ID: H398-LLD-ANA-FNC-2262

The function shall perform the following until the channel from TACH\_CHAN\_1 to TACH\_CHANS minus one when u8ChanState of Sensor\_as\_tach with index loop counter of Ptr\_sensor\_aas\_analog equal to ENABLE :

- Set f32 maxmimun frequency to (f32 Referemce Frequency of Sensor\_as\_tach with index loop counter of Ptr\_sensor\_aas\_analog multiply with (u16MaxRange of Sensor\_as\_tach with index loop counter of Ptr\_sensor\_aas\_analog) divided by u16Scaled of Sensor\_as\_tach with index Loop counter of Ptr\_sensor\_aas\_analog.

##### 10.25.3.7.3 dauanatach-TACHInit-LLR-003

Requirement ID: H398-LLD-ANA-FNC-2607

The function shall perform the following until the channel from TACH\_CHAN\_1 to TACH\_CHANS minus one and until M\_ONE to M\_TACH\_MAX\_DIV - M\_ONE.

- Break from loop when f32 maxmimun frequency divide by u8 requency divider is less than or equal to M\_TACH\_5\_KHZ otherwise do nothing.

when below conditions are satisfied :

- u8ChanState of Sensor\_as\_tach with index loop counter of Ptr\_sensor\_aas\_analog equal to ENABLE.

##### 10.25.3.7.4 dauanatach-TACHInit-LLR-004

Requirement ID: H398-LLD-ANA-FNC-2608

The function shall perform the following until the channel from TACH\_CHAN\_1 to TACH\_CHANS minus one when u8ChanState of Sensor\_as\_tach with index loop counter of Ptr\_sensor\_aas\_analog equal to ENABLE :

* Set f32Scaler of As\_tach\_chan with index loop counter to (u16Scaled of Sensor\_as\_tach with index loop counter of Ptr\_sensor\_aas\_analog multiply with M\_SEC\_TO\_USEC\_UNIT) divided with (f32 Reference Frequency of Sensor\_as\_tach with index Ptr\_sensor\_aas\_analog divided with frequency divider.
* Set u32Resolution of As\_tach\_chan with index loop counter to M\_SAMPLING\_PERIOD Multiply with u16MaxRange of Sensor\_as\_tach with index loop counter of Ptr\_sensor\_aas\_analog.
* Call the function InitTachChan with index u8frequency divider , Aau16\_buffer\_dma with index u8loop counter,M\_TACH\_DMA\_BUFFER\_SIZE of As\_tach\_chan with index u8loop counter to Initialize tach channel.

##### 10.25.3.7.5 dauanatach-TACHInit-LLR-005

Requirement ID: H398-LLD-ANA-FNC-2609

The function shall Call TMRInitTIM12 with parameters TachDriver.

## 10.26 dauanaxdac

The Module implements the DAC routines for 6V OUT Analog communication.

### 10.26.1 XDACinit

Low Level Design Details about CSU XDACinit will follow in the sub sections.

#### 10.26.1.1 Brief description

This function will Install task signaling parameters and create the application task.

#### 10.26.1.2 List of HLR's allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.26.1.3 List of global variables accessed and modified

Accessed: None

Modified: None

#### 10.26.1.4 Parameter List (Input/Output)

None

#### 10.26.1.5 Return Value

None

#### 10.26.1.6 Other CSUs called by this CSU

TbaseTaskSignaling

OsTaskCreate

OsSemCreate

#### 10.26.1.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to XDACinit task

##### 10.26.1.7.1 dauanaxdac-XDACinit-LLR-001

Requirement ID: H398-LLD-ANA-FNC-2272

This function shall do the following to initialize the semaphores:

1. Call TbaseTaskSignaling function by sending the parameters M\_DAC\_TASK\_TICKS (number of tasks per second set as 100), and l\_semXdacTask is set by return of OsSemCreate, semaphor for XDAC task after created and installed.
2. Call OsTaskCreate function for creating application task with parameters xdac task function, second parameter as M\_NULL,3rd parameter as address of the stack used for xdac task and last parameter as M\_DAC\_TASK\_PRIO.

### 10.26.2 XDACWrite

Low Level Design Details about CSU XDACwrite will follow in the sub sections.

#### 10.26.2.1 Brief description

This function will write the XADC buffer to SPI device after swapping the buffer.

#### 10.26.2.2 List of HLR's allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.26.2.3 List of global variables accessed and modified

Accessed: None

Modified: None

#### 10.26.2.4 Parameter List (Input/Output)

Input -> T\_UINT16 u16WriteValue -> unsigned 16 value sent for little and big endian.

Output -> T\_UINT16 u16WriteValue -> unsigned 16 value sent for little and big endian.

#### 10.26.2.5 Return value

None

#### 10.26.2.6 Other CSUS called by this CSU

M\_SWAP16.

writeSPI2.

#### 10.26.2.7 Description of list of LLRS allocated

The following section will list the LLRs allocated to XDACwrite task.

##### 10.26.2.7.1 dauanaxdac-XDACWrite-LLR-001

Requirement ID: H398-LLD-ANA-FNC-2281

This function shall write XADC buffer to SPI device as per following steps:

1. Call M\_SWAP16 function for swapping the 2 bytes between little endian and big endian

and assign it to the SPI buffer.

2) Write 2 bytes of swapped SPI buffer to spi device by calling writeSPI2 Function.

### 10.26.3 SerialIntrSPI2

Low Level Design Details about CSU serialIntrSPI2 will follow in the sub sections

#### 10.26.3.1 Brief Description

This function recieves dummy data from SPI and send the data across SPI bus

#### 10.26.3.2 List of HLR's allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.26.3.3 List of global variables accessed and modified

Accessed: None

Modified: None

#### 10.26.3.4 Parameter List (Input/Output)

None

#### 10.26.3.5 Return Value

None

#### 10.26.3.6 Other CSUs called by this CSU

OsIntEnter.

SPII2SReceiveData.

IntrInstall.

SSPIfinishSPI2.

SSPIreleaseBusSPI2.

SPII2SSendData.

OsIntExit.

#### 10.26.3.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to serialIntrSPI2task.

##### 10.26.3.7.1 dauanaxdac-SerialIntrSPI2-LLR-001

Requirement ID: H398-LLD-ANA-FNC-2290

This function serialIntrSPI2write shall perform the following:

1. Call the function OsIntEnter.
2. Call the function SPII2SReceiveData with M\_SPI2 sent as parameter.

##### 10.26.3.7.2 dauanaxdac-SerialIntrSPI2-LLR-002

Requirement ID: H398-LLD-ANA-FNC-2291

When the incremented value of the data count of the DACcomm buffer is greater than or equal to

Data size of the same buffer following shall be done:

1. Call the function M\_HW\_XDAC\_CS\_HIGH.
2. Load the interrupt by calling IntrInstall function by passing parameters INTR\_SPI\_2(SPI2 interrupt) and place holder for interrupt of SPI2.
3. Call SSPIfinishSPI2 function indicating transferring SPI2 interrupt is complete On SPI bus.
4. Release the SPI bus after transfer is complete by calling SSPIreleaseBusSPI2 function.

##### 10.26.3.7.3 dauanaxdac-SerialIntrSPI2-LLR-003

Requirement ID: H398-LLD-ANA-FNC-2292

When the incremented value of the data count of the DACcomm buffer is less than data size of the same buffer follwing shall be done:

Call the function SPII2SSendData indicating writing data on SPI2 bus by sending parameters M\_SPI2 and sum of data buffer used for SPI2 transmission, and the data count transmitted of the DACcomm buffer.

##### 10.26.3.7.4 dauanaxdac-SerialIntrSPI2-LLR-004

Requirement ID: H398-LLD-ANA-FNC-2293

Once transferring SPI2 interrupt over the SPI bus is completed OsIntExit function shall be called.

### 10.26.4 WriteSPI2

Low Level Design Details about CSU writeSPI2 will follow in the sub sections

#### 10.26.4.1 Brief description

This function performs following: - Send or write data on SPI2.

#### 10.26.4.2 List of HLR's allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 10.26.4.3 List of global variables accessed and modified.

None

#### 10.26.4.4 Parameter List (Input/Output)

Input : T\_UINT16 u16DataSize :- specifies data size of buffer.

T\_UINT8 pu8Buffer : address of the source array.

Output : T\_UINT8 pu8Buffer :- address of the source array.

T\_UINT16 u16DataSize : specifies data size of buffer.

#### 10.26.4.5 Return values

None

#### 10.26.4.6 Other CSUs called by this CSU

SSPIclaimBusSPI2

SPII2SSendData

SSPIwaitSPI2

IntrGetCurrentIntrFn

IntrInstall

#### 10.26.4.7 Description of the List of LLRs allocated

The following section will list the LLRs allocated to writeSPI2task.

##### 10.26.4.7.1 dauanaxdac-WriteSPI2-LLR-001

Requirement ID: H398-LLD-ANA-FNC-2302

This function performs the following:

* Call the function SSPIclaimBusSPI2 indicating claim the SPI2 bus.
* Save the data size to of data to the data size variable of the DACcomm buffer.
* Reset the data count to M\_ZERO.
* Save the buffer pointer to the buffer pointer of the DACcomm buffer.
* L\_FnOldIntrSPI2 is set by return of function IntrGetCurrentIntrFn with parameter INTR\_SPI\_2.
* Call IntrInstall with parameters INTR\_SPI\_2 and serialIntrSPI2.
* Call function M\_HW\_XDAC\_CS\_LOW.
* Call SPI\_I2S\_SendData with parameters M\_SPI2 and pu8Buffer.
* Call the function SSPIwaitSPI2.

### 10.26.5 DoDacOutputs

Low Level Design Details about CSU DoDacOutputs will follow in the sub sections.

#### 10.26.5.1 Brief description

This function updates and outputs different values on DAC channels.

#### 10.26.5.2 List of HLR's allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to

SRS/SAD (H398-003-012-ANA).

#### 10.26.5.3 List of global variables accessed and modified

Accessed: Ptr\_sensor\_aas\_analog, App\_dac\_params, Cal\_aasdata

Modified: none

#### 10.26.5.4 Parameter List (Input/Output)

None

#### 10.26.5.5 Return values

None

#### 10.26.5.6 Other CSUs called by this CSU

XDAC\_write

LookupTableInverse\_32

GpioSetBits

GpioResetBits

#### 10.26.5.7 Description of the List of LLRs allocated

The following section will list the LLRs allocated to DoDacOutputs

##### 10.26.5.7.1 dauanaxdac-DoDacOutputs-LLR-001

Requirement ID: H398-LLD-ANA-FNC-2311

This function shall performs the following:

* Timer is set to M\_ZERO when time exceeds more than or equal to M\_FORTY.

##### 10.26.5.7.2 dauanaxdac-DoDacOutputs-LLR-002

Requirement ID: H398-LLD-ANA-FNC-2312

This function shall performs the following:

* Timer is incremented when Timer is less than M\_THIRTY.

##### 10.26.5.7.3 dauanaxdac-DoDacOutputs-LLR-003

Requirement ID: H398-LLD-ANA-FNC-2313

This function shall performs the following:

1. Call GpiosetBits with parameters M\_GPIOB and M\_GPIOB\_DECODER1 function is called (M\_HW\_DECODER1\_HIGH).

2. Call  GpiosetBits with parameters M\_GPIOB andM\_GPIOB\_DECODER2 function is called (M\_HW\_DECODER2\_ HIGH).

3. Call  GpioResetBits with parameters M\_GPIOB and M\_GPIOB\_DECODER3 function is called (M\_HW\_DECODER3\_LOW).

4. Dac is updated by calling LookupTableInverse32 using the parameters s.table of Sensor\_as\_dacof index size of DAC\_CHAN\_4 of Ptr\_sensor\_aas\_analog and the parameter App\_dac\_params of index size of DAC\_CHAN\_4 and the variable u8 status

5. Dac value is updated to addition of (product of (f32\_Gain of calDAC of index size of DAC\_CHAN\_4 of Cal\_aasdata and dac\_value) and i16\_Offset of calDAC of index size of DAC\_CHAN\_4 of Cal\_aasdata) when f32\_Gain of calDAC of index size of DAC\_CHAN\_4 of Cal\_aasdata is greater than M\_ZERO .

When the below condition is satisfied:

- The variable Timer is less than M\_TEN.

##### 10.26.5.7.4 dauanaxdac-DoDacOutputs-LLR-004

Requirement ID: H398-LLD-ANA-FNC-2314

This function shall performs the following:

1. Call GpioResetBits with parameters M\_GPIOB and M\_GPIOB\_DECODER1 function is called (M\_HW\_DECODER1\_LOW).

2. Call GpioResetBits with parameters M\_GPIOB and M\_GPIOB\_DECODER2 function is called (M\_HW\_DECODER2\_LOW).

3. Call GpiosetBits with parameters M\_GPIOB and M\_GPIOB\_DECODER3 function is called (M\_HW\_DECODER3\_ HIGH).

4.Dac is updated by calling LookupTableInverse32 using the parameters s.table of Sensor\_as\_dacof index size of DAC\_CHAN\_5 of Ptr\_sensor\_aas\_analog and the parameter App\_dac\_params of index size of DAC\_CHAN\_5 and the variable u8 status

5. Dac value is updated to addition of (product of (f32\_Gain of calDAC of index size of DAC\_CHAN\_5 of Cal\_aasdata and dac\_value) and i16\_Offset of calDAC of index size of DAC\_CHAN\_5 of Cal\_aasdata) when f32\_Gain of calDAC of index size of DAC\_CHAN\_5 of Cal\_aasdata is greater than M\_ZERO .

When the below condition is satisfied:

- The variable Timer is greater than or equal to M\_TEN Logical AND with timer is less than M\_TWENTY.

##### 10.26.5.7.5 dauanaxdac-DoDacOutputs-LLR-005

Requirement ID: H398-LLD-ANA-FNC-2315

This function shall perform the following When the variable Timer is greater than or equal to M\_TWENTY AND the variable Timer is less than M\_THIRTY:

1. Call GpiosetBits with parameters M\_GPIOB and M\_GPIOB\_DECODER1 function is called (M\_HW\_DECODER1\_ HIGH).

2. Call GpioResetBits with parameters M\_GPIOB and M\_GPIOB\_DECODER2 function is called (M\_HW\_DECODER2\_LOW).

3. Call GpiosetBits with parameters M\_GPIOB and M\_GPIOB\_DECODER3 function is called (M\_HW\_DECODER3\_ HIGH).

4. Dac is updated by calling LookupTableInverse32 using the parameters s.table of Sensor\_as\_dacof index size of DAC\_CHAN\_6 of Ptr\_sensor\_aas\_analog and the parameter App\_dac\_params of index size of DAC\_CHAN\_6 and the address of variable u8 status

5. Dac value is updated to addition of (product of (f32\_Gain of calDAC of index size of DAC\_CHAN\_6 of Cal\_aasdata and dac\_value) and i16\_Offset of calDAC of index size of DAC\_CHAN\_6 of Cal\_aasdata) when f32\_Gain of calDAC of index size of DAC\_CHAN\_6 of Cal\_aasdata is greater than M\_ZERO .

##### 10.26.5.7.6 dauanaxdac-DoDacOutputs-LLR-007

Requirement ID: H398-LLD-ANA-FNC-2447

The function shall perform the following when the variable Timer is greater than or equal to M\_THIRTY:

1. Call GpiosetBits with parameters M\_GPIOB and M\_GPIOB\_DECODER1 function is called (M\_HW\_DECODER1\_LOW).

2. Call GpioResetBits with parameters M\_GPIOB and M\_GPIOB\_DECODER2 function is called (M\_HW\_DECODER2\_HIGH).

3. Call GpiosetBits with parameters M\_GPIOB and M\_GPIOB\_DECODER3 function is called (M\_HW\_DECODER3\_ HIGH).

4. Dac is updated by calling LookupTableInverse32 using the parameters s.table of Sensor\_as\_dacof index size of DAC\_CHAN\_7 of Ptr\_sensor\_aas\_analog and the parameter App\_dac\_params of index size of DAC\_CHAN\_7 and the variable u8 status

5. Dac value is updated to addition of (product of (f32\_Gain of calDAC of index size of DAC\_CHAN\_7 of Cal\_aasdata and dac\_value) and i16\_Offset of calDAC of index size of DAC\_CHAN\_7 of Cal\_aasdata) when f32\_Gain of calDAC of index size of DAC\_CHAN\_7 of Cal\_aasdata is greater than M\_ZERO .

Otherwise, Do nothing.

##### 10.26.5.7.7 dauanaxdac-DoDacOutputs-LLR-006

Requirement ID: H398-LLD-ANA-FNC-2316

This function shall perform the following:

- XDACWrite is called with parameter of DAC value

### 10.26.6 XdacTask

Low Level Design Details about CSU XdacTask will follow in the sub sections.

#### 10.26.6.1 Brief description

This function waits for OsSemPend and calls DoDacOutputs.

#### 10.26.6.2 List of HLR's allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to

SRS/SAD (H398-003-012-ANA).

#### 10.26.6.3 List of global variables accessed and modified

None

#### 10.26.6.4 Parameter List (Input/Output)

Input : void \*pData

Output : none

#### 10.26.6.5 Return values

None

#### 10.26.6.6 Other CSUs called by this CSU

OsSemPend

DoDacOutputs

#### 10.26.6.7 Description of the List of LLRs allocated

The following section will list the LLRs allocated to XdacTask

##### 10.26.6.7.1 dauanaxdac-XdacTask-LLR-001

Requirement ID: H398-LLD-ANA-FNC-2325

This function shall performs the following loop until TRUE:

1.Call the function OsSemPend using the parameters as L\_semXdacTask, zero and U8error

2.Call the function DoDacOutputs

# 11 Software Low Level Requirements-Analog Module Library

This section specifies the Software Low Level Requirements for Discrete-STM Library.

## 11.1 daulibmisc

This module provides all the miscellaneous firmware functions (add-on to CMSIS functions).

### 11.1.1 NvicPriorityGroupConfig

Low Level Design Details about CSU NvicPriorityGroupConfig will follow in the sub sections.

#### 11.1.1.1 Brief Description

The function NvicPriorityGroupConfig configures the priority grouping: pre-emption priority and sub priority.

#### 11.1.1.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 11.1.1.3 List of global variables accessed and modified

Accessed: None

Modified: None

#### 11.1.1.4 Parameter list (Input/Output)

Inputs: T\_UINT32 nvic\_priority\_group - specifies the priority grouping bits length.

Outputs: None

#### 11.1.1.5 Return Value

None

#### 11.1.1.6 Other CSUs called by this CSU

None

#### 11.1.1.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to NvicPriorityGroupConfig.

##### 11.1.1.7.1 daulibmisc-NvicPriorityGroupConfig-LLR-001

Requirement ID: H398-LLD-ANA-FNC-1159

The function shall set the priority group bits [10:8] in aircr (Application interrupt and reset control register) of M\_SCB(System control block) by performing Bitwise OR of M\_AIRCR\_VECTKEY\_MASK and nvic\_priority\_group value.

### 11.1.2 NvicInit

Low Level Design Details about CSU NvicInit will follow in the sub sections.

#### 11.1.2.1 Brief Description

The function NvicInit initializes the NVIC peripheral according to the specifiedparameters in the nvic\_init\_struct.

#### 11.1.2.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 11.1.2.3 List of global variables accessed and modified

Accessed: None

Modified: None

#### 11.1.2.4 Parameter list (Input/Output)

Inputs: T\_NVIC\_INIT \* nvic\_init\_struct - pointer to a T\_NVIC\_INIT structure that contains the configuration information for the specified NVIC peripheral.

Outputs: None

#### 11.1.2.5 Return Value

None

#### 11.1.2.6 Other CSUs called by this CSU

None

#### 11.1.2.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to NvicInit.

##### 11.1.2.7.1 daulibmisc-NvicInit-LLR-001

Requirement ID: H398-LLD-ANA-FNC-1168

The function shall compute the interrupt priority and set the computed value to Interrupt Priority Register (IP) of NVIC if NVIC interrupt channel command (nvic\_irq\_channel\_cmd of nvic\_init\_struct) is equal to ENABLE i.e,

When DISABLE is not equal to nvic\_irq\_channel\_cmd of nvic\_init\_struct then compute the Corresponding IRQ Priority as per below logic:

1. Extracted Priority Group bits value (Priority group bits [10:8] of Application interrupt and reset control register) is extracted as M\_PRIORITY\_GROUP subtracted by ((aircr of M\_SCB) bitwise AND with M\_PRIORITY\_GROUP)) right shifted by M\_SHIFT\_BY\_8)
2. Preempt Priority value  is set to M\_FOUR subtracted by Extracted Priority Group bits value.
3. Sub Priority value is set to M\_HEX\_FIFTEEN right shift by Extracted Priority Group bits value
4. NVIC IRQ channel preemption priority value is set to (nvic\_irq\_channel\_preemption\_priority of nvic\_init\_struct LEFT\_SHIFT Preempt Priority value)
5. NVIC IRQ channel subpriority value is set to NVIC IRQ channel preemption priority value bitwise OR with (nvic\_irq\_channel\_subpriority of nvic\_init\_struct bitwise AND with Sub Priority value)
6. Priority Value is set to NVIC IRQ channel subpriority value left shift by M\_SHIFT\_BY\_4
7. ip\_reg of index (nvic\_irq\_channel of nvic\_init\_struct) of M\_NVIC is set to Priority Value.

Note: Refer DM00046982-with FPU-ref-manual.pdf for computing interrupt priority.

##### 11.1.2.7.2 daulibmisc-NvicInit-LLR-002

Requirement ID: H398-LLD-ANA-FNC-1169

The function shall configure interrupt set-enable register of NVIC to enable the selected interrupt channel by setting iser of M\_NVIC with index value (nvic\_irq\_channel of nvic\_init\_struct right shift by M\_SHIFT\_BY\_5) to (M\_SHIFT\_BY\_1 left shift by (nvic\_irq\_channel of nvic\_init\_struct   bitwise AND with M\_THIRTY\_ONE)).

##### 11.1.2.7.3 daulibmisc-NvicInit-LLR-003

Requirement ID: H398-LLD-ANA-FNC-1170

The function shall disable the selected interrupt channels by setting Interrupt clear-enable register for selected interrupt channel when NVIC interrupt channel command is disabled i.e,

Set icer of M\_NVIC with index value (nvic\_irq\_channel of nvic\_init\_struct right shift by M\_SHIFT\_BY\_5) to M\_SHIFT\_BY\_1 left shift by (nvic\_irq\_channel of nvic\_init\_struct bitwise AND with M\_THIRTY\_ONE) when DISABLE is equal to nvic\_irq\_channel\_cmd of nvic\_init\_struct.

### 11.1.3 NvicSetVectorTable

Low Level Design Details about CSU NvicSetVectorTable will follow in the sub sections.

#### 11.1.3.1 Brief Description

The function NvicSetVectorTable sets the vector table location and Offset.

#### 11.1.3.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 11.1.3.3 List of global variables accessed and modified

Accessed: None

Modified: None

#### 11.1.3.4 Parameter list (Input/Output)

Inputs: T\_UINT32 nvic\_vect\_tab - specifies if the vector table is in RAM or FLASH memory.

T\_UINT32 offset - Vector Table base offset field.

Outputs: None

#### 11.1.3.5 Return Value

None

#### 11.1.3.6 Other CSUs called by this CSU

None

#### 11.1.3.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to NvicSetVectorTable.

##### 11.1.3.7.1 daulibmisc-NvicSetVectorTable-LLR-001

Requirement ID: H398-LLD-ANA-FNC-1179

The function shall configure vector table offset register of system control block with vector table location (RAM or FLASH, bit 29) and offset (bits [29:7]) by setting vtor of M\_SCB to (nvic\_vect\_tab bitwise OR with (offset bitwise AND with M\_NVIC\_VTOR)).

## 11.2 daulibstm32f4xxadc

This module provides firmware functions to manage the following functionalities of the Analog to Digital Convertor (ADC) peripheral:

* Initialization and Configuration
* Regular Channels Configuration
* Regular Channels DMA Configuration.

### 11.2.1 AdcInit

Low Level Design Details about CSU AdcInit will follow in the sub sections.

#### 11.2.1.1 Brief Description

The function AdcInit initializes the ADC peripheral according to the specified parameters in the initialization structure.

#### 11.2.1.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 11.2.1.3 List of global variables accessed and modified

Accessed: None

Modified: None

#### 11.2.1.4 Parameter list (Input/Output)

Inputs: T\_ADC \*adc\_x - Where x can be 1, 2 or 3 to select the ADC peripheral.

T\_ADC\_INIT \*adc\_init\_struct - Pointer to a T\_ADC\_INIT structure that contains the configuration information for the specified ADC peripheral.

Outputs: T\_ADC \*adc\_x - where x can be 1, 2 or 3 to select the ADC peripheral.

#### 11.2.1.5 Return Value

None

#### 11.2.1.6 Other CSUs called by this CSU

None

#### 11.2.1.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to AdcInit.

##### 11.2.1.7.1 daulibstm32f4xxadc-AdcInit-LLR-001

Requirement ID: H398-LLD-ANA-FNC-1189

The function shall clear the RES (resolution) and SCAN (scan mode) bits of ADC control register 1 and configure with adc\_scan\_convmode and adc\_resolution of adc\_init\_struct as follows:

* + Temp Register is initialized to M\_ZERO.
  + Temp Register 2 is initialized to M\_ZERO.
  + Set Temp Register to cr1 of adc\_x.
  + Set Temp Register to Temp Register bitwise AND with M\_CR1\_CLEAR\_MASK
  + Set Temp Register to Temp Register bitwise OR with ((adc\_scan\_convmode of adc\_init\_struct left shift by M\_EIGHT) bitwise OR with adc\_resolution of adc\_init\_struct)
  + Set cr1 of adc\_x to Temp Register.
  + Set Temp Register to cr2 of adc\_x.

##### 11.2.1.7.2 daulibstm32f4xxadc-AdcInit-LLR-002

Requirement ID: H398-LLD-ANA-FNC-1190

The function shall clear the CONT, ALIGN, EXTEN and EXTSEL bits of ADC control register 2 and configure with adc\_dataalign, adc\_external\_trigconv, adc\_external\_trig\_convedge and adc\_continuous\_convmode of adc\_init\_struct i.e,

* + Set Temp Register to Temp Register bitwise AND with M\_CR2\_CLEAR\_MASK.
  + Set Temp Register to Temp Register bitwise OR with (adc\_dataalign of adc\_init\_struct bitwise OR with adc\_external\_trigconv of adc\_init\_struct bitwise OR with adc\_external\_trig\_convedge of adc\_init\_struct bitwise OR with (adc\_continuous\_convmode of adc\_init\_struct left shift by M\_ONE)).
  + Set cr2 of adc\_x to Temp Register.
  + Set Temp Register to sqr1 of adc\_x.

##### 11.2.1.7.3 daulibstm32f4xxadc-AdcInit-LLR-003

Requirement ID: H398-LLD-ANA-FNC-1191

The function shall clear the L (Regular channel sequence length) bits of ADC regular sequence register 1 and configure total number of conversions in the regular channel conversion sequence i.e,

* + Set Temp Register to (Temp Register bitwise AND with M\_SQR1\_L\_RESET).
  + Set Temp Register 2 to Temp Register 2 bitwise OR with (adc\_nbr\_of\_conversion of adc\_init\_struct subtracted by M\_ONE).
  + Set Temp Register to Temp Register bitwise OR with (Temp Register 2 left shift by M\_SHIFT\_20).
  + Set sqr1 of adc\_x to Temp Register.

### 11.2.2 AdcCommonInit

Low Level Design Details about CSU AdcCommonInit will follow in the sub sections.

#### 11.2.2.1 Brief Description

The function AdcCommonInit initializes the ADCs peripherals according to the specified parameters in the ADC initialization structure.

#### 11.2.2.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 11.2.2.3 List of global variables accessed and modified

Accessed: None

Modified: None

#### 11.2.2.4 Parameter list (Input/Output)

Inputs: T\_ADC\_COMMON\_INIT \*adc\_common\_init\_struct - pointer to a T\_ADC\_COMMON\_INIT structure that contains the configuration information for all ADCs peripherals.

Outputs: None

#### 11.2.2.5 Return Value

None

#### 11.2.2.6 Other CSUs called by this CSU

None

#### 11.2.2.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to AdcCommonInit.

##### 11.2.2.7.1 daulibstm32f4xxadc-AdcCommonInit-LLR-001

Requirement ID: H398-LLD-ANA-FNC-1200

The function shall clear the MULTI, DELAY, DMA and ADCPRE bits of ADC common control register and configure with ADC mode, prescaler, dma access mode and adc sampling delay.

* + Set Temp Register to ccr of M\_ADC.
  + Set Temp Register to (Temp Register bitwise AND with M\_CR\_CLEAR\_MASK.
  + Set Temp Register to (Temp Register bitwise OR with (adc\_mode of adc\_common\_init\_struct bitwise OR with adc\_prescaler of adc\_common\_init\_struct bitwise OR with adc\_dma\_access\_mode of adc\_common\_init\_struct bitwise OR with adc\_two\_sampling\_delay of adc\_common\_init\_struct)).
  + Set ccr of M\_ADC to Temp Register.

### 11.2.3 AdcCmd

Low Level Design Details about CSU AdcCmd will follow in the sub sections.

#### 11.2.3.1 Brief Description

The function AdcCmd enables or disables the specified ADC peripheral.

#### 11.2.3.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 11.2.3.3 List of global variables accessed and modified

Accessed: None

Modified: None

#### 11.2.3.4 Parameter list (Input/Output)

Inputs: T\_ADC \*adc\_x - Where x can be 1, 2 or 3 to select the ADC peripheral.

T\_FUNCTIONAL\_STATE new\_state - New state of the adc\_x peripheral.

Outputs: T\_ADC \*adc\_x - Where x can be 1, 2 or 3 to select the ADC peripheral.

#### 11.2.3.5 Return Value

None

#### 11.2.3.6 Other CSUs called by this CSU

None

#### 11.2.3.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to AdcCmd.

##### 11.2.3.7.1 daulibstm32f4xxadc-AdcCmd-LLR-001

Requirement ID: H398-LLD-ANA-FNC-1209

The function shall set the ADON (A/D Converter ON/OFF) bit of ADC control register 2 to wake up the ADC from power down mode i.e, set cr2 of adc\_x to (cr2 of adc\_x bitwise OR with M\_ADC\_CR2\_ADON) when new\_state is equal to ENABLE.

##### 11.2.3.7.2 daulibstm32f4xxadc-AdcCmd-LLR-002

Requirement ID: H398-LLD-ANA-FNC-1210

The function shall reset the ADON (A/D Converter ON/OFF) bit of ADC control register 2 to disable the selected ADC peripheral i.e, set cr2 of adc\_x to (cr2 of adc\_x bitwise AND with (negated value of M\_ADC\_CR2\_ADON)) when new\_state is equal to DISABLE.

### 11.2.4 AdcRegularChannelConfig

Low Level Design Details about CSU AdcRegularChannelConfig will follow in the sub sections.

#### 11.2.4.1 Brief Description

The function AdcRegularChannelConfig configures for the selected ADC regular channel, its corresponding rank in the sequencer and its sample time.

#### 11.2.4.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 11.2.4.3 List of global variables accessed and modified

Accessed: None

Modified: None

#### 11.2.4.4 Parameter list (Input/Output)

Inputs: T\_ADC \*adc\_x – Where x can be 1, 2 or 3 to select the ADC peripheral.

T\_UINT8 adc\_channel - The ADC channel to configure.

T\_UINT8 rank -The rank in the regular group sequencer.

T\_UINT8 adc\_sampletime - The sample time value to be set for the selected channel.

Outputs: T\_ADC \*adc\_x - Where x can be 1, 2 or 3 to select the ADC peripheral.

#### 11.2.4.5 Return Value

None

#### 11.2.4.6 Other CSUs called by this CSU

None

#### 11.2.4.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to AdcRegularChannelConfig.

##### 11.2.4.7.1 daulibstm32f4xxadc-AdcRegularChannelConfig-LLR-001

Requirement ID: H398-LLD-ANA-FNC-1219

The function shall clear the old sample time for the specified ADC channel bits in ADC sample time register 1 and configure the sample time when ADC channel selected (adc\_channel) is greater than 9 i.e,

Perform as follows when M\_ADC\_CHANNEL\_9 is less than adc\_channel

* + Set Temp Register to smpr1 of adc\_x.
  + Set Temp Register 2 to (M\_SMPR1\_SMP\_SET left shift by (M\_THREE multiplied by (adc\_channel subtracted by M\_TEN))).
  + Set Temp Register to (Temp Register bitwise AND with negated value of Temp register 2).
  + Set Temp Register 2 to (adc\_sampletime left shift by (M\_THREE multiplied by (adc\_channel subtracted by M\_TEN))).
  + Set Temp Register to (Temp Register bitwise OR with Temp Register 2).
  + Set smpr1 of adc\_x to Temp Register.

##### 11.2.4.7.2 daulibstm32f4xxadc-AdcRegularChannelConfig-LLR-002

Requirement ID: H398-LLD-ANA-FNC-1220

The function shall clear the old sample time for the specified ADC channel adc\_channel by clearing the SMPx (where x=adc\_channel) bits in ADC sample time register 2 and configure sample time when ADC channel selected is less than or equal to 9 i.e,

Perform as follows when M\_ADC\_CHANNEL\_9 is greater than or equal to adc\_channel

* + Set Temp Register to smpr2 of adc\_x.
  + Set Temp Register 2 to (M\_SMPR2\_SMP\_SET left shift by (M\_THREE multiplied by adc\_channel)).
  + Set Temp Register to (Temp Register bitwise AND with negated value of Temp register 2).
  + Set Temp Register 2 to (adc\_sampletime left shift by (M\_THREE multiplied by adc\_channel)).
  + Set Temp Register to (Temp Register bitwise OR with Temp Register 2).
  + Set smpr2 of adc\_x to Temp Register.

##### 11.2.4.7.3 daulibstm32f4xxadc-AdcRegularChannelConfig-LLR-003

Requirement ID: H398-LLD-ANA-FNC-1221

The function shall clear the sequence bits for the selected rank in ADC regular sequence register 3 and configure the rank when rank is less than 7 i.e,

Perform as follows when M\_SEVEN is greater than rank

* + Set Temp Register to sqr3 of adc\_x.
  + Set Temp Register 2 to (M\_SQR3\_SQ\_SET left shift by (M\_FIVE multiplied by (rank subtracted by M\_ONE))).
  + Set Temp Register to (Temp Register bitwise AND with negated value of Temp register 2).
  + Set Temp Register 2 to (adc\_channel left shift by (M\_FIVE multiplied by (rank subtracted by M\_ONE))).
  + Set Temp Register to (Temp Register bitwise OR with Temp Register 2).
  + Set sqr3 of adc\_x to Temp Register.

##### 11.2.4.7.4 daulibstm32f4xxadc-AdcRegularChannelConfig-LLR-004

Requirement ID: H398-LLD-ANA-FNC-1222

The function shall clear the sequence bits for the selected rank in ADC regular sequence register 2 and configure the rank when rank is greater than or equal to 7 and less than 13 i.e,

Perform as follows when rank is greater than or equal to M\_SEVEN AND less than M\_THIRTEEN

* + Set Temp Register to sqr2 of adc\_x.
  + Set Temp Register 2 to (M\_SQR2\_SQ\_SET left shift by (M\_FIVE multiplied by (rank subtracted by M\_SEVEN))).
  + Set Temp Register to (Temp Register bitwise AND with negated value of Temp register 2).
  + Set Temp Register 2 to (adc\_channel left shift by (M\_FIVE multiplied by (rank subtracted by M\_SEVEN))).
  + Set Temp Register to (Temp Register bitwise OR with Temp Register 2).
  + Set sqr2 of adc\_x to Temp Register.

##### 11.2.4.7.5 daulibstm32f4xxadc-AdcRegularChannelConfig-LLR-005

Requirement ID: H398-LLD-ANA-FNC-1223

The function shall clear the sequence bits for the selected rank in ADC regular sequence register 1 and configure the rank when rank is greater than or equal 13 i.e,

Perform as follows when rank is greater than or equal to M\_THIRTEEN

* + Set Temp Register to sqr1 of adc\_x.
  + Set Temp Register 2 to (M\_SQR1\_SQ\_SET left shift by (M\_FIVE multiplied by (rank subtracted by M\_THIRTEEN))).
  + Set Temp Register to (Temp Register bitwise AND with negated value of Temp register 2).
  + Set Temp Register 2 to (adc\_channel left shift by (M\_FIVE multiplied by (rank subtracted by M\_THIRTEEN))).
  + Set Temp Register to (Temp Register bitwise OR with Temp Register 2).
  + Set sqr1 of adc\_x to Temp Register.

### 11.2.5 AdcSoftwareStartConv

Low Level Design Details about CSU AdcSoftwareStartConv will follow in the sub sections.

#### 11.2.5.1 Brief Description

The function AdcSoftwareStartConv enables the selected ADC software start conversion of the regular channels.

#### 11.2.5.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 11.2.5.3 List of global variables accessed and modified

Accessed: None

Modified: None

#### 11.2.5.4 Parameter list (Input/Output)

Inputs: T\_ADC \*adc\_x - Where x can be 1, 2 or 3 to select the ADC peripheral.

Outputs: T\_ADC \*adc\_x - Where x can be 1, 2 or 3 to select the ADC peripheral.

#### 11.2.5.5 Return Value

None

#### 11.2.5.6 Other CSUs called by this CSU

None

#### 11.2.5.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to AdcSoftwareStartConv.

##### 11.2.5.7.1 daulibstm32f4xxadc-AdcSoftwareStartConv-LLR-001

Requirement ID: H398-LLD-ANA-FNC-1232

The function shall enable the selected ADC conversion for regular group by setting the SWSTART bit of ADC control register 2 i.e, set cr2 of adc\_x to cr2 of adc\_x bitwise OR with M\_ADC\_CR2\_SWSTART.

### 11.2.6 AdcDmaCmd

Low Level Design Details about CSU AdcDmaCmd will follow in the sub sections.

#### 11.2.6.1 Brief Description

The function AdcDmaCmd enables or disables the ADC DMA request.

#### 11.2.6.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 11.2.6.3 List of global variables accessed and modified

Accessed: None

Modified: None

#### 11.2.6.4 Parameter list (Input/Output)

Inputs: T\_ADC \*adc\_x - Where x can be 1, 2 or 3 to select the ADC peripheral.

T\_FUNCTIONAL\_STATE new\_state - New state of the selected ADC DMA transfer.

Outputs: T\_ADC \*adc\_x - Where x can be 1, 2 or 3 to select the ADC peripheral.

#### 11.2.6.5 Return Value

None

#### 11.2.6.6 Other CSUs called by this CSU

None

#### 11.2.6.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to AdcDmaCmd.

##### 11.2.6.7.1 daulibstm32f4xxadc-AdcDmaCmd-LLR-001

Requirement ID: H398-LLD-ANA-FNC-1241

The function shall enable the ADC DMA request by setting the DMA mode bit of ADC control register 2 when the new state of ADC peripheral is enable i.e, set cr2 of adc\_x to (cr2 of adc\_x bitwise OR with M\_ADC\_CR2\_DMA) when new\_state is equal to ENABLE.

##### 11.2.6.7.2 daulibstm32f4xxadc-AdcDmaCmd-LLR-002

Requirement ID: H398-LLD-ANA-FNC-1242

The function shall disable the ADC DMA request by resetting the DMA mode bit of ADC control register 2 when the new state of ADC peripheral is disable i.e, set cr2 of adc\_x to (cr2 of adc\_x bitwise AND with negated value of M\_ADC\_CR2\_DMA) when new\_state is equal to DISABLE.

### 11.2.7 AdcDmaReqAfterLastTransferCmd

Low Level Design Details about CSU AdcDmaReqAfterLastTransferCmd will follow in the sub sections.

#### 11.2.7.1 Brief Description

The function AdcDmaReqAfterLastTransferCmd enables or disables the ADC DMA request after last transfer (Single-ADC mode).

#### 11.2.7.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 11.2.7.3 List of global variables accessed and modified

Accessed: None

Modified: None

#### 11.2.7.4 Parameter list (Input/Output)

Inputs: T\_ADC \*adc\_x - Where x can be 1, 2 or 3 to select the ADC peripheral.

T\_FUNCTIONAL\_STATE new\_state - New state of the selected ADC DMA request after last transfer

Outputs: T\_ADC \* adc\_x - Where x can be 1, 2 or 3 to select the ADC peripheral.

#### 11.2.7.5 Return Value

None

#### 11.2.7.6 Other CSUs called by this CSU

None

#### 11.2.7.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to AdcDmaReqAfterLastTransferCmd.

##### 11.2.7.7.1 daulibstm32f4xxadc-AdcDmaReqAfterLastTransferCmd-LLR-001

Requirement ID: H398-LLD-ANA-FNC-1251

The function shall set the DDS (DMA disable selection) bit of ADC control register 2 when new state of Adc peripheral is enable i.e, set cr2 of adc\_x to (cr2 of adc\_x bitwise OR with M\_ADC\_CR2\_DDS).

##### 11.2.7.7.2 daulibstm32f4xxadc-AdcDmaReqAfterLastTransferCmd-LLR-002

Requirement ID: H398-LLD-ANA-FNC-1252

The function shall reset the DDS (DMA disable selection) bit of ADC control register 2 when new state of Adc peripheral is disable i.e, set cr2 of adc\_x to (cr2 of adc\_x bitwise AND with negated value of M\_ADC\_CR2\_DDS).

## 11.3 daulibstm32f4xxcan

This module provides firmware functions to manage the following functionalities of the Controller area network (CAN) peripheral:

* Initialization and Configuration
* CAN Frames Transmission
* CAN Frames Reception
* Interrupts and flags

### 11.3.1 CanDeInit

Low Level Design Details about CSU CanDeInit will follow in the sub sections.

#### 11.3.1.1 Brief Description

The function CanDeInit deinitializes the CAN peripheral registers to their default reset values.

#### 11.3.1.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 11.3.1.3 List of global variables accessed and modified

Accessed: None

Modified: None

#### 11.3.1.4 Parameter list (Input/Output)

Inputs: T\_CAN\_TYPEDEF \*can\_x - Where x can be 1 or 2 to select the CAN peripheral.

Outputs: None

#### 11.3.1.5 Return Value

None

#### 11.3.1.6 Other CSUs called by this CSU

RccApb1PeriphResetCmd

#### 11.3.1.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to CanDeInit.

##### 11.3.1.7.1 daulibstm32f4xxcan-CanDeInit-LLR-001

Requirement ID: H398-LLD-ANA-FNC-1262

The function shall do the following when M\_CAN1 is equal to can\_x:

a) Enable CAN1 reset state by calling the function RccApb1PeriphResetCmd with parameters M\_RCC\_APB1PERIPH\_CAN1 and ENABLE.

b) Release CAN1 from reset state by calling the function RccApb1PeriphResetCmd with parameters M\_RCC\_APB1PERIPH\_CAN1 and DISABLE .

##### 11.3.1.7.2 daulibstm32f4xxcan-CanDeInit-LLR-002

Requirement ID: H398-LLD-ANA-FNC-1836

The function shall do the following when M\_CAN1 is not equal to can\_x:

a) Enable CAN2 reset state by calling the function RccApb1PeriphResetCmd with parameters M\_RCC\_APB1PERIPH\_CAN2 and ENABLE

b) Release CAN2 from reset state by calling the function RccApb1PeriphResetCmd with parameters M\_RCC\_APB1PERIPH\_CAN2 and DISABLE.

### 11.3.2 CanInit

Low Level Design Details about CSU CanInit will follow in the sub sections.

#### 11.3.2.1 Brief Description

The function CanInit initializes the CAN peripheral according to the specified parameters in the CAN initialization structure.

#### 11.3.2.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 11.3.2.3 List of global variables accessed and modified

Accessed: None

Modified: None

#### 11.3.2.4 Parameter list (Input/Output)

Inputs: T\_CAN\_TYPEDEF\* can\_x - where x can be 1 or 2 to select the CAN peripheral.

T\_CAN\_INIT\* can\_init\_struct - pointer to a T\_CAN\_INIT structure that contains the configuration information for the CAN peripheral

Outputs: T\_CAN\_TYPEDEF\* can\_x - where x can be 1 or 2 to select the CAN peripheral.

#### 11.3.2.5 Return Value

Indicates initialization status which will be M\_CAN\_INITSTATUS\_FAILED or M\_CAN\_INITSTATUS\_SUCCESS.

#### 11.3.2.6 Other CSUs called by this CSU

None

#### 11.3.2.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to CanInit.

##### 11.3.2.7.1 daulibstm32f4xxcan-CanInit-LLR-001

Requirement ID: H398-LLD-ANA-FNC-1271

The function shall configure CAN master control register to exit from sleep mode (set bit 1 to 0) i.e, set mcr of can\_x to (mcr of can\_x bitwise AND with negated value of M\_CAN\_MCR\_SLEEP).

##### 11.3.2.7.2 daulibstm32f4xxcan-CanInit-LLR-002

Requirement ID: H398-LLD-ANA-FNC-1272

The function shall configure CAN master control register to request initialization (set bit 0 to 1) i.e, set mcr of can\_x to (mcr of can\_x bitwise OR with M\_CAN\_MCR\_INRQ).

##### 11.3.2.7.3 daulibstm32f4xxcan-CanInit-LLR-003

Requirement ID: H398-LLD-ANA-FNC-1273

The function shall loop till INAK bit in CAN master control register is 0 (M\_CAN\_MSR\_INAK is not equal to (msr of can\_x bitwise AND with M\_CAN\_MSR\_INAK)) AND loop count is not equal to M\_INAK\_TIMEOUT and increment the loop counter.

##### 11.3.2.7.4 daulibstm32f4xxcan-CanInit-LLR-004

Requirement ID: H398-LLD-ANA-FNC-1274

The function shall set the initialization status to M\_CAN\_INITSTATUS\_FAILED when INAK bit in CAN master control register is 0 (M\_CAN\_MSR\_INAK not equal to msr of can\_x bitwise AND with M\_CAN\_MSR\_INAK).

##### 11.3.2.7.5 daulibstm32f4xxcan-CanInit-LLR-005

Requirement ID: H398-LLD-ANA-FNC-1275

The function shall enable time triggered communication mode (set TTCM bit in master control register to 1) i.e, set mcr of can\_x to (mcr of can\_x bitwise OR with M\_CAN\_MCR\_TTCM) when can\_ttcm of can\_init\_struct is ENABLE and (M\_CAN\_MSR\_INAK is equal to (msr of can\_x bitwise AND with M\_CAN\_MSR\_INAK ) (INAK bit in msr of can\_x is 1)).

##### 11.3.2.7.6 daulibstm32f4xxcan-CanInit-LLR-006

Requirement ID: H398-LLD-ANA-FNC-1276

The function shall disable time triggered communication mode (reset TTCM bit in mcr of can\_x to 0) i.e, set mcr of can\_x to (mcr of can\_x bitwise AND with negated value of M\_CAN\_MCR\_TTCM) when can\_ttcm of can\_init\_struct is not equal to ENABLE and (M\_CAN\_MSR\_INAK is equal to (msr of can\_x bitwise AND with M\_CAN\_MSR\_INAK ) (INAK bit in msr of can\_x is 1)).

##### 11.3.2.7.7 daulibstm32f4xxcan-CanInit-LLR-007

Requirement ID: H398-LLD-ANA-FNC-1277

The function shall enable automatic bus-off management (set ABOM bit in master control register to 1) i.e, set mcr of can\_x to (mcr of can\_x bitwise OR with M\_CAN\_MCR\_ABOM) when can\_abom of can\_init\_struct is ENABLE and (M\_CAN\_MSR\_INAK is equal to (msr of can\_x bitwise AND with M\_CAN\_MSR\_INAK ) (INAK bit in msr of can\_x is 1)).

##### 11.3.2.7.8 daulibstm32f4xxcan-CanInit-LLR-008

Requirement ID: H398-LLD-ANA-FNC-1278

The function shall disable automatic bus-off management (reset ABOM bit in master control register to 0) i.e., set mcr of can\_x to (mcr of can\_x bitwise AND with negated value of M\_CAN\_MCR\_ABOM) when can\_abom of can\_init\_struct is DISABLE and (M\_CAN\_MSR\_INAK is equal to (msr of can\_x bitwise AND with M\_CAN\_MSR\_INAK ) (INAK bit in msr of can\_x is 1)).

##### 11.3.2.7.9 daulibstm32f4xxcan-CanInit-LLR-009

Requirement ID: H398-LLD-ANA-FNC-1279

The function shall enable automatic wake-up mode (set AWUM bit in master control register to 1) i.e, set mcr of can\_x to (mcr of can\_x bitwise OR with M\_CAN\_MCR\_AWUM) when can\_awum of can\_init\_struct is ENABLE and (M\_CAN\_MSR\_INAK is equal to (msr of can\_x bitwise AND with M\_CAN\_MSR\_INAK ) (INAK bit in msr of can\_x is 1)).

##### 11.3.2.7.10 daulibstm32f4xxcan-CanInit-LLR-010

Requirement ID: H398-LLD-ANA-FNC-1280

The function shall disable automatic wake-up mode (set AWUM bit in master control register to 0) i.e, set mcr of can\_x to (mcr of can\_x bitwise AND with negated value of M\_CAN\_MCR\_AWUM) when can\_awum of can\_init\_struct is DISABLE and (M\_CAN\_MSR\_INAK is equal to (msr of can\_x bitwise AND with M\_CAN\_MSR\_INAK ) (INAK bit in msr of can\_x is 1)).

##### 11.3.2.7.11 daulibstm32f4xxcan-CanInit-LLR-011

Requirement ID: H398-LLD-ANA-FNC-1281

The function shall enable no automatic retransmission (set NART bit in master control register to 1) i.e, set mcr of can\_x to (mcr of can\_x bitwise OR with M\_CAN\_MCR\_NART) when can\_nart of can\_init\_struct is ENABLE and (M\_CAN\_MSR\_INAK is equal to (msr of can\_x bitwise AND with M\_CAN\_MSR\_INAK ) (INAK bit in msr of can\_x is 1)).

##### 11.3.2.7.12 daulibstm32f4xxcan-CanInit-LLR-012

Requirement ID: H398-LLD-ANA-FNC-1282

The function shall disable no automatic retransmission (set NART bit in master control register to 0) i.e, set mcr of can\_x to (mcr of can\_x bitwise AND with negated value of M\_CAN\_MCR\_NART) when can\_nart of can\_init\_struct is DISABLE and (M\_CAN\_MSR\_INAK is equal to (msr of can\_x bitwise AND with M\_CAN\_MSR\_INAK ) (INAK bit in msr of can\_x is 1)).

##### 11.3.2.7.13 daulibstm32f4xxcan-CanInit-LLR-013

Requirement ID: H398-LLD-ANA-FNC-1283

The function shall enable receive FIFO locked mode (set RFLM bit in mter control register to 1) i.e, set mcr of can\_x to (mcr of can\_x bitwise OR with M\_CAN\_MCR\_RFLM) when can\_rflm of can\_init\_struct is ENABLE and (M\_CAN\_MSR\_INAK is equal to (msr of can\_x bitwise AND with M\_CAN\_MSR\_INAK ) (INAK bit in msr of can\_x is 1)).

##### 11.3.2.7.14 daulibstm32f4xxcan-CanInit-LLR-014

Requirement ID: H398-LLD-ANA-FNC-1284

The function shall disable receive FIFO locked mode (set RFLM bit in mter control register to 0) i.e, set mcr of can\_x to (mcr of can\_x bitwise AND with negated value of M\_CAN\_MCR\_RFLM) when can\_rflm of can\_init\_struct is DISABLE and (M\_CAN\_MSR\_INAK is equal to (msr of can\_x bitwise AND with M\_CAN\_MSR\_INAK ) (INAK bit in msr of can\_x is 1)).

##### 11.3.2.7.15 daulibstm32f4xxcan-CanInit-LLR-015

Requirement ID: H398-LLD-ANA-FNC-1285

The function shall set transmit FIFO priority (set TXFP bit in master control register to 1) i.e, set mcr of can\_x to (mcr of can\_x bitwise OR with M\_CAN\_MCR\_TXFP) when can\_txfp of can\_init\_struct is ENABLE and (M\_CAN\_MSR\_INAK is equal to (msr of can\_x bitwise AND with M\_CAN\_MSR\_INAK ) (INAK bit in msr of can\_x is 1)).

##### 11.3.2.7.16 daulibstm32f4xxcan-CanInit-LLR-016

Requirement ID: H398-LLD-ANA-FNC-1286

The function shall reset transmit FIFO priority (set TXFP bit in master control register to 0) i.e, set mcr of can\_x to (mcr of can\_x bitwise AND with negated value of M\_CAN\_MCR\_TXFP) when can\_txfp of can\_init\_struct is DISABLE and (M\_CAN\_MSR\_INAK is equal to (msr of can\_x bitwise AND with M\_CAN\_MSR\_INAK ) (INAK bit in msr of can\_x is 1)).

##### 11.3.2.7.17 daulibstm32f4xxcan-CanInit-LLR-017

Requirement ID: H398-LLD-ANA-FNC-1287

The function shall set bit timing register as below when (M\_CAN\_MSR\_INAK is equal to (msr of can\_x bitwise AND with M\_CAN\_MSR\_INAK ) (INAK bit in msr of can\_x is 1)).

Set btr of can\_x to (can\_mode of can\_init\_struct left shift by M\_SHIFT\_30 bitwise OR with

can\_sjw of can\_init\_struct left shift by M\_SHIFT\_24 bitwise OR with

can\_bs1 of can\_init\_struct left shift by M\_SHIFT\_16 bitwise OR with

can\_bs2 of can\_init\_struct left shift by M\_SHIFT\_20 bitwise OR with

(can\_prescaler of can\_init\_struct minus M\_ONE))

Note:

a) set Bit 30 with Loop back mode

b) set Bit 24 and 25 with Resynchronization jump width

c) set Bit 16-19 with Time segment 1

d) set Bit 20-22 with Time segment 2

e) set Bit 0-9 with Baud rate prescaler

##### 11.3.2.7.18 daulibstm32f4xxcan-CanInit-LLR-018

Requirement ID: H398-LLD-ANA-FNC-1288

The function shall Request leave initialization (set INRQ bit of Master control register 0) i.e, set mcr of can\_x to (mcr of can\_x bitwise AND with Nagation of M\_CAN\_MCR\_INRQ) and Set wait acknowledgement counter to M\_ZERO. when (M\_CAN\_MSR\_INAK is equal to (msr of can\_x bitwise AND with M\_CAN\_MSR\_INAK ) (INAK bit in msr of can\_x is 1)).

##### 11.3.2.7.19 daulibstm32f4xxcan-CanInit-LLR-019

Requirement ID: H398-LLD-ANA-FNC-1289

The function shall do the following when (M\_CAN\_MSR\_INAK is equal to (msr of can\_x bitwise AND with M\_CAN\_MSR\_INAK ) (INAK bit in msr of can\_x is 1)).

1. Increment the wait\_ack until (M\_CAN\_MSR\_INAK is equal to msr of can\_X bitwise AND with M\_CAN\_MSR\_INAK) AND (M\_INAK\_TIMEOUT is not equal to wait acknowledgement counter.
2. Set the initialization status to M\_CAN\_INITSTATUS\_FAILED when INAK bit in MSR of can\_x is 1 (M\_CAN\_MSR\_INAK is equal to msr of can\_X bitwise AND with M\_CAN\_MSR\_INAK) and timeout has occurred (wait acknowledgement counter is equal to M\_INAK\_TIMEOUT).
3. Set the initialization status to M\_CAN\_INITSTATUS\_SUCCESS when INAK bit in MSR of can\_x is 0 (M\_CAN\_MSR\_INAK is not equal to msr of can\_X bitwise AND with M\_CAN\_MSR\_INAK).

##### 11.3.2.7.20 daulibstm32f4xxcan-CanInit-LLR-020

Requirement ID: H398-LLD-ANA-FNC-1290

The function shall return the initialization status.

### 11.3.3 CanFilterInit

Low Level Design Details about CSU CanFilterInit will follow in the sub sections.

#### 11.3.3.1 Brief Description

The function CanFilterInit configures the CAN reception filter according to the specified parameters in the can\_filter\_init\_struct.

#### 11.3.3.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 11.3.3.3 List of global variables accessed and modified

Accessed: None

Modified: None

#### 11.3.3.4 Parameter list (Input/Output)

Inputs: T\_CAN\_FILTER\_INIT\* can\_filter\_init\_struct - pointer to a T\_CAN\_FILTER\_INIT structure that contains the configuration information.

Outputs: None

#### 11.3.3.5 Return Value

None

#### 11.3.3.6 Other CSUs called by this CSU

None

#### 11.3.3.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to CanFilterInit

##### 11.3.3.7.1 daulibstm32f4xxcan-CanFilterInit-LLR-001

Requirement ID: H398-LLD-ANA-FNC-1299

The function shall set (FINIT - bit 0) in filter master register to 1 i.e, set fmr of M\_CAN1 to (fmr of M\_CAN1 bitwise OR with M\_FMR\_FINIT).

##### 11.3.3.7.2 daulibstm32f4xxcan-CanFilterInit-LLR-002

Requirement ID: H398-LLD-ANA-FNC-1300

The function shall deactivate the filter x (where x = can\_filter\_number) in filter activation register by setting the corresponding bit to 0 i.e, falr of M\_CAN1 to (falr of M\_CAN1 bitwise AND with negated value of (M\_ONE left shift by can\_filter\_number of can\_filter\_init\_struct)).

##### 11.3.3.7.3 daulibstm32f4xxcan-CanFilterInit-LLR-003

Requirement ID: H398-LLD-ANA-FNC-1301

The function shall do the following when can\_filter\_scale of can\_filter\_init\_struct is equal to M\_CAN\_FILTERSCALE\_16BIT otherwise do nothing.

* Configure fslr of M\_CAN1 for Dual 16-bit scale configuration by setting the corresponding FSCx bit to 0 (where x= can\_filter\_number of can\_filter\_init\_struct) i.e, set fslr of M\_CAN1 to (fslr of M\_CAN1 bitwise AND with (M\_ONE left shift by can\_filter\_number of can\_filter\_init\_struct).
* set frl of sfilterregister of index (can\_filter\_number of can\_filter\_init\_struct) of M\_CAN1 to ((M\_MASK\_16 bitwise AND with (can\_filter\_mask\_id\_low of can\_filter\_init\_struct left shift by M\_SHIFT\_16)) bitwise OR with (M\_MASK\_16 bitwise AND with (can\_filter\_id\_low of can\_filter\_init\_struct))).
* set fr2 of sfilterregister of index (can\_filter\_number of can\_filter\_init\_struct) of M\_CAN1 to ((M\_MASK\_16 bitwise AND with (can\_filter\_mask\_id\_high of can\_filter\_init\_struct left shift by M\_SHIFT\_16)) bitwise OR with (M\_MASK\_16 bitwise AND with (can\_filter\_id\_high of can\_filter\_init\_struct))).

##### 11.3.3.7.4 daulibstm32f4xxcan-CanFilterInit-LLR-004

Requirement ID: H398-LLD-ANA-FNC-1302

The function shall do the following when can\_filter\_scale of can\_filter\_init\_struct is equal to M\_CAN\_FILTERSCALE\_32BIT otherwise do nothing.

* Configure fs1r of M\_CAN1 for Dual 32-bit scale configuration by setting the corresponding FSCx bit to 1 (where x=can\_filter\_init\_struct->can\_filter\_number) i.e, set fslr of M\_CAN1 to (fslr of M\_CAN1 bitwise OR with (M\_ONE left shift by can\_filter\_number of can\_filter\_init\_struct).
* set frl of sfilterregister of index (can\_filter\_number of can\_filter\_init\_struct) of M\_CAN1 to ((M\_MASK\_16 bitwise AND with (can\_filter\_id\_high of can\_filter\_init\_struct left shift by M\_SHIFT\_16)) bitwise OR with (M\_MASK\_16 bitwise AND with (can\_filter\_id\_low of can\_filter\_init\_struct))).
* set fr2 of sfilterregister of index (can\_filter\_number of can\_filter\_init\_struct) of M\_CAN1 to ((M\_MASK\_16 bitwise AND with (can\_filter\_mask\_id\_high of can\_filter\_init\_struct left shift by M\_SHIFT\_16)) bitwise OR with (M\_MASK\_16 bitwise AND with (can\_filter\_mask\_id\_low of can\_filter\_init\_struct))).

##### 11.3.3.7.5 daulibstm32f4xxcan-CanFilterInit-LLR-005

Requirement ID: H398-LLD-ANA-FNC-1303

The function shall set the FBMx bit (where x=can\_filter\_init\_struct->can\_filter\_number) in filter mode register to 0 i.e, set fmlr of M\_CAN1 to (fmlr of M\_CAN1 bitwise AND with negated value (M\_ONE left shift by can\_filter\_number of can\_filter\_init\_struct)) when can\_filter\_mode of can\_filter\_init\_struct is equal to M\_CAN\_FILTERMODE\_IDMASK.

##### 11.3.3.7.6 daulibstm32f4xxcan-CanFilterInit-LLR-006

Requirement ID: H398-LLD-ANA-FNC-1304

The function shall set the FBMx bit (where x=can\_filter\_init\_struct->can\_filter\_number) in filter mode register to 1 i.e, set fmlr of M\_CAN1 to (fmlr of M\_CAN1 bitwise OR with (M\_ONE left shift by can\_filter\_number of can\_filter\_init\_struct)) when can\_filter\_mode of can\_filter\_init\_struct is not equal to M\_CAN\_FILTERMODE\_IDMASK.

##### 11.3.3.7.7 daulibstm32f4xxcan-CanFilterInit-LLR-007

Requirement ID: H398-LLD-ANA-FNC-1305

The function shall set the FFAx bit (where x=can\_filter\_init\_struct->can\_filter\_number) in CAN filter FIFO assignment register to 0 i.e, set ffalr of M\_CAN1 to (ffalr of M\_CAN1bitwise AND with negated value of (M\_ONE left shift by can\_filter\_number of can\_filter\_init\_struct)) when can\_filter\_fifo\_assignment of can\_filter\_init\_struct is equal to M\_CAN\_FILTER\_FIFO0 otherwise do nothing.

##### 11.3.3.7.8 daulibstm32f4xxcan-CanFilterInit-LLR-008

Requirement ID: H398-LLD-ANA-FNC-1306

The function shall set the FFAx bit (where x=can\_filter\_init\_struct->can\_filter\_number) in CAN filter FIFO assignment register to 1 i.e, set ffalr of M\_CAN1 to (ffalr of M\_CAN1bitwise OR with (M\_ONE left shift by can\_filter\_number of can\_filter\_init\_struct)) when can\_filter\_fifo\_assignment of can\_filter\_init\_struct is equal to M\_CAN\_FILTER\_FIFO1 otherwise do nothing.

##### 11.3.3.7.9 daulibstm32f4xxcan-CanFilterInit-LLR-009

Requirement ID: H398-LLD-ANA-FNC-1307

The function shall set the FACTx bit (where x=can\_filter\_init\_struct->can\_filter\_number) in CAN filter activation register to 1 i.e, set falr of M\_CAN1 to (falr of M\_CAN1 bitwise OR with (M\_ONE left shift by can\_filter\_number of can\_filter\_init\_struct) when can\_filter\_activation of can\_filter\_init\_struct is ENABLE otherwise do nothing.

##### 11.3.3.7.10 daulibstm32f4xxcan-CanFilterInit-LLR-010

Requirement ID: H398-LLD-ANA-FNC-1308

The function shall set the FINIT bit to 0 (active filters mode) of filter master register i.e, set fmr of M\_CAN1 to (fmr of M\_CAN1 bitwise AND with negated value of M\_FMR\_FINIT).

### 11.3.4 CanTransmit

Low Level Design Details about CSU CanTransmit will follow in the sub sections.

#### 11.3.4.1 Brief Description

The function CanTransmit initiates and transmits a CAN frame message.

#### 11.3.4.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 11.3.4.3 List of global variables accessed and modified

Accessed: None

Modified: None

#### 11.3.4.4 Parameter list (Input/Output)

Inputs: T\_CAN\_FILTER\_INIT\* can\_x - Where x can be 1 or 2 to select the CAN peripheral.

T\_CAN\_TX\_MSG\* tx\_message - Pointer to a structure which contains CAN Id, CAN dlc and CAN data.

Outputs: T\_CAN\_FILTER\_INIT\* can\_x - Where x can be 1 or 2 to select the CAN peripheral.

T\_CAN\_TX\_MSG\* tx\_message - Pointer to a structure which contains CAN Id, CAN dlc and CAN data.

#### 11.3.4.5 Return Value

The number of the mailbox that is used for transmission to M\_CAN\_TXSTATUS\_NOMAILBOX if there is no empty mailbox.

#### 11.3.4.6 Other CSUs called by this CSU

None

#### 11.3.4.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to CanTransmit.

##### 11.3.4.7.1 daulibstm32f4xxcan-CanTransmit-LLR-001

Requirement ID: H398-LLD-ANA-FNC-1317

The function shall set transmit mailbox to 0 when TME0 bit of transmit status register is set when (M\_CAN\_TSR\_TME0 is equal to (tsr of can\_x bitwise AND with M\_CAN\_TSR\_TME0)).

##### 11.3.4.7.2 daulibstm32f4xxcan-CanTransmit-LLR-002

Requirement ID: H398-LLD-ANA-FNC-1318

The function shall set transmit mailbox to 1 when TME0 bit of transmit status register is set to 0 i.e, (M\_CAN\_TSR\_TME0 is not equal to (tsr of can\_x bitwise AND with M\_CAN\_TSR\_TME0)) AND (TME1 bit of transmit status register is set to 1 i.e, (M\_CAN\_TSR\_TME1 is equal to (tsr of can\_x bitwise AND with M\_CAN\_TSR\_TME1))).

##### 11.3.4.7.3 daulibstm32f4xxcan-CanTransmit-LLR-003

Requirement ID: H398-LLD-ANA-FNC-1319

The function shall set transmit mailbox to 2 when TME0 bit of transmit status register is set to 0 i.e, (M\_CAN\_TSR\_TME0 is not equal to (tsr of can\_x bitwise AND with M\_CAN\_TSR\_TME0)) AND (TME1 bit of transmit status register is set to 0 i.e, (M\_CAN\_TSR\_TME1 is not equal to (tsr of can\_x bitwise AND with M\_CAN\_TSR\_TME1))) AND (TME2 bit of transmit status register is set to 1 i.e, (M\_CAN\_TSR\_TME2 is equal to (tsr of can\_x bitwise AND with M\_CAN\_TSR\_TME2))).

##### 11.3.4.7.4 daulibstm32f4xxcan-CanTransmit-LLR-004

Requirement ID: H398-LLD-ANA-FNC-1320

The function shall set transmit mailbox to M\_CAN\_TXSTATUS\_NOMAILBOX when TME0 bit of transmit status register is set to 0 i.e, (M\_CAN\_TSR\_TME0 is not equal to (tsr of can\_x bitwise AND with M\_CAN\_TSR\_TME0)) AND (TME1 bit of transmit status register is set to 0 i.e, (M\_CAN\_TSR\_TME1 is not equal to (tsr of can\_x bitwise AND with M\_CAN\_TSR\_TME1))) AND (TME2 bit of transmit status register is set to 0 i.e, (M\_CAN\_TSR\_TME2 is not equal to (tsr of can\_x bitwise AND with M\_CAN\_TSR\_TME2))).

##### 11.3.4.7.5 daulibstm32f4xxcan-CanTransmit-LLR-005

Requirement ID: H398-LLD-ANA-FNC-1321

The function shall reset TXRQ bit of TX mailbox identifier register i.e, set tir of stxmailbox with index transmit mailbox of can\_x to (set tir of stxmailbox with index transmit mailbox of can\_x bitwise AND with M\_TMIDXR\_TXRQ) when any of the transmit mailbox (TME0, TME1 or TME2) is empty i.e, M\_CAN\_TXSTATUS\_NOMAILBOX is not equal to transmit mailbox otherwise no nothing.

##### 11.3.4.7.6 daulibstm32f4xxcan-CanTransmit-LLR-006

Requirement ID: H398-LLD-ANA-FNC-1322

The function shall configure TX mailbox identifier register with Standard identifier and RTR data i.e, set tir of stxmailbox with index as transmit mailbox of can\_x to (tir of stxmailbox with index as transmit mailbox of can\_x bitwise OR with ((std\_id of tx\_message left shift by M\_SHIFT\_21) bitwsie OR with (rtr of tx\_message))) when (any of the transmit mailbox (TME0, TME1 or TME2) is empty i.e, M\_CAN\_TXSTATUS\_NOMAILBOX is not equal to transmit mailbox) AND (M\_CAN\_ID\_STD is equal to ide of tx\_message).

##### 11.3.4.7.7 daulibstm32f4xxcan-CanTransmit-LLR-007

Requirement ID: H398-LLD-ANA-FNC-1323

The function shall configure TX mailbox identifier register with Extended identifier, identifier extension and RTR data i.e, set tir of stxmailbox with index as transmit mailbox of can\_x to (tir of stxmailbox with index as transmit mailbox of can\_x bitwise OR with ((ext\_id of tx\_message left shift by M\_SHIFT\_3) bitwsie OR with (ide of tx\_message) bitwise OR with (rtr of tx\_message))) when (any of the transmit mailbox (TME0, TME1 or TME2) is empty i.e, M\_CAN\_TXSTATUS\_NOMAILBOX is not equal to transmit mailbox) AND (M\_CAN\_ID\_STD is not equal to ide of tx\_message).

##### 11.3.4.7.8 daulibstm32f4xxcan-CanTransmit-LLR-008

Requirement ID: H398-LLD-ANA-FNC-1324

The function shall clear the tdtr bits from 4-31 and configure with Data length code of CAN mailbox data length control and time stamp register i.e,

1. set dlc of tx\_message to (dlc of tx\_message bitwise AND with M\_MASK\_4) when (any of the transmit mailbox (TME0, TME1 or TME2) is empty i.e, M\_CAN\_TXSTATUS\_NOMAILBOX is not equal to transmit mailbox).
2. Set tdtr of stxmailbox with index as transmit mailbox of can\_x to (tdtr of stxmailbox with index as transmit mailbox of can\_x bitwise AND with M\_MASK\_28.
3. Set tdtr of stxmailbox with index as transmit mailbox of can\_x to (tdtr of stxmailbox with index as transmit mailbox of can\_x bitwise OR with dlc of tx\_message.

when (any of the transmit mailbox (TME0, TME1 or TME2) is empty i.e, M\_CAN\_TXSTATUS\_NOMAILBOX is not equal to transmit mailbox)

##### 11.3.4.7.9 daulibstm32f4xxcan-CanTransmit-LLR-009

Requirement ID: H398-LLD-ANA-FNC-1325

The function shall configure CAN mailbox data high register with data field i.e,

1. set tdlr of stxmailbox with index as transmit mailbox of can\_x to ((data with index M\_THREE of tx\_message left shift by M\_SHIFT\_24) bitwise OR with ((data with index M\_TWO of tx\_message left shift by M\_SHIFT\_16) bitwise OR with ((data with index M\_ONE of tx\_message left shift by M\_SHIFT\_8) bitwise OR with ((data with index M\_ZERO of tx\_message)).
2. set tdhr of stxmailbox with index as transmit mailbox of can\_x to ((data with index M\_SEVEN of tx\_message left shift by M\_SHIFT\_24) bitwise OR with ((data with index M\_SIX of tx\_message left shift by M\_SHIFT\_16) bitwise OR with ((data with index M\_FIVE of tx\_message left shift by M\_SHIFT\_8) bitwise OR with ((data with index M\_FOUR of tx\_message)).

when (any of the transmit mailbox (TME0, TME1 or TME2) is empty when M\_CAN\_TXSTATUS\_NOMAILBOX is not equal to transmit mailbox)

##### 11.3.4.7.10 daulibstm32f4xxcan-CanTransmit-LLR-010

Requirement ID: H398-LLD-ANA-FNC-1326

The function shall Request for transmission (set TXRQ bit in CAN TX mailbox identifier register) i.e, set tir of stxmailbox with index transmit mailbox of can\_x to (tir of stxmailbox with index transmit mailbox of can\_x bitwise OR with M\_TMIDXR\_TXRQ) when (any of the transmit mailbox (TME0, TME1 or TME2) is empty i.e, M\_CAN\_TXSTATUS\_NOMAILBOX is not equal to transmit mailbox).

##### 11.3.4.7.11 daulibstm32f4xxcan-CanTransmit-LLR-011

Requirement ID: H398-LLD-ANA-FNC-1327

The function shall return the transmit mailbox.

### 11.3.5 CanReceive

Low Level Design Details about CSU CanReceive will follow in the sub sections.

#### 11.3.5.1 Brief Description

The function CanReceive Receives a correct CAN frame.

#### 11.3.5.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 11.3.5.3 List of global variables accessed and modified

Accessed: None

Modified: None

#### 11.3.5.4 Parameter list (Input/Output)

Inputs: T\_CAN\_TYPEDEF\* can\_x - Where x can be 1 or 2 to select the CAN peripheral.

T\_UINT8 fifo\_number: Receive FIFO number, M\_CAN\_FIFO0 or M\_CAN\_FIFO1.

Outputs: T\_CAN\_RX\_MSG\* rx\_message - Pointer to a structure receive frame which contains CAN Id, CAN DLC, CAN data and FMI number.

T\_CAN\_TYPEDEF\* can\_x - Where x can be 1 or 2 to select the CAN peripheral.

#### 11.3.5.5 Return Value

None

#### 11.3.5.6 Other CSUs called by this CSU

None

#### 11.3.5.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to CanReceive

##### 11.3.5.7.1 daulibstm32f4xxcan-CanReceive-LLR-001

Requirement ID: H398-LLD-ANA-FNC-1336

The function shall extract the IDE (identifier extension) bit value in receive FIFO mailbox identifier register when ide of rx\_message is equal to (M\_FOUR bitwise AND with rir of sfifomailbox with index fifo\_number of can\_x).

##### 11.3.5.7.2 daulibstm32f4xxcan-CanReceive-LLR-002

Requirement ID: H398-LLD-ANA-FNC-1337

The function shall extract Standard identifier value in receive FIFO mailbox identifier register i.e, set std\_id of rx\_message to (M\_MASK\_11 bitwise AND with (rir of sfifomailbox with index fifo number of can\_x right shift by M\_SHIFT\_21)) when ide of rx\_message is equal to M\_CAN\_ID\_STD (standard ID).

##### 11.3.5.7.3 daulibstm32f4xxcan-CanReceive-LLR-003

Requirement ID: H398-LLD-ANA-FNC-1338

The function shall extract extended identifier value in receive FIFO mailbox identifier register i.e, set ext\_id of rx\_message to (M\_MASK\_29 bitwsie AND with rir of sfifomailbox with index fifo number of can\_x is right shifted with M\_SHIFT\_3 when ide of rx\_message is not equal to M\_CAN\_ID\_STD (Extended ID).

##### 11.3.5.7.4 daulibstm32f4xxcan-CanReceive-LLR-004

Requirement ID: H398-LLD-ANA-FNC-1339

The function shall extract remote transmission request (frame type) value in receive FIFO mailbox identifier register i.e, set rtr of rx\_message to (M\_TWO bitwise AND with rir of sfifomailbox with index fifo number of can\_x).

##### 11.3.5.7.5 daulibstm32f4xxcan-CanReceive-LLR-005

Requirement ID: H398-LLD-ANA-FNC-1340

The function shall extract dlc value in receive FIFO mailbox identifier register i.e, set dlc of rx\_message to (M\_MASK\_4 bitwise AND with rdtr of sfifomailbox with index fifo number of can\_x).

##### 11.3.5.7.6 daulibstm32f4xxcan-CanReceive-LLR-006

Requirement ID: H398-LLD-ANA-FNC-1341

The function shall extract FMI (Filter match index) in receive FIFO mailbox identifier register i.e, set fmi of rx\_message to (M\_MASK\_8 bitwise AND with (rdtr of sfifomailbox with index fifo number of can\_x right shift by M\_SHIFT\_8)).

##### 11.3.5.7.7 daulibstm32f4xxcan-CanReceive-LLR-007

Requirement ID: H398-LLD-ANA-FNC-1342

The function shall extract data field data in receive FIFO mailbox identifier register i.e,

1. Set data of index M\_ZERO of rx\_message to (M\_MASK\_8 bitwise AND with rdlr of sfifomailbox with index fifo number of can\_x).
2. Set data of index M\_ONE of rx\_message to (M\_MASK\_8 bitwise AND with rdlr of sfifomailbox with index fifo number of can\_x right shift by M\_SHIFT\_8).
3. Set data of index M\_TWO of rx\_message to (M\_MASK\_8 bitwise AND with rdlr of sfifomailbox with index fifo number of can\_x right shift by M\_SHIFT\_16).
4. Set data of index M\_THREE of rx\_message to (M\_MASK\_8 bitwise AND with rdlr of sfifomailbox with index fifo number of can\_x right shift by M\_SHIFT\_24).
5. Set data of index M\_FOUR of rx\_message to (M\_MASK\_8 bitwise AND with rdhr of sfifomailbox with index fifo number of can\_x).
6. Set data of index M\_FIVE of rx\_message to (M\_MASK\_8 bitwise AND with rdhr of sfifomailbox with index fifo number of can\_x right shift by M\_SHIFT\_8).
7. Set data of index M\_SIX of rx\_message to (M\_MASK\_8 bitwise AND with rdhr of sfifomailbox with index fifo number of can\_x right shift by M\_SHIFT\_16).
8. Set data of index M\_SEVEN of rx\_message to (M\_MASK\_8 bitwise AND with rdhr of sfifomailbox with index fifo number of can\_x right shift by M\_SHIFT\_24).

##### 11.3.5.7.8 daulibstm32f4xxcan-CanReceive-LLR-008

Requirement ID: H398-LLD-ANA-FNC-1343

The function shall release FIFO 0 (set RFOM0 bit in receive FIFO 0 register (rf0r)) i.e, set rf0r of can\_x to (rf0r of can\_x bitwise OR with M\_CAN\_RF0R\_RFOM0) when M\_CAN\_FIFO0 is equal to fifo number.

##### 11.3.5.7.9 daulibstm32f4xxcan-CanReceive-LLR-009

Requirement ID: H398-LLD-ANA-FNC-1344

The function shall release FIFO 1 (set RFOM1 bit in receive FIFO 1 register (rf1r)) i.e, set rflr of can\_x to (rflr of can\_x bitwise OR with M\_CAN\_RF1R\_RFOM1) when M\_CAN\_FIFO0 is not equal to fifo number.

### 11.3.6 CanItConfig

Low Level Design Details about CSU CanItConfig will follow in the sub sections.

#### 11.3.6.1 Brief Description

The function CanItConfig enables or disables the specified CANx interrupts.

#### 11.3.6.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 11.3.6.3 List of global variables accessed and modified

Accessed: None

Modified: None

#### 11.3.6.4 Parameter list (Input/Output)

Inputs: T\_CAN\_TYPEDEF\* can\_x - Where x can be 1 or 2 to select the CAN peripheral.

T\_UINT32 can\_it - Specifies the CAN interrupt sources to be enabled or disabled.

T\_FUNCTIONAL\_STATE new\_state - New state of the CAN interrupts.

Outputs: T\_CAN\_TYPEDEF\* can\_x - Where x can be 1 or 2 to select the CAN peripheral.

#### 11.3.6.5 Return Value

None

#### 11.3.6.6 Other CSUs called by this CSU

None

#### 11.3.6.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to CanItConfig.

##### 11.3.6.7.1 daulibstm32f4xxcan-CanItConfig-LLR-001

Requirement ID: H398-LLD-ANA-FNC-1353

The function shall configure interrupt enable register (ier) to enable the selected can interrupt i.e, set ier of can\_x to (ier of can\_x bitwise OR with can\_it) when the new\_state is ENABLE.

##### 11.3.6.7.2 daulibstm32f4xxcan-CanItConfig-LLR-002

Requirement ID: H398-LLD-ANA-FNC-1354

The function shall configure interrupt enable register (ier) to disable the selected can interrupt i.e, set ier of can\_x to (ier of can\_x bitwise AND with negated value of can\_it when the new\_state is DISABLE.

## 11.4 daulibstm32f4xxcrc

The module provides all the CRC firmware functions.

### 11.4.1 CrcResetDr

Low Level Design Details about CSU CrcResetDr will follow in the sub sections.

#### 11.4.1.1 Brief Description

The function CrcResetDr resets the CRC control register (cr).

#### 11.4.1.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 11.4.1.3 List of global variables accessed and modified

None

#### 11.4.1.4 Parameter list (Input/Output)

Inputs: None.

Outputs: None.

#### 11.4.1.5 Return Value

None

#### 11.4.1.6 Other CSUs called by this CSU

None

#### 11.4.1.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to CrcResetDr.

##### 11.4.1.7.1 daulibstm32f4xxcrc-CrcResetDr-LLR-001

Requirement ID: H398-LLD-ANA-FNC-1364

The function shall set the RESET bit of CRC Control register (cr) to reset the CRC calculation unit i.e, set cr of M\_CRC to M\_CRC\_CR\_RESET.

### 11.4.2 CrcCalcBlockCrc

Low Level Design Details about CSU CrcCalcBlockCrc will follow in the sub sections.

#### 11.4.2.1 Brief Description

The function computes the 32-bit M\_CRC of a given buffer of data word (32-bit).

#### 11.4.2.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 11.4.2.3 List of global variables accessed and modified

None

#### 11.4.2.4 Parameter list (Input/Output)

Inputs: T\_UINT32 pbuffer: buffer containing the data to be computed.

T\_UINT32 buffer\_length: length of the buffer to be computed.

Outputs: None.

#### 11.4.2.5 Return Value

T\_UINT32 - returns 32-bit CRC.

#### 11.4.2.6 Other CSUs called by this CSU

None

#### 11.4.2.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to CrcCalcBlockCrc.

##### 11.4.2.7.1 daulibstm32f4xxcrc-CrcCalcBlockCrc-LLR-001

Requirement ID: H398-LLD-ANA-FNC-1373

The function shall copy the data from pbuffer with index as loop counter index to CRC data register (dr of M\_CRC) for loop counter index from M\_ZERO to (buffer\_length - 1) and returns the calculated crc (dr of M\_CRC).

## 11.5 daulibstm32f4xxdma

This module provides firmware functions to manage the following functionalities of the Direct Memory Access controller (DMA):

* Initialization and Configuration
* Interrupts and flags management

### 11.5.1 DmaInit

Low Level Design Details about CSU DmaInit will follow in the sub sections.

#### 11.5.1.1 Brief Description

The function DmaInit initializes the DMAy Streamx according to the specified parameters in the dma\_init\_struct structure.

#### 11.5.1.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 11.5.1.3 List of global variables accessed and modified

Accessed: None

Modified: None

#### 11.5.1.4 Parameter list (Input/Output)

Inputs: T\_DMA\_STREAM\* dmay\_streamx - Where y can be 1 or 2 to select the DMA and x can be 0 to 7 to select the DMA STREAM.

T\_DMA\_INIT\* dma\_init\_struct - Pointer to a T\_DMA\_INIT structure that contains the configuration information for the specified DMA STREAM.

Outputs: T\_DMA\_STREAM\* dmay\_streamx - Where y can be 1 or 2 to select the DMA and x can be 0 to 7 to select the DMA STREAM.

#### 11.5.1.5 Return Value

None

#### 11.5.1.6 Other CSUs called by this CSU

None

#### 11.5.1.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to DmaInit.

##### 11.5.1.7.1 daulibstm32f4xxdma-DmaInit-LLR-001

Requirement ID: H398-LLD-ANA-FNC-1383

The function shall

1. Set temporary register to cr of dmay\_streamx.
2. Set temporary register to temporary register Bitwise AND with (Negation of (M\_DMA\_SXCR\_CHSEL Bitwise OR M\_DMA\_SXCR\_MBURST Bitwise OR M\_DMA\_SXCR\_PBURST Bitwise OR M\_DMA\_SXCR\_PL Bitwise OR M\_DMA\_SXCR\_MSIZE Bitwise OR M\_DMA\_SXCR\_PSIZE Bitwise OR M\_DMA\_SXCR\_MINC Bitwise OR M\_DMA\_SXCR\_PINC Bitwise OR M\_DMA\_SXCR\_CIRC Bitwise OR M\_DMA\_SXCR\_DIR)).

##### 11.5.1.7.2 daulibstm32f4xxdma-DmaInit-LLR-002

Requirement ID: H398-LLD-ANA-FNC-1384

The function shall

1. Set temporary register to temporary register Bitwise OR with dma\_channel of dma\_init\_struct bitwise OR dma\_dir of dma\_init\_struct bitwise OR dma\_peripheral\_inc of dma\_init\_struct bitwise OR dma\_memory\_inc of dma\_init\_struct bitwise OR dma\_peripheral\_datasize of dma\_init\_struct bitwise OR dma\_memory\_datasize of dma\_init\_struct bitwise OR dma\_mode of dma\_init\_struct bitwise OR dma\_priority of dma\_init\_struct bitwise OR dma\_memory\_burst of dma\_init\_struct bitwise OR dma\_peripheral\_burst of dma\_init\_struct.
2. Set cr of dmay\_streamx to temporary register.
3. Set temporary register to fcr of dmay\_streamx.
4. Set temporary register to temporary register Bitwise AND with Negation of (M\_DMA\_SXFCR\_DMDIS Bitwise OR M\_DMA\_SXFCR\_FTH).
5. Set temporary register to temporary register Bitwise OR with dma\_fifo\_mode of dma\_init\_struct Bitwise OR with dma\_fifo\_threshold of dma\_init\_struct.
6. Set fcr of dmay\_streamx to temporary register.

##### 11.5.1.7.3 daulibstm32f4xxdma-DmaInit-LLR-003

Requirement ID: H398-LLD-ANA-FNC-1385

The function shall configure DMA stream x number of data register (ndtr), DMA stream x peripheral address register (par) and DMA stream x memory 0 address register (m0ar) with dma\_buffersize, dma\_peripheral\_baseaddr and dma\_memory0\_baseaddr of dma\_init\_struct respectively.

### 11.5.2 DmaCmd

Low Level Design Details about CSU DmaCmd will follow in the sub sections.

#### 11.5.2.1 Brief Description

The function DmaCmd enables or disables the specified DMAy Streamx.

#### 11.5.2.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 11.5.2.3 List of global variables accessed and modified

Accessed: None

Modified: None

#### 11.5.2.4 Parameter list (Input/Output)

Inputs: T\_DMA\_STREAM\* dmay\_streamx - Where y can be 1 or 2 to select the DMAand x can be 0 to 7 to select the DMA STREAM.

T\_FUNCTIONAL\_STATE new\_state - New state of the DMAy Streamx.

Outputs: T\_DMA\_STREAM\* dmay\_streamx - Where y can be 1 or 2 to select the DMAand x can be 0 to 7 to select the DMA STREAM.

#### 11.5.2.5 Return Value

None

#### 11.5.2.6 Other CSUs called by this CSU

None

#### 11.5.2.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to DmaCmd

##### 11.5.2.7.1 daulibstm32f4xxdma-DmaCmd-LLR-001

Requirement ID: H398-LLD-ANA-FNC-1403

The function shall set the selected DMAy Streamx to bitwise OR of M\_DMA\_SXCR\_EN and cr of DMAy Streamx when new\_state is ENABLE.

##### 11.5.2.7.2 daulibstm32f4xxdma-DmaCmd-LLR-002

Requirement ID: H398-LLD-ANA-FNC-1404

The function shall reset the selected DMAy Streamx to bitwise AND nagation of M\_DMA\_SXCR\_EN and cr of dmay\_streamx when new\_state is DISABLE.

### 11.5.3 DmaClearItPendingBit

Low Level Design Details about CSU DmaClearItPendingBit will follow in the sub sections.

#### 11.5.3.1 Brief Description

The function DmaClearItPendingBit clears the DMAy Streamx's interrupt pending bits.

#### 11.5.3.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 11.5.3.3 List of global variables accessed and modified

Accessed: None

Modified: None

#### 11.5.3.4 Parameter list (Input/Output)

Inputs: T\_DMA\_STREAM \* dmay\_streamx - Where y can be 1 or 2 to select the DMA and x can be 0 to 7 to select the DMA STREAM.

T\_UINT32 dma\_interrupt - Specifies the DMA interrupt pending bit to clear.

Outputs: None.

#### 11.5.3.5 Return Value

None

#### 11.5.3.6 Other CSUs called by this CSU

None

#### 11.5.3.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to DmaClearItPendingBit.

##### 11.5.3.7.1 daulibstm32f4xxdma-DmaClearItPendingBit-LLR-001

Requirement ID: H398-LLD-ANA-FNC-1413

The function shall set the dmay as DMA1 if dmay\_streamx is less than M\_DMA2\_STREAM0.

##### 11.5.3.7.2 daulibstm32f4xxdma-DmaClearItPendingBit-LLR-002

Requirement ID: H398-LLD-ANA-FNC-1414

The function shall set the dmay as DMA2 if dmay\_streamx is greater than or equal to M\_DMA2\_STREAM0.

##### 11.5.3.7.3 daulibstm32f4xxdma-DmaClearItPendingBit-LLR-003

Requirement ID: H398-LLD-ANA-FNC-1415

The function shall clear the interrupt bits in HIFCR register for the dmay when (dma\_interrupt Bitwise AND with M\_HIGH\_ISR\_MASK) is equal to SET.

##### 11.5.3.7.4 daulibstm32f4xxdma-DmaClearItPendingBit-LLR-004

Requirement ID: H398-LLD-ANA-FNC-1416

The function shall clear the interrupt bits in LIFCR register for the dmay when (dma\_interrupt Bitwise AND with M\_HIGH\_ISR\_MASK) is equal to RESET.

### 11.5.4 DmaItConfig

Low Level Design Details about CSU DmaItConfig will follow in the sub sections.

#### 11.5.4.1 Brief Description

The function DmaItConfig enables or disables the specified DMAy Streamx interrupts.

#### 11.5.4.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 11.5.4.3 List of global variables accessed and modified

Accessed: None

Modified: None

#### 11.5.4.4 Parameter list (Input/Output)

Inputs: T\_DMA\_STREAM\* dmay\_streamx - DMA Stream, where y can be 1 or 2 to select the DMA and x can be 0 to 7 to select the DMA Stream

T\_UINT32 dma\_interrupt - Specifies the DMA interrupt sources to be enabled or disabled.

T\_FUNCTIONAL\_STATE new\_state - New state of the DMAy Streamx.

Outputs: T\_DMA\_STREAM\* dmay\_streamx - DMA Stream, where y can be 1 or 2 to select the DMA and x can be 0 to 7 to select the DMA Stream

#### 11.5.4.5 Return Value

None

#### 11.5.4.6 Other CSUs called by this CSU

None

#### 11.5.4.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to DmaItConfig

##### 11.5.4.7.1 daulibstm32f4xxdma-DmaItConfig-LLR-001

Requirement ID: H398-LLD-ANA-FNC-1425

The function shall enable the selected DMA FIFO interrupts by setting the FEIE bit in DMA stream x FIFO control register (fcr) when dma\_interrupt contains FIFO interrupt(bit 7 is set) and new\_state is ENABLE.

##### 11.5.4.7.2 daulibstm32f4xxdma-DmaItConfig-LLR-002

Requirement ID: H398-LLD-ANA-FNC-1426

The function shall disable the selected DMA FIFO interrupts by resetting the FEIE bit in DMA stream x FIFO control register (fcr) when dma\_interrupt contains FIFO interrupt (bit 7 is set) and new\_state is DISABLE.

##### 11.5.4.7.3 daulibstm32f4xxdma-DmaItConfig-LLR-003

Requirement ID: H398-LLD-ANA-FNC-1427

The function shall do nothing when (dma\_interrupt Bitwise AND with M\_DMA\_IT\_FE) is equal to M\_ZERO.

##### 11.5.4.7.4 daulibstm32f4xxdma-DmaItConfig-LLR-004

Requirement ID: H398-LLD-ANA-FNC-1428

The function shall enable the selected DMA transfer interrupts by setting the TCIE, HTIE,TEIE and DMEIE bits of DMA stream x configuration register (cr) for the received dma\_interrupt when dma\_interrupt is not M\_DMA\_IT\_FE and new\_state is ENABLE.

##### 11.5.4.7.5 daulibstm32f4xxdma-DmaItConfig-LLR-005

Requirement ID: H398-LLD-ANA-FNC-1848

The function shall disable the selected DMA transfer interrupts by resetting the TCIE, HTIE, TEIE and DMEIE bits of DMA stream x configuration register(cr) for the received dma\_interrupt if dma\_interrupt is not M\_DMA\_IT\_FE and new\_state is DISABLE.

##### 11.5.4.7.6 daulibstm32f4xxdma-DmaItConfig-LLR-006

Requirement ID: H398-LLD-ANA-FNC-1849

The function shall do nothing when dma\_interrupt is equal to M\_DMA\_IT\_FE.

### 11.5.5 DmaDeInit

Low Level Design Details about CSU DmaDeInit will follow in the sub sections.

#### 11.5.5.1 Brief Description

The function DmaDeInit initializes the DMAY Streamx according to the specified parameters in the dma\_init\_struct structure.

#### 11.5.5.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 11.5.5.3 List of global variables accessed and modified

Accessed: None

Modified: None

#### 11.5.5.4 Parameter list (Input/Output)

Inputs: T\_DMA\_STREAM\* dmay\_streamx - Where y can be 1 or 2 to select the DMA and x can be 0 to 7 to select the DMA STREAM.

Outputs: T\_DMA\_STREAM\* dmay\_streamx - Where y can be 1 or 2 to select the DMA and x can be 0 to 7 to select the DMA STREAM.

#### 11.5.5.5 Return Value

None

#### 11.5.5.6 Other CSUs called by this CSU

None

#### 11.5.5.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to DmaDeInit.

##### 11.5.5.7.1 daulibstm32f4xxdma-DmaDeInit-LLR-001

Requirement ID: H398-LLD-ANA-FNC-2369

The function shall set cr of dmay\_streamx to bitwise AND of cr of dmay\_streamx and negation of M\_DMA\_SXCR\_EN.

##### 11.5.5.7.2 daulibstm32f4xxdma-DmaDeInit-LLR-002

Requirement ID: H398-LLD-ANA-FNC-2370

The function shall perform the following:

1. Update cr of dmay\_streamx to M\_ZERO
2. Set ndtr of dmay\_streamx to M\_ZERO
3. Set par of dmay\_streamx to M\_ZERO
4. Set m0ar of dmay\_streamx to M\_ZERO
5. Set m1ar of dmay\_streamx to M\_ZERO
6. Set fcr of dmay\_streamx to M\_0X00000021

##### 11.5.5.7.3 daulibstm32f4xxdma-DmaDeInit-LLR-003

Requirement ID: H398-LLD-ANA-FNC-2371

The function shall set lifcr of M\_DMA1 to M\_DMA\_STREAM0\_IT\_MASK when dmay\_streamx is equal to M\_DMA1\_STREAM0

##### 11.5.5.7.4 daulibstm32f4xxdma-DmaDeInit-LLR-004

Requirement ID: H398-LLD-ANA-FNC-2372

The function shall set lifcr of M\_DMA1 to M\_DMA\_STREAM1\_IT\_MASK when dmay\_streamx is equal to M\_DMA1\_STREAM1

##### 11.5.5.7.5 daulibstm32f4xxdma-DmaDeInit-LLR-005

Requirement ID: H398-LLD-ANA-FNC-2373

The function shall set lifcr of M\_DMA1 to M\_DMA\_STREAM2\_IT\_MASK when dmay\_streamx is equal to M\_DMA1\_STREAM2

##### 11.5.5.7.6 daulibstm32f4xxdma-DmaDeInit-LLR-006

Requirement ID: H398-LLD-ANA-FNC-2374

The function shall set lifcr of M\_DMA1 to M\_DMA\_STREAM3\_IT\_MASK when dmay\_streamx is equal to M\_DMA1\_STREAM3

##### 11.5.5.7.7 daulibstm32f4xxdma-DmaDeInit-LLR-007

Requirement ID: H398-LLD-ANA-FNC-2375

The function shall set hifcr of M\_DMA1 to M\_DMA\_STREAM4\_IT\_MASK when dmay\_streamx is equal to M\_DMA1\_STREAM4

##### 11.5.5.7.8 daulibstm32f4xxdma-DmaDeInit-LLR-008

Requirement ID: H398-LLD-ANA-FNC-2376

The function shall set hifcr of M\_DMA1 to M\_DMA\_STREAM5\_IT\_MASK when dmay\_streamx is equal to M\_DMA1\_STREAM5

##### 11.5.5.7.9 daulibstm32f4xxdma-DmaDeInit-LLR-009

Requirement ID: H398-LLD-ANA-FNC-2377

The function shall set hifcr of M\_DMA1 to M\_DMA\_STREAM6\_IT\_MASK when dmay\_streamx is equal to M\_DMA1\_STREAM6

##### 11.5.5.7.10 daulibstm32f4xxdma-DmaDeInit-LLR-010

Requirement ID: H398-LLD-ANA-FNC-2378

The function shall set hifcr of M\_DMA1 to M\_DMA\_STREAM7\_IT\_MASK when dmay\_streamx is equal to M\_DMA1\_STREAM7

##### 11.5.5.7.11 daulibstm32f4xxdma-DmaDeInit-LLR-011

Requirement ID: H398-LLD-ANA-FNC-2379

The function shall set lifcr of M\_DMA2 to M\_DMA\_STREAM0\_IT\_MASK when dmay\_streamx is equal to M\_DMA2\_STREAM0

##### 11.5.5.7.12 daulibstm32f4xxdma-DmaDeInit-LLR-012

Requirement ID: H398-LLD-ANA-FNC-2380

The function shall set lifcr of M\_DMA2 to M\_DMA\_STREAM1\_IT\_MASK when dmay\_streamx is equal to M\_DMA2\_STREAM1

##### 11.5.5.7.13 daulibstm32f4xxdma-DmaDeInit-LLR-013

Requirement ID: H398-LLD-ANA-FNC-2381

The function shall set lifcr of M\_DMA2 to M\_DMA\_STREAM2\_IT\_MASK when dmay\_streamx is equal to M\_DMA2\_STREAM2

##### 11.5.5.7.14 daulibstm32f4xxdma-DmaDeInit-LLR-014

Requirement ID: H398-LLD-ANA-FNC-2382

The function shall set lifcr of M\_DMA2 to M\_DMA\_STREAM3\_IT\_MASK when dmay\_streamx is equal to M\_DMA2\_STREAM3

##### 11.5.5.7.15 daulibstm32f4xxdma-DmaDeInit-LLR-015

Requirement ID: H398-LLD-ANA-FNC-2383

The function shall set hifcr of M\_DMA2 to M\_DMA\_STREAM4\_IT\_MASK when dmay\_streamx is equal to M\_DMA2\_STREAM4

##### 11.5.5.7.16 daulibstm32f4xxdma-DmaDeInit-LLR-016

Requirement ID: H398-LLD-ANA-FNC-2384

The function shall set hifcr of M\_DMA2 to M\_DMA\_STREAM5\_IT\_MASK when dmay\_streamx is equal to M\_DMA2\_STREAM5

##### 11.5.5.7.17 daulibstm32f4xxdma-DmaDeInit-LLR-017

Requirement ID: H398-LLD-ANA-FNC-2385

The function shall set hifcr of M\_DMA2 to M\_DMA\_STREAM6\_IT\_MASK when dmay\_streamx is equal to M\_DMA2\_STREAM6

##### 11.5.5.7.18 daulibstm32f4xxdma-DmaDeInit-LLR-018

Requirement ID: H398-LLD-ANA-FNC-2386

The function shall set hifcr of M\_DMA2 to M\_DMA\_STREAM7\_IT\_MASK when dmay\_streamx is equal to M\_DMA2\_STREAM7 otherwise do nothing.

## 11.6 daulibstm32f4xxflash

This module provides firmware functions to manage the following functionalities of the FLASH peripheral:

* FLASH Interface configuration
* Set the latency
* Enable/Disable the prefetch buffer
* Enable/Disable the Instruction cache and the Data cache
* Interrupts and flags management

### 11.6.1 FlashSetLatency

Low Level Design Details about CSU FlashSetLatency will follow in the sub sections.

#### 11.6.1.1 Brief Description

The function FlashSetLatency sets the code latency value.

#### 11.6.1.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 11.6.1.3 List of global variables accessed and modified

Accessed: None

Modified: None

#### 11.6.1.4 Parameter list (Input/Output)

Inputs: T\_UINT32 flash\_latency - specifies the FLASH Latency value.

Outputs: None

#### 11.6.1.5 Return Value

None

#### 11.6.1.6 Other CSUs called by this CSU

None

#### 11.6.1.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to FlashSetLatency.

##### 11.6.1.7.1 daulibstm32f4xxflash-FlashSetLatency-LLR-001

Requirement ID: H398-LLD-ANA-FNC-1438

The function shall set the code latency value in acr (Flash access control register) i.e, Set content of the address M\_ACR\_BYTE0\_ADDRESS to flash\_latency.

### 11.6.2 FlashPrefetchBufferCmd

Low Level Design Details about CSU FlashPrefetchBufferCmd will follow in the sub sections.

#### 11.6.2.1 Brief Description

The function FlashPrefetchBufferCmd enables or disables the Prefetch Buffer.

#### 11.6.2.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 11.6.2.3 List of global variables accessed and modified

Accessed: None

Modified: None

#### 11.6.2.4 Parameter list (Input/Output)

Inputs: T\_FUNCTIONAL\_STATE new\_state - New state of the Prefetch Buffer.

Outputs: None

#### 11.6.2.5 Return Value

None

#### 11.6.2.6 Other CSUs called by this CSU

None

#### 11.6.2.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to FlashPrefetchBufferCmd.

##### 11.6.2.7.1 daulibstm32f4xxflash-FlashPrefetchBufferCmd-LLR-001

Requirement ID: H398-LLD-ANA-FNC-1447

The function shall enable the prefetch buffer (set PRFTEN – bit 8) in acr (Flash access control register) when new\_state is ENABLE i.e, set acr of M\_FLASH with (acr of M\_FLASH bitwise OR with M\_FLASH\_ACR\_PRFTEN).

##### 11.6.2.7.2 daulibstm32f4xxflash-FlashPrefetchBufferCmd-LLR-002

Requirement ID: H398-LLD-ANA-FNC-1448

The function shall disable the prefetch buffer (reset PRFTEN – bit 8) in acr (Flash access control register) when new\_state is DISABLE i.e, set acr of M\_FLASH with (acr of M\_FLASH bitwise AND with negated value of M\_FLASH\_ACR\_PRFTEN).

### 11.6.3 FlashInstructionCacheCmd

Low Level Design Details about CSU FlashInstructionCacheCmd will follow in the sub sections.

#### 11.6.3.1 Brief Description

The function FlashInstructionCacheCmd enables or disables the Instruction Cache feature.

#### 11.6.3.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 11.6.3.3 List of global variables accessed and modified

Accessed: None

Modified: None

#### 11.6.3.4 Parameter list (Input/Output)

Inputs: T\_FUNCTIONAL\_STATE new\_state - New state of the Instruction Cache.

Outputs: None

#### 11.6.3.5 Return Value

None

#### 11.6.3.6 Other CSUs called by this CSU

None

#### 11.6.3.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to FlashInstructionCacheCmd.

##### 11.6.3.7.1 daulibstm32f4xxflash-FlashInstructionCacheCmd-LLR-001

Requirement ID: H398-LLD-ANA-FNC-1457

The function shall enable the Instruction Cache (set ICEN – bit 9) in acr (Flash access control register) when new\_state is ENABLE i.e, set acr of M\_FLASH with (acr of M\_FLASH bitwise OR with M\_FLASH\_ACR\_ICEN).

##### 11.6.3.7.2 daulibstm32f4xxflash-FlashInstructionCacheCmd-LLR-002

Requirement ID: H398-LLD-ANA-FNC-1458

The function shall disable the Instruction Cache (reset ICEN – bit 9) in acr (Flash access control register) when new\_state is DISABLE i.e, set acr of M\_FLASH with (acr of M\_FLASH bitwise AND with negated value of M\_FLASH\_ACR\_ICEN).

### 11.6.4 FlashDataCacheCmd

Low Level Design Details about CSU FlashDataCacheCmd will follow in the sub sections.

#### 11.6.4.1 Brief Description

The function FlashDataCacheCmd enables or disables the Data Cache feature.

#### 11.6.4.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 11.6.4.3 List of global variables accessed and modified

Accessed: None

Modified: None

#### 11.6.4.4 Parameter list (Input/Output)

Inputs: T\_FUNCTIONAL\_STATE new\_state - New state of the data Cache.

Outputs: None

#### 11.6.4.5 Return Value

None

#### 11.6.4.6 Other CSUs called by this CSU

None

#### 11.6.4.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to FlashDataCacheCmd.

##### 11.6.4.7.1 daulibstm32f4xxflash-FlashDataCacheCmd-LLR-001

Requirement ID: H398-LLD-ANA-FNC-1467

The function shall enable the Data Cache (set DCEN - bit 10) in acr (Flash access control register) when new\_state is ENABLE i.e, set acr of M\_FLASH with (acr of M\_FLASH bitwise OR with M\_FLASH\_ACR\_DCEN).

##### 11.6.4.7.2 daulibstm32f4xxflash-FlashDataCacheCmd-LLR-002

Requirement ID: H398-LLD-ANA-FNC-1468

The function shall disable the Data Cache (reset DCEN - bit 10) of acr (Flash access control register) when new\_state is DISABLE i.e, set acr of M\_FLASH with (acr of M\_FLASH bitwise AND with negated value of M\_FLASH\_ACR\_DCEN).

## 11.7 daulibstm32f4xxfsmc

This module provides firmware functions to manage the following functionality of the FSMC peripheral:

* Interface with SRAM and NOR memories.

### 11.7.1 FsmcNorSramInit

Low Level Design Details about CSU FsmcNorSramInit will follow in the sub sections.

#### 11.7.1.1 Brief Description

The function FsmcNorSramInit initializes the FSMC NOR/SRAM Banks according to the specified parameters in the FSMC NOR/SRAM initialization structure.

#### 11.7.1.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 11.7.1.3 List of global variables accessed and modified

Accessed: None

Modified: None

#### 11.7.1.4 Parameter list (Input/Output)

Inputs: T\_FSMC\_NORSRAM\_INIT\* fsmc\_norsraminit\_struct - Pointer to a T\_FSMC\_NORSRAM\_INIT structure that contains the configuration information for the FSMC NOR/SRAM specified Banks.

Outputs: None

#### 11.7.1.5 Return Value

None

#### 11.7.1.6 Other CSUs called by this CSU

None

#### 11.7.1.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to FsmcNorSramInit.

##### 11.7.1.7.1 daulibstm32f4xxfsmc-FsmcNorSramInit-LLR-001

Requirement ID: H398-LLD-ANA-FNC-1478

The function shall configure the FSMC BANK1 NOR/SRAM FSMC Bank control register with the received configuration information (Address/data multiplexing enable bit, Memory Type, Memory databus width, Burst enable bit, Wait signal during asynchronous transfers, Wait signal polarity bit, Wrapped burst mode, Wait timing configuration, Write enable bit, Wait enable bit, Extended mode enable, Write burst enable) i.e,

Set btcr of M\_FSMC\_BANK1 with index (fsmc\_bank of fsmc\_norsraminit\_struct) to (fsmc\_data\_address\_mux of fsmc\_norsraminit\_struct bitwise OR with

fsmc\_memory\_type of fsmc\_norsraminit\_struct bitwise OR with

fsmc\_memory\_datawidth of fsmc\_norsraminit\_struct bitwise OR with

fsmc\_burst\_accessmode of fsmc\_norsraminit\_struct bitwise OR with

fsmc\_asynchronous\_wait of fsmc\_norsraminit\_struct bitwise OR with

fsmc\_waitsignal\_polarity of fsmc\_norsraminit\_struct bitwise OR with

fsmc\_wrap\_mode of fsmc\_norsraminit\_struct bitwise OR with

fsmc\_waitsignal\_active of fsmc\_norsraminit\_struct bitwise OR with

fsmc\_write\_operation of fsmc\_norsraminit\_struct bitwise OR with

fsmc\_waitsignal of fsmc\_norsraminit\_struct bitwise OR with

fsmc\_extended\_mode of fsmc\_norsraminit\_struct bitwise OR with

fsmc\_write\_burst of fsmc\_norsraminit\_struct).

##### 11.7.1.7.2 daulibstm32f4xxfsmc-FsmcNorSramInit-LLR-002

Requirement ID: H398-LLD-ANA-FNC-1479

The function shall enable NOR Flash memory access operations in BCR1 register when FACCEN - bit 6 in BCR1 is set i.e,

Set btcr with index (fsmc\_bank of fsmc\_norsraminit\_struct) of M\_FSMC\_BANK1 to (btcr of M\_FSMC\_BANK1 with index (fsmc\_bank of fsmc\_norsraminit\_struct) bitwise OR with M\_BCR\_FACCEN\_SET) when fsmc\_memory\_type of fsmc\_norsraminit\_struct is equal to M\_FSMC\_MEMORYTYPE\_NOR otherwise do nothing.

##### 11.7.1.7.3 daulibstm32f4xxfsmc-FsmcNorSramInit-LLR-003

Requirement ID: H398-LLD-ANA-FNC-1480

The function shall configure the FSMC BANK1 NOR/SRAM Bank Timing register with the received configuration information (Address setup phase duration, Address-hold phase duration, Memory databus width, Data-phase duration, Bus turnaround phase duration, Clock divide ratio, Data latency, Access mode) i.e,

Set btcr with index (fsmc\_bank of fsmc\_norsraminit\_struct added with M\_ONE) of M\_FSMC\_BANK1 to (fsmc\_address\_setuptime of fsmc\_readwrite\_timing\_struct of fsmc\_norsraminit\_struct bitwise OR with (fsmc\_address\_holdtime of fsmc\_readwrite\_timing\_struct of fsmc\_norsraminit\_struct left shift by M\_FOUR) bitwise OR with

(fsmc\_data\_setuptime of fsmc\_readwrite\_timing\_struct of fsmc\_norsraminit\_struct left shift by M\_EIGHT) bitwise OR with

(fsmc\_bus\_turnaround\_duration of fsmc\_readwrite\_timing\_struct of fsmc\_norsraminit\_struct left shift by M\_SIXTEEN) bitwise OR with

(fsmc\_clk\_division of fsmc\_readwrite\_timing\_struct of fsmc\_norsraminit\_struct left shift by M\_TWENTY) bitwise OR with

(fsmc\_data\_latency of fsmc\_readwrite\_timing\_struct of fsmc\_norsraminit\_struct left shift by M\_TWENTYFOUR) bitwise OR with

(fsmc\_access\_mode of fsmc\_readwrite\_timing\_struct of fsmc\_norsraminit\_struct)).

##### 11.7.1.7.4 daulibstm32f4xxfsmc-FsmcNorSramInit-LLR-004

Requirement ID: H398-LLD-ANA-FNC-1481

The function shall configure Bank1 NOR/SRAM Flash write timing register for write configuration with the received Timing parameters (fsmc address setuptime, fsmc holdtime, fsmc data setuptime, fsmc clock division, fsmc data latency and fsmc access mode) when extended mode is enabled i.e,

Set bwtr with index (fsmc\_bank of fsmc\_norsraminit\_struct) of M\_FSMC\_BANK1E to

((fsmc\_address\_setuptime of fsmc\_write\_timing\_struct of fsmc\_norsraminit\_struct) bitwise OR with (fsmc\_address\_holdtime of fsmc\_write\_timing\_struct of fsmc\_norsraminit\_struct left shift by M\_FOUR) bitwise OR with

(fsmc\_data\_setuptime of fsmc\_write\_timing\_struct of fsmc\_norsraminit\_struct left shift by M\_EIGHT) bitwise OR with

(fsmc\_clk\_division of fsmc\_write\_timing\_struct of fsmc\_norsraminit\_struct left shift by M\_TWENTY) bitwise OR with

(fsmc\_data\_latency of fsmc\_write\_timing\_struct of fsmc\_norsraminit\_struct left shift by M\_TWENTYFOUR) bitwise OR with

(fsmc\_access\_mode of fsmc\_write\_timing\_struct of fsmc\_norsraminit\_struct)) when fsmc\_extended\_mode of fsmc\_norsraminit\_struct is equal to M\_FSMC\_EXTENDEDMODE\_ENABLE.

##### 11.7.1.7.5 daulibstm32f4xxfsmc-FsmcNorSramInit-LLR-005

Requirement ID: H398-LLD-ANA-FNC-1482

The function shall reset Bank1 NOR/SRAM Flash write timing register when extended mode is not used i.e, Set bwtr with index (fsmc\_bank of fsmc\_norsraminit\_struct) of M\_FSMC\_BANK1E to M\_0X0FFFFFFF when fsmc\_extended\_mode of fsmc\_norsraminit\_struct is not equal to M\_FSMC\_EXTENDEDMODE\_ENABLE.

### 11.7.2 FsmcNorSramCmd

Low Level Design Details about CSU FsmcNorSramCmd will follow in the sub sections.

#### 11.7.2.1 Brief Description

The function FsmcNorSramCmd enables or disables the specified NOR/SRAM Memory Bank.

#### 11.7.2.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 11.7.2.3 List of global variables accessed and modified

Accessed: None

Modified: None

#### 11.7.2.4 Parameter list (Input/Output)

Inputs: T\_UINT32 fsmc\_bank - Specifies the FSMC Bank to be used

T\_FUNCTIONAL\_STATE new\_state - New state of the fsmc\_bank.

Outputs: None

#### 11.7.2.5 Return Value

None

#### 11.7.2.6 Other CSUs called by this CSU

None

#### 11.7.2.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to FsmcNorSramCmd.

##### 11.7.2.7.1 daulibstm32f4xxfsmc-FsmcNorSramCmd-LLR-001

Requirement ID: H398-LLD-ANA-FNC-1491

The function shall enable selected NOR/SRAM Memory Bank (MBKEN - bit 0) in the Bank Control Register when new state of FSMC bank is ENABLE i.e,Set btcr of M\_FSMC\_BANK1 with index fsmc\_bank to (btcr of M\_FSMC\_BANK1 with index fsmc\_bank bitwise OR with M\_BCR\_MBKEN\_SET).

##### 11.7.2.7.2 daulibstm32f4xxfsmc-FsmcNorSramCmd-LLR-002

Requirement ID: H398-LLD-ANA-FNC-1492

The function shall disable selected NOR/SRAM Memory Bank (MBKEN - bit 0) in the Bank Control Register when new state of FSMC bank is DISABLE i.e,Set btcr of M\_FSMC\_BANK1 with index fsmc\_bank to (btcr of M\_FSMC\_BANK1 with index fsmc\_bank bitwise AND with M\_BCR\_MBKEN\_RESET).

## 11.8 daulibstm32f4xxgpio

This module provides the implementation of firmware functions to manage the following functionalities of the GPIO peripheral:

* Initialization and Configuration
* GPIO Read and Write
* GPIO Alternate functions configuration

### 11.8.1 GpioInit

Low Level Design Details about CSU GpioInit will follow in the sub sections.

#### 11.8.1.1 Brief Description

The function GpioInit initializes the gpio\_x peripheral according to the specified parameters in the GPIO initialization structure.

#### 11.8.1.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 11.8.1.3 List of global variables accessed and modified

Accessed: None

Modified: None

#### 11.8.1.4 Parameter list (Input/Output)

Inputs: T\_GPIO\* gpio\_x - GPIO peripheral, where x can be (A...I) to select the GPIO peripheral.

T\_GPIO\_INIT\* gpio\_init\_struct - Pointer to a T\_GPIO\_INIT structure that contains the configuration information for the specified GPIO peripheral.

Outputs: T\_GPIO\* gpio\_x - GPIO peripheral, where x can be (A...I) to select the GPIO peripheral.

#### 11.8.1.5 Return Value

None

#### 11.8.1.6 Other CSUs called by this CSU

None

#### 11.8.1.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to GpioInit.

##### 11.8.1.7.1 daulibstm32f4xxgpio-GpioInit-LLR-001

Requirement ID: H398-LLD-ANA-FNC-1502

The function shall loop for all gpio pin position (M\_ZERO to M\_GPIO\_PERIPHERALS) and does the following:

1. Set position variable to M\_ONE left shift by gpio pin position
2. Calculate the current pin position i.e, set current pin position to (gpio\_pin of gpio\_init\_struct bitwise AND with position variable).
3. when current pin position is equal to position variable, perform as follows otherwise do nothing.

* Clear the GPIO\_MODER register for the current pin position i.e, set moder of gpio\_x to (moder of gpio\_x bitwise AND with negated value of (M\_GPIO\_MODER\_MODER0 left shift by (gpio pin position multiplied with M\_TWO))).
* Set the GPIO\_MODER register with the received gpio mode i.e, set moder of gpio\_x to (moder of gpio\_x bitwise OR with (gpio\_mode of gpio\_init\_struct left shift by (gpio pin position multiplied by M\_TWO))).
* When the received gpio mode is output or alternate function(GPIO\_MODE\_OUT is equal to gpio\_mode of gpio\_init\_struct OR GPIO\_MODE\_AF is equal to gpio\_mode of gpio\_init\_struct), perform as follows otherwise do nothing.
* Clear the GPIO\_OSPEEDR register for the current pin position i.e, set ospeedr of gpio\_x to (ospeedr of gpio\_x bitwise AND with negated value of (M\_GPIO\_OSPEEDER\_OSPEEDR0 left shift by (gpio pin position multiplied by M\_TWO))).
* Set the GPIO\_OSPEEDR register with the received gpio speed i.e, set ospeedr of gpio\_x to (ospeedr of gpio\_x bitwise OR with (gpio\_speed of gpio\_init\_struct left shift by (gpio pin position multiplied by M\_TWO))).
* Clear the GPIO\_OTYPER register for the current pin position i.e, set otyper of gpio\_x to (otyper of gpio\_x bitwise AND with negated value of (M\_GPIO\_OTYPER\_OT\_0 left shift by gpio pin position)).
* Set the GPIO\_OTYPER register with the received gpio output type i.e, set otyper of gpio\_x to (otyper of gpio\_x bitwise OR with (gpio\_otype of gpio\_init\_struct left shift by gpio pin position)).
* Clear the GPIO\_PUPDR register for the current pin position i.e, set pupdr of gpio\_x to (pupdr of gpio\_x bitwise AND with negated value of (M\_GPIO\_PUPDR\_PUPDR0 left shift by (gpio pin position multiplied by M\_TWO))).
* Set the GPIOx\_PUPDR register with Pull-up Pull down resistor configuration, i.e, set pupdr of gpio\_x to (pupdr of gpio\_x bitwise OR with gpio\_pupd of gpio\_init\_struct left shift by (gpio pin position multiplied by M\_TWO)).

### 11.8.2 GpioSetBits

Low Level Design Details about CSU GpioSetBits will follow in the sub sections.

#### 11.8.2.1 Brief Description

The function GpioSetBits sets the selected data port bits.

#### 11.8.2.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 11.8.2.3 List of global variables accessed and modified

Accessed: None

Modified: None

#### 11.8.2.4 Parameter list (Input/Output)

Inputs: T\_UINT16 gpio\_pin - Specifies the port bits to be set.

Outputs: T\_GPIO\* gpio\_x - GPIO peripheral, where x can be (A...I) to select the GPIO peripheral.

#### 11.8.2.5 Return Value

None

#### 11.8.2.6 Other CSUs called by this CSU

None

#### 11.8.2.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to GpioSetBits.

##### 11.8.2.7.1 daulibstm32f4xxgpio-GpioSetBits-LLR-001

Requirement ID: H398-LLD-ANA-FNC-1520

The function shall set the GPIO port bit set/reset low register i.e, set bsrrl of gpio\_x to gpio\_pin.

### 11.8.3 GpioResetBits

Low Level Design Details about CSU GpioResetBits will follow in the sub sections.

#### 11.8.3.1 Brief Description

The function GpioResetBits clears the selected data port bits.

#### 11.8.3.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 11.8.3.3 List of global variables accessed and modified

Accessed: None

Modified: None

#### 11.8.3.4 Parameter list (Input/Output)

Inputs: T\_UINT16 gpio\_pin - specifies the port bits to be reset.

Outputs: T\_GPIO\* gpio\_x - GPIO peripheral, where x can be (A...I) to select the GPIO peripheral.

#### 11.8.3.5 Return Value

None

#### 11.8.3.6 Other CSUs called by this CSU

None

#### 11.8.3.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to GpioResetBits.

##### 11.8.3.7.1 daulibstm32f4xxgpio-GpioResetBits-LLR-001

Requirement ID: H398-LLD-ANA-FNC-1529

The function shall set the GPIO port bit set/reset high register i.e, bsrrh of gpio\_x to gpio\_pin.

### 11.8.4 GpioToggleBits

Low Level Design Details about CSU GpioToggleBits will follow in the sub sections.

#### 11.8.4.1 Brief Description

The function GpioToggleBits toggles the specified GPIO pins.

#### 11.8.4.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 11.8.4.3 List of global variables accessed and modified

Accessed: None

Modified: None

#### 11.8.4.4 Parameter list (Input/Output)

Inputs: T\_GPIO\* gpio\_x - GPIO peripheral, where x can be (A...I) to select the GPIO peripheral.

T\_UINT16 gpio\_pin - Specifies the pins to be toggled.

Outputs: T\_GPIO\* gpio\_x - GPIO peripheral, where x can be (A...I) to select the GPIO peripheral.

#### 11.8.4.5 Return Value

None

#### 11.8.4.6 Other CSUs called by this CSU

None

#### 11.8.4.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to GpioToggleBits.

##### 11.8.4.7.1 daulibstm32f4xxgpio-GpioToggleBits-LLR-001

Requirement ID: H398-LLD-ANA-FNC-1538

The function shall toggle the received gpio pin gpio\_pin in Gpio output data register i.e, set odr of gpio\_x to (odr of gpio\_x bitwise XOR with gpio\_pin).

### 11.8.5 GpioPinAFConfig

Low Level Design Details about CSU GpioPinAFConfig will follow in the sub sections.

#### 11.8.5.1 Brief Description

The function GpioPinAFConfig changes the mapping of the specified pin.

#### 11.8.5.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 11.8.5.3 List of global variables accessed and modified

Accessed: None

Modified: None

#### 11.8.5.4 Parameter list (Input/Output)

Inputs: T\_GPIO\* gpio\_x - GPIO peripheral, where x can be (A...I) to select the GPIO peripheral.

T\_UINT16 gpio\_pinsource - specifies the pin for the Alternate function.

T\_UINT8 gpio\_af - selects the pin to be used as Alternate function.

Outputs: T\_GPIO\* gpio\_x - GPIO peripheral, where x can be (A...I) to select the GPIO peripheral.

#### 11.8.5.5 Return Value

None

#### 11.8.5.6 Other CSUs called by this CSU

None

#### 11.8.5.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to GpioPinAFConfig.

##### 11.8.5.7.1 daulibstm32f4xxgpio-GpioPinAFConfig-LLR-001

Requirement ID: H398-LLD-ANA-FNC-1547

The function shall change the mapping of specified pin gpio\_pinsource to the alternate function by configuring the afr (Alternate function register) i.e,

1. set Temp afr register to (gpio\_af left shift by (gpio\_pinsource bitwise AND with (M\_SEVEN multiplied with M\_FOUR))).
2. Set afr of gpio\_x with index (gpio\_ pinsource right shift by M\_THREE) to (afr of gpio\_x with index (gpio\_ pinsource right shift by M\_THREE) bitwise AND with negated value of (M\_FIFTEEN left shift by (gpio\_ pinsource bitwise AND with (M\_SEVEN multiplied with M\_FOUR)))).
3. set Temp afr register 2 to (afr of gpio\_x with index (gpio\_ pinsource right shift by M\_THREE) bitwise OR with Temp afr register.
4. Set afr of gpio\_x with index (gpio\_ pinsource right shift by M\_THREE) to Temp afr register 2.

## 11.9 daulibstm32f4xxiwdg

This module provides firmware functions to manage the following functionality of the Independent watchdog (IWDG) peripheral:

* Prescaler and Counter configuration
* IWDG activation

### 11.9.1 IwdgWriteAccessCmd

Low Level Design Details about CSU IwdgWriteAccessCmd will follow in the sub sections.

#### 11.9.1.1 Brief Description

The function IwdgWriteAccessCmd enables or disables write access to IWDG\_PR (Prescaler register) and IWDG\_RLR (Reload register).

#### 11.9.1.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 11.9.1.3 List of global variables accessed and modified

Accessed: None

Modified: None

#### 11.9.1.4 Parameter list (Input/Output)

Inputs: T\_UINT16 iwdg\_write\_access – New state of write access to IWDG\_PR and IWDG\_RLR registers.

Outputs: None

#### 11.9.1.5 Return Value

None

#### 11.9.1.6 Other CSUs called by this CSU

None

#### 11.9.1.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to IwdgWriteAccessCmd.

##### 11.9.1.7.1 daulibstm32f4xxiwdg-IwdgWriteAccessCmd-LLR-001

Requirement ID: H398-LLD-ANA-FNC-1557

The function shall configure IWDG Key register with iwdg\_write\_access to enable or disable access to the IWDG\_PR and IWDG\_RLR registers i.e, set kr of M\_IWDG to iwdg\_write\_access.

Note:

1. Write access to the IWDG\_PR and IWDG\_RLR registers is protected. To modify these registers, 0x5555 (IWDG\_PR), 0xAAAA (IWDG\_RLR) has to be written in the IWDG\_KR register.
2. Writing the key value CCCCh starts the watchdog.

### 11.9.2 IwdgSetPrescaler

Low Level Design Details about CSU IwdgSetPrescaler will follow in the sub sections.

#### 11.9.2.1 Brief Description

The function IwdgSetPrescaler sets IWDG prescaler value in Prescaler register.

#### 11.9.2.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 11.9.2.3 List of global variables accessed and modified

Accessed: None

Modified: None

#### 11.9.2.4 Parameter list (Input/Output)

Inputs: T\_UINT8 iwdg\_prescaler - Specifies the IWDG Prescaler value.

Outputs: None

#### 11.9.2.5 Return Value

None

#### 11.9.2.6 Other CSUs called by this CSU

None

#### 11.9.2.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to IwdgSetPrescaler.

##### 11.9.2.7.1 daulibstm32f4xxiwdg-IwdgSetPrescaler-LLR-001

Requirement ID: H398-LLD-ANA-FNC-1566

The function shall configure IWDG Prescaler register (pr) with prescaler value iwdg\_prescaler i.e, set pr of M\_IWDG to iwdg\_prescaler.

### 11.9.3 IwdgSetReload

Low Level Design Details about CSU IwdgSetReloadwill follow in the sub sections.

#### 11.9.3.1 Brief Description

The function IwdgSetReload sets IWDG reload value in Reload register.

#### 11.9.3.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 11.9.3.3 List of global variables accessed and modified

Accessed: None

Modified: None

#### 11.9.3.4 Parameter list (Input/Output)

Inputs: T\_UINT16 reload - specifies the IWDG Reload value.

Outputs: None

#### 11.9.3.5 Return Value

None

#### 11.9.3.6 Other CSUs called by this CSU

None

#### 11.9.3.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to IwdgSetReload.

##### 11.9.3.7.1 daulibstm32f4xxiwdg-IwdgSetReload-LLR-001

Requirement ID: H398-LLD-ANA-FNC-1575

The function shall configure IWDG Reload register (rlr) with reload value i.e, set rlr of M\_IWDG to reload.

### 11.9.4 IwdgReloadCounter

Low Level Design Details about CSU IwdgReloadCounter will follow in the sub sections.

#### 11.9.4.1 Brief Description

The function IwdgReloadCounter reloads IWDG counter with value defined in the reload register.

#### 11.9.4.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 11.9.4.3 List of global variables accessed and modified

Accessed: None

Modified: None

#### 11.9.4.4 Parameter list (Input/Output)

Inputs: None

Outputs: None

#### 11.9.4.5 Return Value

None

#### 11.9.4.6 Other CSUs called by this CSU

None

#### 11.9.4.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to IwdgReloadCounter.

##### 11.9.4.7.1 daulibstm32f4xxiwdg-IwdgReloadCounter-LLR-001

Requirement ID: H398-LLD-ANA-FNC-1584

The function shall reload the watchdog counter with the reload value in reload register (IWDG\_RLR) i.e, set kr of M\_IWDG to M\_KR\_KEY\_RELOAD.

### 11.9.5 IwdgEnable

Low Level Design Details about CSU IwdgEnable will follow in the sub sections.

#### 11.9.5.1 Brief Description

The function IwdgEnable enables IWDG (write access to IWDG\_PR and IWDG\_RLR registers disabled).

#### 11.9.5.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 11.9.5.3 List of global variables accessed and modified

Accessed: None

Modified: None

#### 11.9.5.4 Parameter list (Input/Output)

Inputs: None

Outputs: None

#### 11.9.5.5 Return Value

None

#### 11.9.5.6 Other CSUs called by this CSU

None

#### 11.9.5.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to IwdgEnable

##### 11.9.5.7.1 daulibstm32f4xxiwdg-IwdgEnable-LLR-001

Requirement ID: H398-LLD-ANA-FNC-1593

The function shall configure IWDG Key register with the value M\_KR\_KEY\_ENABLE to start the watchdog i.e, set kr of M\_IWDG to M\_KR\_KEY\_ENABLE.

## 11.10 daulibstm32f4xxpwr

This module provides firmware functions to manage the following functionality of the Power Controller (PWR) peripheral:

* Main Regulator configuration

### 11.10.1 PwrMainRegulatorModeConfig

Low Level Design Details about CSU PwrMainRegulatorModeConfig will follow in the sub sections.

#### 11.10.1.1 Brief Description

The function PwrMainRegulatorModeConfig configures the main internal regulator output voltage.

#### 11.10.1.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 11.10.1.3 List of global variables accessed and modified

Accessed: None

Modified: None

#### 11.10.1.4 Parameter list (Input/Output)

Inputs: T\_UINT32 pwr\_regulator\_voltage - Specifies the regulator output voltage to achieve a tradeoff between performance and power consumption when the device does not operate at the maximum frequency.

Outputs: None

#### 11.10.1.5 Return Value

None

#### 11.10.1.6 Other CSUs called by this CSU

None

#### 11.10.1.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to PwrMainRegulatorModeConfig.

##### 11.10.1.7.1 daulibstm32f4xxpwr-PwrMainRegulatorModeConfig-LLR-001

Requirement ID: H398-LLD-ANA-FNC-1603

The function shall configure power control register (PWR\_CR) to Scale 2 mode (bit 14) when scale 2 is seleted i.e, set cr of M\_PWR to (cr of M\_PWR bitwise AND with negated value of M\_PWR\_REGULATOR\_VOLTAGE\_SCALE1) when M\_PWR\_REGULATOR\_VOLTAGE\_SCALE2 is equal to pwr\_regulator\_voltage.

##### 11.10.1.7.2 daulibstm32f4xxpwr-PwrMainRegulatorModeConfig-LLR-002

Requirement ID: H398-LLD-ANA-FNC-1604

The function shall configure power control register (PWR\_CR) to Scale 1 mode (bit 14) when scale 1 is seleted i.e, set cr of M\_PWR to (cr of M\_PWR bitwise OR with M\_PWR\_REGULATOR\_VOLTAGE\_SCALE1) when M\_PWR\_REGULATOR\_VOLTAGE\_SCALE2 is not equal to pwr\_regulator\_voltage.

## 11.11 daulibstm32f4xxrcc

The daulibstm32f4xxrcc CSC provides firmware functions to manage the following functionalities of the Reset and clock control (RCC) peripheral:

* Internal/external clocks, PLL, CSS and MCO configuration
* System, AHB and APB busses clocks configuration
* Peripheral clocks configuration
* Interrupts and flags management

### 11.11.1 RccDeInit

Low Level Design Details about CSU RccDeInit will follow in the sub sections.

#### 11.11.1.1 Brief Description

The function RccDeInit Resets the RCC clock configuration to the default reset state.

#### 11.11.1.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 11.11.1.3 List of global variables accessed and modified

Accessed: None

Modified: None

#### 11.11.1.4 Parameter list (Input/Output)

Inputs: None

Outputs: None

#### 11.11.1.5 Return Value

None

#### 11.11.1.6 Other CSUs called by this CSU

None

#### 11.11.1.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to RccDeInit.

##### 11.11.1.7.1 daulibstm32f4xxrcc-RccDeInit-LLR-001

Requirement ID: H398-LLD-ANA-FNC-1614

The function shall set the HSION bit in RCC clock control register (cr).(i.e., cr of M\_RCC Bitwise OR with M\_HEX\_ONE).

##### 11.11.1.7.2 daulibstm32f4xxrcc-RccDeInit-LLR-002

Requirement ID: H398-LLD-ANA-FNC-1615

The function shall reset the cfgr register of RCC (i.e., set cfgr of M\_RCC to M\_HEX\_ZERO).

##### 11.11.1.7.3 daulibstm32f4xxrcc-RccDeInit-LLR-003

Requirement ID: H398-LLD-ANA-FNC-1616

The function shall reset HSEON, CSSON and PLLON bits of RCC clock control register (cr).(i.e., cr of M\_RCC Bitwise AND with M\_RESET\_HSEON\_CSSON\_PLLON)

##### 11.11.1.7.4 daulibstm32f4xxrcc-RccDeInit-LLR-004

Requirement ID: H398-LLD-ANA-FNC-1617

The function shall reset pllcfgr register of RCC.(i.e., pllcfgr of M\_RCC to M\_RESET\_PLLCFGR)

##### 11.11.1.7.5 daulibstm32f4xxrcc-RccDeInit-LLR-005

Requirement ID: H398-LLD-ANA-FNC-1618

The function shall reset HSEBYP bit of RCC clock control register (cr).(i.e., cr of M\_RCC Bitwise AND with M\_RESET\_HSEBYP)

##### 11.11.1.7.6 daulibstm32f4xxrcc-RccDeInit-LLR-006

Requirement ID: H398-LLD-ANA-FNC-1619

The function shall disable all interrupts by resetting RCC clock interrupt register (cir).(i.e., cir of M\_RCC to M\_HEX\_ZERO)

### 11.11.2 RccHseConfig

Low Level Design Details about CSU RccHseConfig will follow in the sub sections.

#### 11.11.2.1 Brief Description

The function RccHseConfig configures the External High Speed oscillator (HSE).

#### 11.11.2.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 11.11.2.3 List of global variables accessed and modified

Accessed: None

Modified: None

#### 11.11.2.4 Parameter list (Input/Output)

Inputs: T\_UINT8 rcc\_hse - Specifies the new state of the HSE

Outputs: None

#### 11.11.2.5 Return Value

None

#### 11.11.2.6 Other CSUs called by this CSU

None

#### 11.11.2.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to RccHseConfig

##### 11.11.2.7.1 daulibstm32f4xxrcc-RccHseConfig-LLR-001

Requirement ID: H398-LLD-ANA-FNC-1628

The function shall reset HSEON and HSEBYP bits in 3rd byte of RCC clock control register (cr) and configure with new configuration rcc\_hse.(i.e., pointer to M\_CR\_BYTE3\_ADDRESS to M\_RCC\_HSE\_OFF and pointer to M\_CR\_BYTE3\_ADDRESS to rcc\_hse).

### 11.11.3 RccWaitForHseStartUp

Low Level Design Details about CSU RccWaitForHseStartUp will follow in the sub sections.

#### 11.11.3.1 Brief Description

This function waits for HSE start-up.

#### 11.11.3.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 11.11.3.3 List of global variables accessed and modified

Accessed: None

Modified: None

#### 11.11.3.4 Parameter list (Input/Output)

Inputs: None

Outputs: None

#### 11.11.3.5 Return Value

T\_ERROR\_STATUS - Returns error status.

#### 11.11.3.6 Other CSUs called by this CSU

None

#### 11.11.3.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to RccWaitForHseStartUp.

##### 11.11.3.7.1 daulibstm32f4xxrcc-RccWaitForHseStartUp-LLR-001

Requirement ID: H398-LLD-ANA-FNC-1637

The function shall wait till HSE is ready and increments the counter. When Time out is reached as below:

a) Loop till the loop count is not equal to M\_HSE\_STARTUP\_TIMEOUT AND

b) Return value of the function RccGetFlagStatus called with the parameter M\_RCC\_FLAG\_HSERDY is equal to RESET.

##### 11.11.3.7.2 daulibstm32f4xxrcc-RccWaitForHseStartUp-LLR-002

Requirement ID: H398-LLD-ANA-FNC-1638

The function shall return SUCCESS when RCC flag is SET i.e return value of the function RccGetFlagStatus with parameter M\_RCC\_FLAG\_HSERDY is not equal to RESET.

##### 11.11.3.7.3 daulibstm32f4xxrcc-RccWaitForHseStartUp-LLR-003

Requirement ID: H398-LLD-ANA-FNC-1639

The function shall return ERROR when RCC flag is reset i.e return value of the function RccGetFlagStatus with parameter M\_RCC\_FLAG\_HSERDY is equal to RESET and returns status.

### 11.11.4 RccPllConfig

Low Level Design Details about CSU RccPllConfig will follow in the sub sections.

#### 11.11.4.1 Brief Description

The function RccPllConfig configures the main PLL clock source, multiplication and division factors.

#### 11.11.4.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 11.11.4.3 List of global variables accessed and modified

Accessed: None

Modified: None

#### 11.11.4.4 Parameter list (Input/Output)

Inputs: T\_UINT32 rcc\_pll\_source - Specifies the PLL entry clock source.

T\_UINT32 pll\_m - Specifies the division factor for PLL VCO input clock

T\_UINT32 pll\_n - Specifies the multiplication factor for PLL VCO output clock

T\_UINT32 pll\_p - Specifies the division factor for main system clock (SYSCLK)

T\_UINT32 pll\_q - Specifies the division factor for OTG FS, SDIO and RNG clocks

Outputs: None

#### 11.11.4.5 Return Value

None

#### 11.11.4.6 Other CSUs called by this CSU

None

#### 11.11.4.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to RccPllConfig.

##### 11.11.4.7.1 daulibstm32f4xxrcc-RccPllConfig-LLR-001

Requirement ID: H398-LLD-ANA-FNC-1648

The function shall configure RCC PLL configuration register with pll\_m, pll\_n, pll\_p, pll\_q and rcc\_pll\_source.(i.e., pllcfgr of M\_RCC to (pll\_m bitwise OR (pll\_n bit shifted to left by M\_SHIFT\_6) bitwise OR (((pll\_p bit shifted to right by M\_SHIFT\_1) Subtracted by M\_ONE) bit shifted to left by M\_SHIFT\_16)bitwise OR (rcc\_pll\_source) bitwise OR (pll\_q bit shifted to left by M\_SHIFT\_24)))

### 11.11.5 RccPllCmd

Low Level Design Details about CSU RccPllCmd will follow in the sub sections.

#### 11.11.5.1 Brief Description

The function RccPllCmd enables or disables the main PLL.

#### 11.11.5.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 11.11.5.3 List of global variables accessed and modified

Accessed: None

Modified: None

#### 11.11.5.4 Parameter list (Input/Output)

Inputs: T\_FUNCTIONAL\_STATE new\_state - New state of the main PLL.

Outputs: None

#### 11.11.5.5 Return Value

None

#### 11.11.5.6 Other CSUs called by this CSU

None

#### 11.11.5.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to RccPllCmd.

##### 11.11.5.7.1 daulibstm32f4xxrcc-RccPllCmd-LLR-001

Requirement ID: H398-LLD-ANA-FNC-1657

The function shall configure PLLON bit of RCC clock control register with new\_state (i.e., pointer to M\_CR\_PLLON\_BB is set to new state).

### 11.11.6 RccSysClkConfig

Low Level Design Details about CSU RccSysClkConfig will follow in the sub sections.

#### 11.11.6.1 Brief Description

The function RccSysClkConfig configures the system clock (SYSCLK).

#### 11.11.6.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 11.11.6.3 List of global variables accessed and modified

Accessed: None

Modified: None

#### 11.11.6.4 Parameter list (Input/Output)

Inputs: T\_UINT32 rcc\_sysclk\_source - specifies the clock source used as system clock.

Outputs: None

#### 11.11.6.5 Return Value

None

#### 11.11.6.6 Other CSUs called by this CSU

None

#### 11.11.6.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to RccSysClkConfig.

##### 11.11.6.7.1 daulibstm32f4xxrcc-RccSysClkConfig-LLR-001

Requirement ID: H398-LLD-ANA-FNC-1666

The function shall clear the System clock switch bits (bit 0 and 1) in RCC clock configuration register (cfgr) and configure with clock source used as system clock (rcc\_sysclk\_source)(i.e., sets the temporary register to cfgr of M\_RCC, Bitwise AND with not of M\_RCC\_CFGR\_SW, Bitwise OR with rcc\_sysclk\_source).

And store the new value (i.e., sets the cfgr of M\_RCC to temporary register).

### 11.11.7 RccGetSysClkSource

Low Level Design Details about CSU RccGetSysClkSource will follow in the sub sections.

#### 11.11.7.1 Brief Description

The function RccGetSysClkSource returns the clock source used as system clock.

#### 11.11.7.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 11.11.7.3 List of global variables accessed and modified

Accessed: None

Modified: None

#### 11.11.7.4 Parameter list (Input/Output)

Inputs: None

Outputs: None

#### 11.11.7.5 Return Value

T\_UINT8 - Returns the clock source used as system clock.

#### 11.11.7.6 Other CSUs called by this CSU

None

#### 11.11.7.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to RccGetSysClkSource.

##### 11.11.7.7.1 daulibstm32f4xxrcc-RccGetSysClkSource-LLR-001

Requirement ID: H398-LLD-ANA-FNC-1675

The function shall return the clock source used as system clock by extracting the SWS0 and SWS1 bits in RCC clock configuration register (cfgr).(i.e., returns the cfgr of M\_RCC Bitwise AND with M\_RCC\_CFGR\_SWS).

### 11.11.8 RccHclkConfig

Low Level Design Details about CSU RccHclkConfig will follow in the sub sections.

#### 11.11.8.1 Brief Description

The function RccHclkConfig configures the AHB clock (HCLK).

#### 11.11.8.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 11.11.8.3 List of global variables accessed and modified

Accessed: None

Modified: None

#### 11.11.8.4 Parameter list (Input/Output)

Inputs: T\_UINT32 rcc\_sysclk - Defines the AHB clock divider. This clock is derived from the system clock (SYSCLK).

Outputs: None

#### 11.11.8.5 Return Value

None

#### 11.11.8.6 Other CSUs called by this CSU

None

#### 11.11.8.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to RccHclkConfig.

##### 11.11.8.7.1 daulibstm32f4xxrcc-RccHclkConfig-LLR-001

Requirement ID: H398-LLD-ANA-FNC-1684

The function shall clear the AHB prescaler bits (bit 4 to 7) in RCC clock configuration register (cfgr) and configure with AHB clock divider rcc\_sysclk.(i.e., sets the temporary register to cfgr of M\_RCC, Bitwise AND with not of M\_RCC\_CFGR\_HPRE, Bitwise OR with rcc\_sysclk).

And stores the new value(i.e., cfgr of M\_RCC to temporary register).

### 11.11.9 RccPclk1Config

Low Level Design Details about CSU RccPclk1Config will follow in the sub sections.

#### 11.11.9.1 Brief Description

The function RccPclk1Config configures the Low Speed APB clock (PCLK1).

#### 11.11.9.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 11.11.9.3 List of global variables accessed and modified

Accessed: None

Modified: None

#### 11.11.9.4 Parameter list (Input/Output)

Inputs: T\_UINT32 rcc\_hclk - Defines the APB1 clock divider. This clock is derived from the AHB clock (HCLK).

Outputs: None

#### 11.11.9.5 Return Value

None

#### 11.11.9.6 Other CSUs called by this CSU

None

#### 11.11.9.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to RccPclk1Config.

##### 11.11.9.7.1 daulibstm32f4xxrcc-RccPclk1Config-LLR-001

Requirement ID: H398-LLD-ANA-FNC-1693

The function shall clear the APB Low speed prescaler bits (bit 10 to 12) in RCC clock configuration register (cfgr) and configure with APB1 clock divider rcc\_hclk.(i.e., sets the temporary register to cfgr of M\_RCC, Bitwise AND with not of M\_RCC\_CFGR\_PPRE1, Bitwise OR with rcc\_hclk ).

And stores the new value(i.e., cfgr of M\_RCC to temporary register).

### 11.11.10 RccPclk2Config

Low Level Design Details about CSU RccPclk2Config will follow in the sub sections.

#### 11.11.10.1 Brief Description

The function RccPclk2Config configures the High Speed APB clock (PCLK2).

#### 11.11.10.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 11.11.10.3 List of global variables accessed and modified

Accessed: None

Modified: None

#### 11.11.10.4 Parameter list (Input/Output)

Inputs: T\_UINT32 rcc\_hclk - Defines the APB2 clock divider. This clock is derived from the AHB clock (HCLK).

Outputs: None

#### 11.11.10.5 Return Value

None

#### 11.11.10.6 Other CSUs called by this CSU

None

#### 11.11.10.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to RccPclk2Config.

##### 11.11.10.7.1 daulibstm32f4xxrcc-RccPclk2Config-LLR-001

Requirement ID: H398-LLD-ANA-FNC-1702

The function shall clear the APB high-speed prescaler bits (bit 13 to 15) in RCC clock configuration register(cfgr) and write APB2 clock divider 'rcc\_hclk' value into APB high-speed prescaler bits of cfgr.

(i.e., sets the temporary register to cfgr of M\_RCC,

Sets the temporary register to temporary register Bitwise AND with Negation of M\_RCC\_CFGR\_PPRE2,

Sets the temporary register to temporary register Bitwise OR with rcc\_hclk left shift by M\_SHIFT\_3).

And stores the new value(cfgr of M\_RCC to temporary register).

### 11.11.11 RccAhb1PeriphClockCmd

Low Level Design Details about CSU RccAhb1PeriphClockCmd will follow in the sub sections.

#### 11.11.11.1 Brief Description

The function RccGetClocksFreq enables or disables the AHB1 peripheral clock.

#### 11.11.11.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 11.11.11.3 List of global variables accessed and modified

Accessed: None

Modified: None

#### 11.11.11.4 Parameter list (Input/Output)

Inputs: T\_UINT32 rcc\_ahb1\_periph - Specifies the AHB1 peripheral to gate its clock.

T\_FUNCTIONAL\_STATE new\_state - New state of the specified peripheral clock.

Outputs: None

#### 11.11.11.5 Return Value

None

#### 11.11.11.6 Other CSUs called by this CSU

None

#### 11.11.11.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to RccAhb1PeriphClockCmd.

##### 11.11.11.7.1 daulibstm32f4xxrcc-RccAhb1PeriphClockCmd-LLR-001

Requirement ID: H398-LLD-ANA-FNC-1727

The function shall configure RCC AHB1 peripheral clock enable register to set the bit for received peripheral rcc\_ahb1\_periph i.e, ahb1enr of M\_RCC Bitwise OR with rcc\_ahb1\_periph when new\_state is not equal to DISABLE.

##### 11.11.11.7.2 daulibstm32f4xxrcc-RccAhb1PeriphClockCmd-LLR-002

Requirement ID: H398-LLD-ANA-FNC-1728

The function shall configure RCC AHB1 peripheral clock enable register to reset the bit for received peripheral rcc\_ahb1\_periph(i.e., ahb1enr of M\_RCC Bitwise AND with not of rcc\_ahb1\_periph) when new\_state is equal to DISABLE.

### 11.11.12 RccAhb3PeriphClockCmd

Low Level Design Details about CSU RccAhb3PeriphClockCmd will follow in the sub sections.

#### 11.11.12.1 Brief Description

The function RccAhb3PeriphClockCmd enables or disables the AHB3 peripheral clock.

#### 11.11.12.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 11.11.12.3 List of global variables accessed and modified

Accessed: None

Modified: None

#### 11.11.12.4 Parameter list (Input/Output)

Inputs: T\_UINT32 rcc\_ahb3\_periph - Specifies the AHB3 peripheral to gates its clock.

T\_FUNCTIONAL\_STATE new\_state - New state of the specified peripheral clock.

Outputs: None

#### 11.11.12.5 Return Value

None

#### 11.11.12.6 Other CSUs called by this CSU

None

#### 11.11.12.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to RccAhb3PeriphClockCmd.

##### 11.11.12.7.1 daulibstm32f4xxrcc-RccAhb3PeriphClockCmd-LLR-001

Requirement ID: H398-LLD-ANA-FNC-1737

The function shall configure RCC AHB3 peripheral clock enable register to set the bit for received peripheral rcc\_ahb3\_periph (i.e., set ahb3enr of M\_RCC to ahb3enr of M\_RCC Bitwise OR with rcc\_ahb3\_periph) when new state is not equal to DISABLE.

##### 11.11.12.7.2 daulibstm32f4xxrcc-RccAhb3PeriphClockCmd-LLR-002

Requirement ID: H398-LLD-ANA-FNC-1738

The function shall configure RCC AHB3 peripheral clock enable register to reset the bit for received peripheral rcc\_ahb3\_periph (ahb3enr of M\_RCC Bitwise AND with not of rcc\_ahb3\_periph) when new state is equal to DISABLE.

### 11.11.13 RccApb1PeriphClockCmd

Low Level Design Details about CSU RccApb1PeriphClockCmd will follow in the sub sections.

#### 11.11.13.1 Brief Description

The function RccApb1PeriphClockCmd enables or disables the Low Speed APB (APB1) peripheral clock.

#### 11.11.13.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 11.11.13.3 List of global variables accessed and modified

Accessed: None

Modified: None

#### 11.11.13.4 Parameter list (Input/Output)

Inputs: T\_UINT32 rcc\_apb1\_periph - Specifies the APB1 peripheral to gate its clock.

T\_FUNCTIONAL\_STATE new\_state - New state of the specified peripheral clock.

Outputs: None

#### 11.11.13.5 Return Value

None

#### 11.11.13.6 Other CSUs called by this CSU

None

#### 11.11.13.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to RccApb1PeriphClockCmd.

##### 11.11.13.7.1 daulibstm32f4xxrcc-RccApb1PeriphClockCmd-LLR-001

Requirement ID: H398-LLD-ANA-FNC-1747

The function shall configure RCC APB1 peripheral clock enable register to set the bit for received peripheral rcc\_apb1\_periph (i.e., apb1enr of M\_RCC Bitwise OR with rcc\_apb1\_periph) when new state is not equal to DISABLE.

##### 11.11.13.7.2 daulibstm32f4xxrcc-RccApb1PeriphClockCmd-LLR-002

Requirement ID: H398-LLD-ANA-FNC-1748

The function shall configure RCC APB1 peripheral clock enable register to reset the bit for received peripheral rcc\_apb1\_periph (i.e., apb1enr of M\_RCC Bitwise AND with not of rcc\_apb1\_periph) when new state is DISABLE.

### 11.11.14 RccApb2PeriphClockCmd

Low Level Design Details about CSU RccApb2PeriphClockCmd will follow in the sub sections.

#### 11.11.14.1 Brief Description

This function enables or disables the High Speed APB (APB2) peripheral clock.

#### 11.11.14.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 11.11.14.3 List of global variables accessed and modified

Accessed: None

Modified: None

#### 11.11.14.4 Parameter list (Input/Output)

Inputs: T\_UINT32 rcc\_apb2\_periph - specifies the APB2 peripheral to gate its clock.

T\_FUNCTIONAL\_STATE new\_state - new state of the specified peripheral clock.

Outputs: None

#### 11.11.14.5 Return Value

None

#### 11.11.14.6 Other CSUs called by this CSU

None

#### 11.11.14.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to RccApb2PeriphClockCmd.

##### 11.11.14.7.1 daulibstm32f4xxrcc-RccApb2PeriphClockCmd-LLR-001

Requirement ID: H398-LLD-ANA-FNC-1757

The function shall configure RCC APB2 peripheral clock enable register to set the bit for received peripheral rcc\_apb2\_periph(i.e., apb2enr of M\_RCC Bitwise OR with rcc\_apb2\_periph) when new state is not equal to DISABLE.

##### 11.11.14.7.2 daulibstm32f4xxrcc-RccApb2PeriphClockCmd-LLR-002

Requirement ID: H398-LLD-ANA-FNC-1758

The function shall configure RCC APB2 peripheral clock enable register to reset the bit for received peripheral rcc\_apb2\_periph(i.e., apb2enr of M\_RCC Bitwise AND with not of rcc\_apb2\_periph) when new state is equal to DISABLE.

### 11.11.15 RccGetFlagStatus

Low Level Design Details about CSU RccGetFlagStatus will follow in the sub sections.

#### 11.11.15.1 Brief Description

The function RccGetFlagStatus checks whether the specified RCC flag is set or not.

#### 11.11.15.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 11.11.15.3 List of global variables accessed and modified

Accessed: None

Modified: None

#### 11.11.15.4 Parameter list (Input/Output)

Inputs: T\_UINT8 rcc\_flag - Specifies the flag to check.

Outputs: None

#### 11.11.15.5 Return Value

T\_FLAG\_STATUS - The new state of rcc\_flag (SET or RESET)

#### 11.11.15.6 Other CSUs called by this CSU

None

#### 11.11.15.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to RccGetFlagStatus.

##### 11.11.15.7.1 daulibstm32f4xxrcc-RccGetFlagStatus-LLR-001

Requirement ID: H398-LLD-ANA-FNC-1767

The function shall perform the following:

- Set register index to rcc flag right shifted with M\_SHIFT\_5

- Set status register to cr of M\_RCC when register index is equal to M\_ONE

- Set status register to bdcr of M\_RCC when register index is equal to M\_TWO

- Set status register to csr of M\_RCC when register index is not equal to M\_ONE and M\_TWO

- Set register index to rcc flag bitwise AND with M\_FLAG\_MASK

- Set bit status to SET when RESET is not equal to (status register bitwise AND of M\_ONE left shift register index otherwise set bit status to RESET).

- return bit status.

### 11.11.16 RccApb1PeriphResetCmd

Low Level Design Details about CSU RccApb1PeriphResetCmd will follow in the sub sections.

#### 11.11.16.1 Brief Description

The function RccApb1PeriphResetCmd forces or releases Low Speed APB (APB1) peripheral reset.

#### 11.11.16.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 11.11.16.3 List of global variables accessed and modified

Accessed: None

Modified: None

#### 11.11.16.4 Parameter list (Input/Output)

Inputs: T\_UINT32 rcc\_apb1\_periph - Specifies the APB1 peripheral to reset.

T\_FUNCTIONAL\_STATE new\_state - new state of the specified peripheral clock.

Outputs: None

#### 11.11.16.5 Return Value

None

#### 11.11.16.6 Other CSUs called by this CSU

None

#### 11.11.16.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to RccApb1PeriphResetCmd

##### 11.11.16.7.1 daulibstm32f4xxrcc-RccApb1PeriphResetCmd-LLR-001

Requirement ID: H398-LLD-ANA-FNC-1846

The function shall configure RCC APB1 peripheral reset register to set the bit for received peripheral rcc\_apb1\_periph(apb1rstr of M\_RCC bitwise OR with rcc\_apb1\_periph) when new state is not equal to DISABLE.

##### 11.11.16.7.2 daulibstm32f4xxrcc-RccApb1PeriphResetCmd-LLR-002

Requirement ID: H398-LLD-ANA-FNC-1847

The function shall configure RCC APB1 peripheral reset register to reset the bit for received peripheral rcc\_apb1\_periph(apb1rstr of M\_RCC bitwise AND with negation of rcc\_apb1\_periph) when new state is equal to DISABLE.

## 11.12 daulibstm32f4xxtim

This module provides firmware functions to manage the following functionalities of the TIM peripheral:

* TimeBase management
* Interrupts, DMA and flags management
* Synchronization management
* Interface management and remapping management

### 11.12.1 TimPrescalerConfig

Low Level Design Details about CSU TimPrescalerConfig will follow in the sub sections.

#### 11.12.1.1 Brief Description

The function TimPrescalerConfig configures the Timer prescaler and event generation.

#### 11.12.1.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 11.12.1.3 List of global variables accessed and modified

Accessed: None

Modified: None

#### 11.12.1.4 Parameter list (Input/Output)

Inputs: T\_UINT16 prescaler - Specifies the prescaler Register value.

T\_UINT16 tim\_psc\_reloadmode - Specifies the TIM prescaler Reload mode.

Outputs:T\_TIM\* tim\_x - TIM peripheral,where x can be 1 to 14 to select the TIM peripheral.

#### 11.12.1.5 Return Value

None

#### 11.12.1.6 Other CSUs called by this CSU

None

#### 11.12.1.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to TimPrescalerConfig.

##### 11.12.1.7.1 daulibstm32f4xxtim-TimPrescalerConfig-LLR-001

Requirement ID: H398-LLD-ANA-FNC-1777

The function shall configure prescaler register (psc) of tim\_x with the value prescaler i.e, set psc of tim\_x to prescaler.

##### 11.12.1.7.2 daulibstm32f4xxtim-TimPrescalerConfig-LLR-002

Requirement ID: H398-LLD-ANA-FNC-1778

The function shall configure event generation register (egr) of tim\_x i.e, set egr of tim\_x to tim\_psc\_reloadmode.

### 11.12.2 TimSetAutoReload

Low Level Design Details about CSU TimSetAutoReload will follow in the sub sections.

#### 11.12.2.1 Brief Description

The function TimSetAutoReload sets the timer auto reload register value.

#### 11.12.2.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 11.12.2.3 List of global variables accessed and modified

Accessed: None

Modified: None

#### 11.12.2.4 Parameter list (Input/Output)

Inputs: T\_UINT32 autoreload - Specifies the auto reload register new value.

Outputs: T\_TIM\* tim\_x - TIM peripheral, where x can be 1 to 14 to select the TIM peripheral.

#### 11.12.2.5 Return Value

None

#### 11.12.2.6 Other CSUs called by this CSU

None

#### 11.12.2.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to TimSetAutoReload.

##### 11.12.2.7.1 daulibstm32f4xxtim-TimSetAutoReload-LLR-001

Requirement ID: H398-LLD-ANA-FNC-1787

The function shall configure auto-reload register (arr) with auto reload value i.e, set arr of tim\_x to autoreload.

### 11.12.3 TimCmd

Low Level Design Details about CSU TimCmd will follow in the sub sections.

#### 11.12.3.1 Brief Description

The function TimCmd enables or disables the specified TIM peripheral.

#### 11.12.3.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 11.12.3.3 List of global variables accessed and modified

Accessed: None

Modified: None

#### 11.12.3.4 Parameter list (Input/Output)

Inputs: T\_FUNCTIONAL\_STATE new\_state - New state of the tim\_x peripheral.

T\_TIM\* tim\_x - TIM peripheral, where x can be 1 to 14 to select the TIM peripheral.

Outputs: T\_TIM\* tim\_x - TIM peripheral, where x can be 1 to 14 to select the TIM peripheral.

#### 11.12.3.5 Return Value

None

#### 11.12.3.6 Other CSUs called by this CSU

None

#### 11.12.3.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to TimCmd.

##### 11.12.3.7.1 daulibstm32f4xxtim-TimCmd-LLR-001

Requirement ID: H398-LLD-ANA-FNC-1796

The function shall configure control register 1 of timer to enable the counter (set bit 0 to 1) i.e, set cr1 of tim\_x to (cr1 of tim\_x bitwise OR with M\_TIM\_CR1\_CEN) when new\_state is ENABLE.

##### 11.12.3.7.2 daulibstm32f4xxtim-TimCmd-LLR-002

Requirement ID: H398-LLD-ANA-FNC-1797

The function shall configure control register 1 of timer to disable the counter (set bit 0 to 0) i.e, set cr1 of tim\_x to (cr1 of tim\_x bitwise AND with negated value of M\_TIM\_CR1\_CEN) when new\_state is DISABLE.

### 11.12.4 TimItConfig

Low Level Design Details about CSU TimItConfig will follow in the sub sections.

#### 11.12.4.1 Brief Description

The function TimItConfig enables or disables the specified TIM interrupts.

#### 11.12.4.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 11.12.4.3 List of global variables accessed and modified

Accessed: None

Modified: None

#### 11.12.4.4 Parameter list (Input/Output)

Inputs: T\_FUNCTIONAL\_STATE new\_state - New state of the tim\_x peripheral.

T\_TIM\* tim\_x - TIM peripheral, where x can be 1 to 14 to select the TIM peripheral.

T\_UINT16 tim\_it - specifies the TIM interrupts sources to be enabled or disabled.

Outputs: T\_TIM\* tim\_x - TIM peripheral, where x can be 1 to 14 to select the TIM peripheral.

#### 11.12.4.5 Return Value

None

#### 11.12.4.6 Other CSUs called by this CSU

None

#### 11.12.4.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to TimItConfig.

##### 11.12.4.7.1 daulibstm32f4xxtim-TimItConfig-LLR-001

Requirement ID: H398-LLD-ANA-FNC-1806

The function shall configure DMA/Interrupt enable register to enable the received Interrupt source i.e, set dier of tim\_x to (dier of tim\_x bitwise OR with tim\_it) when new\_state is ENABLE.

##### 11.12.4.7.2 daulibstm32f4xxtim-TimItConfig-LLR-002

Requirement ID: H398-LLD-ANA-FNC-1807

The function shall configure DMA/Interrupt enable register to disable the received Interrupt source i.e, set dier of tim\_x to (dier of tim\_x bitwise AND with negated value of tim\_it) when new\_state is DISABLE.

### 11.12.5 TimClearITPendingBit

Low Level Design Details about CSU TimClearITPendingBit will follow in the sub sections.

#### 11.12.5.1 Brief Description

The function TimClearITPendingBit clears the timer’s interrupt pending bits.

#### 11.12.5.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 11.12.5.3 List of global variables accessed and modified

Accessed: None

Modified: None

#### 11.12.5.4 Parameter list (Input/Output)

Inputs: T\_UINT16 tim\_it - specifies the TIM interrupts sources to be enabled or disabled.

Outputs: T\_TIM\* tim\_x - TIM peripheral, where x can be 1 to 14 to select the TIM peripheral.

#### 11.12.5.5 Return Value

None

#### 11.12.5.6 Other CSUs called by this CSU

None

#### 11.12.5.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to TimClearITPendingBit.

##### 11.12.5.7.1 daulibstm32f4xxtim-TimClearITPendingBit-LLR-001

Requirement ID: H398-LLD-ANA-FNC-1816

The function shall configure sr (status resister) of timer to clear the received interrupt pending bit i.e, set sr of tim\_x to negated value of tim\_it.

### 11.12.6 TimICInit

Low Level Design Details about CSU **TimICInit** will follow in the sub sections.

#### 11.12.6.1 Brief Description

The function **TimICInit** initializes the TIMER peripheral according to the specified tim\_ic\_init.

#### 11.12.6.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 11.12.6.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 11.12.6.4 Parameter list (Input/Output)

Inputs: T\_TIM\* tim\_x, to select the TIM peripheral.

T\_TIM\_ICINIT\* tim\_ic\_init, pointer to a T\_TIM\_ICINIT structure that contains the configuration information for specified TIM peripheral.

Outputs: T\_TIM\* tim\_x, to select the TIM peripheral.

#### 11.12.6.5 Return Value

None

#### 11.12.6.6 Other CSUs called by this CSU

Ti1Config

TimSetIC1Prescaler

Ti2Config

TimSetIC2Prescaler

Ti3Config

TimSetIC3Prescaler

Ti4Config

TimSetIC4Prescaler

#### 11.12.6.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to **TimICInit**

##### 11.12.6.7.1 daulibstm32f4xxtim-TimICInit-LLR-001

Requirement ID: H398-LLD-ANA-FNC-2442

The function shall perform the following when tim\_channel of tim\_ic\_init is equal to M\_TIM\_CHANNEL\_1

1. Call the function Ti1Config with parameters (tim\_x , tim\_ic\_polarity of tim\_ic\_init, tim\_ic\_selection of tim\_ic\_init and tim\_ic\_filter of tim\_ic\_init).
2. Call the function TimSetIC1Prescaler with parameters (tim\_x and tim\_ic\_prescaler of tim\_ic\_init).

##### 11.12.6.7.2 daulibstm32f4xxtim-TimICInit-LLR-002

Requirement ID: H398-LLD-ANA-FNC-2443

The function shall perform the following when tim\_channel of tim\_ic\_init is equal to M\_TIM\_CHANNEL\_2

1. Call the function Ti2Config with parameters (tim\_x , tim\_ic\_polarity of tim\_ic\_init, tim\_ic\_selection of tim\_ic\_init and tim\_ic\_filter of tim\_ic\_init).
2. Call the function TimSetIC2Prescaler with parameters (tim\_x and tim\_ic\_prescaler of tim\_ic\_init).

##### 11.12.6.7.3 daulibstm32f4xxtim-TimICInit-LLR-003

Requirement ID: H398-LLD-ANA-FNC-2444

The function shall perform the following when tim\_channel of tim\_ic\_init is equal to M\_TIM\_CHANNEL\_3

1. Call the function Ti3Config with parameters (tim\_x , tim\_ic\_polarity of tim\_ic\_init, tim\_ic\_selection of tim\_ic\_init and tim\_ic\_filter of tim\_ic\_init).
2. Call the function TimSetIC3Prescaler with parameters (tim\_x and tim\_ic\_prescaler of tim\_ic\_init).

##### 11.12.6.7.4 daulibstm32f4xxtim-TimICInit-LLR-004

Requirement ID: H398-LLD-ANA-FNC-2445

The function shall perform the following when tim\_channel of tim\_ic\_init is not equal to M\_TIM\_CHANNEL\_1, M\_TIM\_CHANNEL\_2, M\_TIM\_CHANNEL\_3

1. Call the function Ti4Config with parameters (tim\_x , tim\_ic\_polarity of tim\_ic\_init, tim\_ic\_selection of tim\_ic\_init and tim\_ic\_filter of tim\_ic\_init).
2. Call the function TimSetIC4Prescaler with parameters (tim\_x and tim\_ic\_prescaler of tim\_ic\_init).

### 11.12.7 TimSetIC1Prescaler

Low Level Design Details about CSU **TimSetIC1Prescaler** will follow in the sub sections.

#### 11.12.7.1 Brief Description

The function **TimSetIC1Prescaler** Sets the tim\_x Input Capture 1 prescaler

#### 11.12.7.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 11.12.7.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 11.12.7.4 Parameter list (Input/Output)

Inputs: T\_TIM\* tim\_x, to select the TIM peripheral

T\_UINT16 tim\_icpsc specifies the Input Capture1 prescaler new value.

Outputs: T\_TIM\* tim\_x

#### 11.12.7.5 Return Value

None

#### 11.12.7.6 Other CSUs called by this CSU

None

#### 11.12.7.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to **TimSetIC1Prescaler**

##### 11.12.7.7.1 daulibstm32f4xxtim-TimSetIC1Prescaler-LLR-001

Requirement ID: H398-LLD-ANA-FNC-2454

The function shall set ccmr1 of tim\_x to BITWISE AND of ccmr1 of tim\_x and BITWISE compliment of M\_TIM\_CCMR1\_IC1PSC.

##### 11.12.7.7.2 daulibstm32f4xxtim-TimSetIC1Prescaler-LLR-002

Requirement ID: H398-LLD-ANA-FNC-2455

The function shall set ccmr1 of tim\_x to BITWISE OR of ccmr1 of tim\_x and tim\_icpsc.

### 11.12.8 TimSetIC2Prescaler

Low Level Design Details about CSU **TimSetIC2Prescaler** will follow in the sub sections.

#### 11.12.8.1 Brief Description

The function **TimSetIC2Prescaler** Sets the tim\_x Input Capture 2 prescaler.

#### 11.12.8.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 11.12.8.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 11.12.8.4 Parameter list (Input/Output)

Inputs: T\_TIM\* tim\_x, to select the TIM peripheral

T\_UINT16 tim\_icpsc specifies the Input Capture2 prescaler new value.

Outputs: T\_TIM\* tim\_x

#### 11.12.8.5 Return Value

None

#### 11.12.8.6 Other CSUs called by this CSU

None

#### 11.12.8.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to **TimSetIC2Prescaler**

##### 11.12.8.7.1 daulibstm32f4xxtim-TimSetIC2Prescaler-LLR-001

Requirement ID: H398-LLD-ANA-FNC-2464

The function shall set ccmr1 of tim\_x to BITWISE AND of ccmr1 of tim\_x and BITWISE compliment of M\_TIM\_CCMR1\_IC2PSC.

##### 11.12.8.7.2 daulibstm32f4xxtim-TimSetIC2Prescaler-LLR-002

Requirement ID: H398-LLD-ANA-FNC-2465

The function shall set ccmr1 of tim\_x to BITWISE OR of ccmr1 of tim\_x and (tim\_icpsc LEFTSHIFT by M\_GET\_IC2PSC\_POS\_IN\_CCMR1.

### 11.12.9 TimSetIC3Prescaler

Low Level Design Details about CSU **TimSetIC3Prescaler** will follow in the sub sections.

#### 11.12.9.1 Brief Description

The function TimSetIC3PrescalerSets the tim\_x Input Capture 3 prescaler.

#### 11.12.9.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 11.12.9.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 11.12.9.4 Parameter list (Input/Output)

Inputs: T\_TIM\* tim\_x, to select the TIM peripheral

T\_UINT16 tim\_icpsc specifies the Input Capture3 prescaler new value.

Outputs: T\_TIM\* tim\_x

#### 11.12.9.5 Return Value

None

#### 11.12.9.6 Other CSUs called by this CSU

None

#### 11.12.9.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to **TimSetIC3Prescaler**

##### 11.12.9.7.1 daulibstm32f4xxtim-TimSetIC3Prescaler-LLR-001

Requirement ID: H398-LLD-ANA-FNC-2474

The function shall set ccmr2 of tim\_x to BITWISE AND of ccmr2 of tim\_x and BITWISE compliment of M\_TIM\_CCMR2\_IC3PSC.

##### 11.12.9.7.2 daulibstm32f4xxtim-TimSetIC3Prescaler-LLR-002

Requirement ID: H398-LLD-ANA-FNC-2475

The function shall set ccmr2 of tim\_x to BITWISE OR of ccmr2 of tim\_x and tim\_icpsc.

### 11.12.10 TimSetIC4Prescaler

Low Level Design Details about CSU **TimSetIC4Prescaler** will follow in the sub sections.

#### 11.12.10.1 Brief Description

The function **TimSetIC4Prescaler** Sets the tim\_x Input Capture 4 prescaler.

#### 11.12.10.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 11.12.10.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 11.12.10.4 Parameter list (Input/Output)

Inputs: T\_TIM\* tim\_x, to select the TIM peripheral

T\_UINT16 tim\_icpsc specifies the Input Capture4 prescaler new value.

Outputs: T\_TIM\* tim\_x

#### 11.12.10.5 Return Value

None

#### 11.12.10.6 Other CSUs called by this CSU

None

#### 11.12.10.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to **TimSetIC4Prescaler**

##### 11.12.10.7.1 daulibstm32f4xxtim-TimSetIC4Prescaler-LLR-001

Requirement ID: H398-LLD-ANA-FNC-2484

The function shall set ccmr2 of tim\_x to BITWISE AND of ccmr2 of tim\_x and BITWISE compliment of M\_TIM\_CCMR2\_IC4PSC.

##### 11.12.10.7.2 daulibstm32f4xxtim-TimSetIC4Prescaler-LLR-002

Requirement ID: H398-LLD-ANA-FNC-2485

The function shall set ccmr2 of tim\_x to BITWISE OR of ccmr2 of tim\_x and (tim\_icpsc LEFTSHIFT by M\_GET\_IC4PSC\_POS\_IN\_CCMR2).

### 11.12.11 TimDmaCmd

Low Level Design Details about CSU **TimDmaCmd** will follow in the sub sections

#### 11.12.11.1 Brief Description

The function **TimDmaCmd** Enables or disables the tim\_x's DMA Requests.

#### 11.12.11.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 11.12.11.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 11.12.11.4 Parameter list (Input/Output)

Inputs : T\_TIM\* tim\_x - to select the TIM peripheral.

T\_UINT16 tim\_dmasource - specifies the DMA Request sources.

T\_FUNCTIONAL\_STATE new\_state - state of DMA Request

Outputs: T\_TIM\* tim\_x

#### 11.12.11.5 Return Value

None

#### 11.12.11.6 Other CSUs called by this CSU

None

#### 11.12.11.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to **TimDmaCmd**

##### 11.12.11.7.1 daulibstm32f4xxtim-TimDmaCmd-LLR-001

Requirement ID: H398-LLD-ANA-FNC-2494

The function shall set dier of tim\_x to BITWISE OR of dier of tim\_x and tim\_dmasource when new state is not equal to DISABLE.

##### 11.12.11.7.2 daulibstm32f4xxtim-TimDmaCmd-LLR-002

Requirement ID: H398-LLD-ANA-FNC-2495

The function shall set dier of tim\_x to BITWISE AND of dier of tim\_x and (BITWISE compliment of tim\_dmasource).

### 11.12.12 Ti1Config

Low Level Design Details about CSU **Ti1Config** will follow in the sub sections.

#### 11.12.12.1 Brief Description

The function Ti1ConfigConfigure the TI1 as Input

#### 11.12.12.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 11.12.12.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 11.12.12.4 Parameter list (Input/Output)

Inputs : T\_TIM\* tim\_x - to select the TIM peripheral.

T\_UINT16 tim\_ic\_polarity -IN: The Input Polarity.

T\_UINT16 tim\_ic\_selection - IN: specifies the input to be used.

T\_UINT16 tim\_ic\_filter - IN: Specifies the Input Capture Filter

Outputs : T\_TIM\* tim\_x

#### 11.12.12.5 Return Value

None

#### 11.12.12.6 Other CSUs called by this CSU

None

#### 11.12.12.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to **Ti1Config**

##### 11.12.12.7.1 daulibstm32f4xxtim-Ti1Config-LLR-001

Requirement ID: H398-LLD-ANA-FNC-2504

The function shall perform the following:

1. Set ccer of tim\_x to BITWISE AND of ccer of tim\_x and (BITWISE compliment of M\_TIM\_CCER\_CC1E).
2. Set tempccmr1 to ccmr1 of tim\_x.
3. Set tempccer to ccer of tim\_x.
4. Update tempccmr1 with (tempccmr1 BITWISE AND (BITWISE compliment of M\_TIM\_CCMR1\_CC1S) BITWISE AND (BITWISE compliment of M\_TIM\_CCMR1\_IC1F)).
5. Update tempccmr1 with tempccmr1 BITWISE OR (tim\_ic\_selection BITWISE OR (tim\_ic\_filter LEFTSHIFT by M\_FOUR)).
6. Update tempccer with tempccer BITWISE AND (BITWISE compliment (M\_TIM\_CCER\_CC1P BITWISE OR M\_TIM\_CCER\_CC1NP)).
7. Update tempccer with tempccer BITWISE OR (tim\_ic\_polarity BITWISE OR M\_TIM\_CCER\_CC1E).
8. Set ccmr1 of tim\_x to tempccmr1.
9. Set ccer of tim\_x to tempccer.

### 11.12.13 Ti2Config

Low Level Design Details about CSU Ti2Configwill follow in the sub sections.

#### 11.12.13.1 Brief Description

The function **Ti2Config** Configure the TI2 as Input

#### 11.12.13.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 11.12.13.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 11.12.13.4 Parameter list (Input/Output)

Inputs : T\_TIM\* tim\_x - to select the TIM peripheral.

T\_UINT16 tim\_ic\_polarity - The Input Polarity.

T\_UINT16 tim\_ic\_selection - specifies the input to be used.

T\_UINT16 tim\_ic\_filter - Specifies the Input Capture Filter

Outputs : T\_TIM\* tim\_x

#### 11.12.13.5 Return Value

None

#### 11.12.13.6 Other CSUs called by this CSU

None

#### 11.12.13.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to **Ti2Config**

##### 11.12.13.7.1 daulibstm32f4xxtim-Ti2Config-LLR-001

Requirement ID: H398-LLD-ANA-FNC-2513

The function shall perform the following:

1. Set ccer of tim\_x to BITWISE AND of ccer of tim\_x and (BITWISE compliment of M\_TIM\_CCER\_CC2E).
2. Set tempccmr1 to ccmr1 of tim\_x.
3. Set tempccer to ccer of tim\_x.
4. Set selectpollty to tim\_ic\_polarity LEFTSHIFT by M\_FOUR.
5. Update tempccmr1 with (tempccmr1 BITWISE AND (BITWISE compliment of M\_TIM\_CCMR1\_CC2S) BITWISE AND (BITWISE compliment of M\_TIM\_CCMR1\_IC2F)).
6. Update tempccmr1 with BITWISE OR of tempccmr1 and (tim\_ic\_filter LEFTSHIFT by M\_TWELVE).
7. Update tempccmr1 with BITWISE OR of tempccmr1 and (tim\_ic\_selection LEFTSHIFT by M\_GET\_IC\_SELECTION\_POS\_IN\_CCMR1).
8. Update tempccer with tempccer BITWISE AND (BITWISE compliment (M\_TIM\_CCER\_CC2P BITWISE OR M\_TIM\_CCER\_CC2NP)).
9. Update tempccer with tempccer BITWISE OR (selectpollty BITWISE OR M\_TIM\_CCER\_CC2E).
10. Set ccmr1 of tim\_x to tempccmr1.
11. Set ccer of tim\_x to tempccer.

### 11.12.14 Ti3Config

Low Level Design Details about CSU Ti3Configwill follow in the sub sections.

#### 11.12.14.1 Brief Description

The function Ti3ConfigConfigure the TI3 as Input.

#### 11.12.14.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 11.12.14.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 11.12.14.4 Parameter list (Input/Output)

Inputs : T\_TIM\* tim\_x - to select the TIM peripheral.

T\_UINT16 tim\_ic\_polarity - The Input Polarity.

T\_UINT16 tim\_ic\_selection - specifies the input to be used.

T\_UINT16 tim\_ic\_filter - Specifies the Input Capture Filter

Outputs : T\_TIM\* tim\_x

#### 11.12.14.5 Return Value

None

#### 11.12.14.6 Other CSUs called by this CSU

None

#### 11.12.14.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to **Ti3Config**

##### 11.12.14.7.1 daulibstm32f4xxtim-Ti3Config-LLR-001

Requirement ID: H398-LLD-ANA-FNC-2522

The function shall perform the following:

1. Set ccer of tim\_x to BITWISE AND of ccer of tim\_x and (BITWISE compliment of M\_TIM\_CCER\_CC3E).
2. Set tempccmr2 to ccmr2 of tim\_x.
3. Set tempccer to ccer of tim\_x.
4. Set selectpollty to tim\_ic\_polarity LEFTSHIFT by M\_GET\_IC\_POLARITY.
5. Update tempccmr2 with (tempccmr2 BITWISE AND (BITWISE compliment of M\_TIM\_CCMR1\_CC1S) BITWISE AND (BITWISE compliment of M\_TIM\_CCMR2\_IC3F)).
6. Update tempccmr2 with tempccmr2 BITWISE OR (tim\_ic\_selection BITWISE OR (tim\_ic\_filter LEFTSHIFT by M\_FOUR)).
7. Update tempccer with tempccer BITWISE AND (BITWISE compliment (M\_TIM\_CCER\_CC3P BITWISE OR M\_TIM\_CCER\_CC3NP)).
8. Update tempccer with tempccer BITWISE OR (selectpollty BITWISE OR M\_TIM\_CCER\_CC3E).
9. Set ccmr2 of tim\_x to tempccmr2.
10. Set ccer of tim\_x to tempccer.

### 11.12.15 Ti4Config

Low Level Design Details about CSU Ti4Configwill follow in the sub sections.

#### 11.12.15.1 Brief Description

The function Ti4ConfigConfigure the TI4 as Input.

#### 11.12.15.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 11.12.15.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 11.12.15.4 Parameter list (Input/Output)

Inputs : T\_TIM\* tim\_x - to select the TIM peripheral.

T\_UINT16 tim\_ic\_polarity - The Input Polarity.

T\_UINT16 tim\_ic\_selection - specifies the input to be used.

T\_UINT16 tim\_ic\_filter - Specifies the Input Capture Filter

Outputs : T\_TIM\* tim\_x

#### 11.12.15.5 Return Value

None

#### 11.12.15.6 Other CSUs called by this CSU

None

#### 11.12.15.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to **Ti4Config**

##### 11.12.15.7.1 daulibstm32f4xxtim-Ti4Config-LLR-001

Requirement ID: H398-LLD-ANA-FNC-2531

The function shall perform the following:

1. Set ccer of tim\_x to BITWISE AND of ccer of tim\_x and (BITWISE compliment of M\_TIM\_CCER\_CC4E).
2. Set tempccmr2 to ccmr2 of tim\_x.
3. Set tempccer to ccer of tim\_x.
4. Set selectpollty to tim\_ic\_polarity LEFTSHIFT by M\_TWELVE.
5. Update tempccmr2 with (tempccmr2 BITWISE AND (BITWISE compliment of M\_TIM\_CCMR1\_CC2S) BITWISE AND (BITWISE compliment of M\_TIM\_CCMR1\_IC2F)).
6. Update tempccmr2 with BITWISE OR of tempccmr2 and (tim\_ic\_selection LEFTSHIFT by M\_EIGHT).
7. Update tempccmr2 with BITWISE OR of tempccmr2 and (tim\_ic\_filter LEFTSHIFT by M\_TWELVE).
8. Update tempccer with tempccer BITWISE AND (BITWISE compliment (M\_TIM\_CCER\_CC4P BITWISE OR M\_TIM\_CCER\_CC4NP)).
9. Update tempccer with tempccer BITWISE OR (selectpollty BITWISE OR M\_TIM\_CCER\_CC4E).
10. Set ccmr2 of tim\_x to tempccmr2.
11. Set ccer of tim\_x to tempccer.

### 11.12.16 TIMSelectInputTrigger

Low Level Design Details about CSU TIMSelectInputTriggerwill follow in the sub sections.

#### 11.12.16.1 Brief Description

The function TIMSelectInputTriggerSelects the Input Trigger source.

#### 11.12.16.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 11.12.16.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 11.12.16.4 Parameter list (Input/Output)

Inputs : T\_TIM\* tim\_x - to select the TIM peripheral.

T\_UINT16 TIM\_InputTriggerSource - The Input Trigger source

Outputs : T\_TIM\* tim\_x

#### 11.12.16.5 Return Value

None

#### 11.12.16.6 Other CSUs called by this CSU

None

#### 11.12.16.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to TIMSelectInputTrigger.

##### 11.12.16.7.1 daulibstm32f4xxtim-TIMSelectInputTrigger-LLR-001

Requirement ID: H398-LLD-ANA-FNC-2540

The function shall perform the following:

1. Set temp smcr to smcr of TIMx.
2. Update temp smcr to BITWISE AND of temp\_smcr and (BITWISE compliment of M\_TIM\_SMCR\_TS).
3. Update temp smcr to BITWISE OR of temp\_smcr and TIM\_InputTriggerSource.
4. Set smcr of TIMx to temp smcr.

### 11.12.17 TIMSelectSlaveMode

Low Level Design Details about CSU TIMSelectSlaveModewill follow in the sub sections.

#### 11.12.17.1 Brief Description

The function TIMSelectSlaveModespecifies the Timer Slave Mode.

#### 11.12.17.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 11.12.17.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 11.12.17.4 Parameter list (Input/Output)

Inputs : T\_TIM\* tim\_x - to select the TIM peripheral.

T\_UINT16 TIM\_SlaveMode - specifies the Timer Slave Mode.

Outputs : T\_TIM\* tim\_x

#### 11.12.17.5 Return Value

None

#### 11.12.17.6 Other CSUs called by this CSU

None

#### 11.12.17.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to TIMSelectSlaveMode.

##### 11.12.17.7.1 daulibstm32f4xxtim-TIMSelectSlaveMode-LLR-001

Requirement ID: H398-LLD-ANA-FNC-2549

The function shall perform the following:

1. Set smcr of TIMx to BITWISE AND of smcr of TIMx and (BITWISE compliment of M\_TIM\_SMCR\_SMS).
2. Set smcr of TIMx to BITWISE OR of smcr of TIMx and TIM\_SlaveMode.

### 11.12.18 TIMSelectMasterSlaveMode

Low Level Design Details about CSU TIMSelectMasterSlaveModewill follow in the sub sections.

#### 11.12.18.1 Brief Description

The function TIMSelectMasterSlaveMode Sets or Resets the TIMx Master/Slave Mode.

#### 11.12.18.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 11.12.18.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 11.12.18.4 Parameter list (Input/Output)

Inputs : T\_TIM\* tim\_x - to select the TIM peripheral.

T\_UINT16 TIM\_MasterSlaveMode - specifies the Timer Master Slave Mode.

Outputs : T\_TIM\* tim\_x

#### 11.12.18.5 Return Value

None

#### 11.12.18.6 Other CSUs called by this CSU

None

#### 11.12.18.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to TIMSelectMasterSlaveMode.

##### 11.12.18.7.1 daulibstm32f4xxtim-TIMSelectMasterSlaveMode-LLR-001

Requirement ID: H398-LLD-ANA-FNC-2558

The function shall perform the following:

1. Set smcr of TIMx to BITWISE AND of smcr of TIMx and (BITWISE compliment of M\_TIM\_SMCR\_MSM).
2. Set smcr of TIMx to BITWISE OR of smcr of TIMx and TIM\_MasterSlaveMode.

### 11.12.19 TIMUpdateRequestConfig

Low Level Design Details about CSU TIMUpdateRequestConfig will follow in the sub sections.

#### 11.12.19.1 Brief Description

The function TIMUpdateRequestConfig Configures the TIMx Update Request Interrupt source.

#### 11.12.19.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

11.12.19.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 11.12.19.4 Parameter list (Input/Output)

Inputs : T\_TIM\* tim\_x - to select the TIM peripheral.

T\_UINT16 TIM\_UpdateSource - specifies the Update source.

Outputs : T\_TIM\* tim\_x

#### 11.12.19.5 Return Value

None

#### 11.12.19.6 Other CSUs called by this CSU

None

#### 11.12.19.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to TIMUpdateRequestConfig.

##### 11.12.19.7.1 daulibstm32f4xxtim-TIMUpdateRequestConfig-LLR-001

Requirement ID: H398-LLD-ANA-FNC-2567

The function shall set cr1 of TIMx to BITWISE OR of cr1 of TIMx and M\_TIM\_CR1\_URS when TIM\_UpdateSource is not equal to M\_TIM\_UPDATESOURCE\_GLOBAL,

Otherwise set cr1 of TIMx to BITWISE AND of cr1 of TIMx and (BITWISE compliment of M\_TIM\_CR1\_URS).

### 11.12.20 TIMGetCapture1

Low Level Design Details about CSU TIMGetCapture1 will follow in the sub sections.

#### 11.12.20.1 Brief Description

The function TIMGetCapture1 Gets the TIMx Input Capture 1 value.

#### 11.12.20.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 11.12.20.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 11.12.20.4 Parameter list (Input/Output)

Inputs : T\_TIM\* tim\_x - to select the TIM peripheral.

Outputs : T\_TIM\* tim\_x

#### 11.12.20.5 Return Value

Ccr2 of TIMx - Capture Compare 2 Register value

#### 11.12.20.6 Other CSUs called by this CSU

None

#### 11.12.20.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to TIMGetCapture1.

##### 11.12.20.7.1 daulibstm32f4xxtim-TIMGetCapture1-LLR-001

Requirement ID: H398-LLD-ANA-FNC-2585

The function shall return ccr1 of TIMx.

### 11.12.21 TIMGetCapture2

Low Level Design Details about CSU TIMGetCapture2 will follow in the sub sections.

#### 11.12.21.1 Brief Description

The function TIMGetCapture2 Gets the TIMx Input Capture 2 value.

#### 11.12.21.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 11.12.21.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 11.12.21.4 Parameter list (Input/Output)

Inputs : T\_TIM\* tim\_x - to select the TIM peripheral.

Outputs : T\_TIM\* tim\_x

#### 11.12.21.5 Return Value

Ccr2 of TIMx - Capture Compare 2 Register value

#### 11.12.21.6 Other CSUs called by this CSU

None

#### 11.12.21.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to TIMGetCapture2.

##### 11.12.21.7.1 daulibstm32f4xxtim-TIMGetCapture2-LLR-001

Requirement ID: H398-LLD-ANA-FNC-2586

The function shall return ccr2 of TIMx.

### 11.12.22 TIMGetFlagStatus

Low Level Design Details about CSU TIMGetFlagStatus will follow in the sub sections.

#### 11.12.22.1 Brief Description

The function TIMGetFlagStatus Checks whether the specified TIM flag is set or not.

#### 11.12.22.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 11.12.22.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 11.12.22.4 Parameter list (Input/Output)

Inputs : T\_TIM\* tim\_x - to select the TIM peripheral.

T\_UINT16 TIM\_FLAG - specifies the flag to check

Outputs : T\_TIM\* tim\_x

#### 11.12.22.5 Return Value

Status - The new state of TIM\_FLAG (SET or RESET)

#### 11.12.22.6 Other CSUs called by this CSU

None

#### 11.12.22.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to TIMGetFlagStatus.

##### 11.12.22.7.1 daulibstm32f4xxtim-TIMGetFlagStatus-LLR-001

Requirement ID: H398-LLD-ANA-FNC-2595

The function shall set the status of bit to SET when the following condition is satisfied:

1. (sr of TIMx BITWISE AND TIM\_FLAG) not equal to RESET.

Otherwise set status of bit to RESET.

ii) return status of bit.

## 11.13 daulibstm32f4xxspi

Low Level Design Details about CSU SPI will follow in the sub sections.

### 11.13.1 SPIInit

Low Level Design Details about CSU SPIInit will follow in the sub sections.

#### 11.13.1.1 Brief Description

The function SPIInit initializes the SPIx peripheral according to the specified parameters in the SPI\_InitStruct.

#### 11.13.1.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 11.13.1.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 11.13.1.4 Parameter list (Input/Output)

Inputs: T\_SPI\* SPIx, SPI\_InitTypeDef\* SPI\_InitStruct

Outputs: T\_SPI\* SPIx

#### 11.13.1.5 Return Value

None

#### 11.13.1.6 Other CSUs called by this CSU

assert\_param

M\_IS\_SPI\_DIRECTION\_MODE

M\_IS\_SPI\_MODE

M\_IS\_SPI\_DATASIZE

M\_IS\_SPI\_CPOL

M\_IS\_SPI\_CPHA

M\_IS\_SPI\_NSS

M\_IS\_SPI\_BAUDRATE\_PRESCALER

M\_IS\_SPI\_FIRST\_BIT

M\_IS\_SPI\_CRC\_POLYNOMIAL

M\_IS\_SPI\_ALL\_PERIPH

IS\_FUNCTIONAL\_STATE

M\_IS\_SPI\_ALL\_PERIPH\_EXT

IS\_FUNCTIONAL\_STATE

M\_IS\_SPI\_I2S\_CONFIG\_IT

#### 11.13.1.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to SPIInit.

##### 11.13.1.7.1 daulibstm32f4xxspi-SPIInit-LLR-001

Requirement ID: H398-LLD-ANA-FNC-2396

The function shall perform the following:

* Call the function assert\_param with following parameters:
* Call the function M\_IS\_SPI\_ALL\_PERIPH with parameter SPIx.
* Call the function assert\_param with following parameters:
* Call the function M\_IS\_SPI\_DIRECTION\_MODE with parameter SPI\_Direction of SPI\_InitStruct.
* Call the function assert\_param with following parameters:
* Call the function M\_IS\_SPI\_MODE with parameter SPI\_Mode of SPI\_InitStruct.
* Call the function assert\_param with following parameters:
* Call the function M\_IS\_SPI\_DATASIZE with parameter SPI\_DataSize of SPI\_InitStruct.
* Call the function assert\_param with following parameters:
* Call the function M\_IS\_SPI\_CPOL with parameter SPI\_CPOL of SPI\_InitStruct.
* Call the function assert\_param with following parameters:
* Call the function M\_IS\_SPI\_CPHA with parameter SPI\_CPHA of SPI\_InitStruct.
* Call the function assert\_param with following parameters:
* Call the function M\_IS\_SPI\_NSS with parameter SPI\_NSS of SPI\_InitStruct.
* Call the function assert\_param with following parameters:
* Call the function M\_IS\_SPI\_BAUDRATE\_PRESCALER with parameter SPI\_BaudRatePrescaler of SPI\_InitStruct.
* Call the function assert\_param with following parameters:
* Call the function M\_IS\_SPI\_FIRST\_BIT with parameter SPI\_FirstBit of SPI\_InitStruct.
* Call the function assert\_param with following parameters:
* Call the function M\_IS\_SPI\_CRC\_POLYNOMIAL with parameter SPI\_CRCPolynomial of SPI\_InitStruct.
* Set temporary register to CR1 of SPIx
* Update temporary register to bitwise AND of temporary register and CR1\_CLEAR\_MASK
* Update temporary register to bitwise OR of temporary register and (bitwise OR of SPI\_Direction of SPI\_InitStruct, SPI\_Mode of SPI\_InitStruct, SPI\_DataSize of SPI\_InitStruct, SPI\_CPOL of SPI\_InitStruct, SPI\_CPHA of SPI\_InitStruct, SPI\_NSS of SPI\_InitStruct, SPI\_BaudRatePrescaler of SPI\_InitStruct and SPI\_FirstBit of SPI\_InitStruct)
* Update CR1 of SPIx temporary register
* Set I2SCFGR of SPIx to bitwise AND of I2SCFGR of SPIx and negation of SPI\_I2SCFGR\_I2SMOD
* Set CRCPR of SPIx to SPI\_CRCPolynomial of SPI\_InitStruct.

### 11.13.2 SPICmd

Low Level Design Details about CSU SPICmd will follow in the sub sections.

#### 11.13.2.1 Brief Description

Enables or disables the specified SPI peripheral.

#### 11.13.2.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 11.13.2.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 11.13.2.4 Parameter list (Input/Output)

Inputs: T\_FUNCTIONAL\_STATE NewState, T\_SPI\* SPIx

Outputs: T\_SPI\* SPIx

#### 11.13.2.5 Return Value

None

#### 11.13.2.6 Other CSUs called by this CSU

assert\_param

M\_IS\_SPI\_ALL\_PERIPH

IS\_FUNCTIONAL\_STATE

#### 11.13.2.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to SPICmd.

##### 11.13.2.7.1 daulibstm32f4xxspi-SPICmd-LLR-001

Requirement ID: H398-LLD-ANA-FNC-2405

The function shall perform the following:

1. Call the function assert\_param with following parameters
2. Call the function M\_IS\_SPI\_ALL\_PERIPH with parameter SPIx.

b) Call the function assert\_param with following parameters

1. Call the function IS\_FUNCTIONAL\_STATE with NewState.

c) Set CR1 of SPIx to BITWISE OR of CR1 of SPIx and SPI\_CR1\_SPE when NewState is not equal to DISABLE Otherwise, Set CR1 of SPIx to BITWISE AND of CR1 of SPIx and (BITWISE compliment of SPI\_CR1\_SPE).

### 11.13.3 SPII2SReceiveData

Low Level Design Details about CSU SPII2SReceiveData will follow in the sub sections.

#### 11.13.3.1 Brief Description

The function SPII2SReceiveData Returns the most recent received data by the SPIx/I2Sx peripheral.

#### 11.13.3.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 11.13.3.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 11.13.3.4 Parameter list (Input/Output)

Inputs: T\_SPI\* SPIx

Outputs: T\_SPI\* SPIx

#### 11.13.3.5 Return Value

Return DR of SPIx

#### 11.13.3.6 Other CSUs called by this CSU

assert\_param

M\_IS\_SPI\_ALL\_PERIPH\_EXT

#### 11.13.3.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to SPII2SReceiveData

##### 11.13.3.7.1 daulibstm32f4xxspi-SPII2SReceiveData-LLR-001

Requirement ID: H398-LLD-ANA-FNC-2414

The function shall perform the following:

1. Call the function assert\_param with following parameters
2. Call the function M\_IS\_SPI\_ALL\_PERIPH\_EXT with parameter SPIx.

b) Function returns DR of SPIx.

### 11.13.4 SPII2SSendData

Low Level Design Details about CSU SPII2SSendData will follow in the sub sections.

#### 11.13.4.1 Brief Description

The function SPII2SSendData Transmits a Data through the SPIx/I2Sx peripheral.

#### 11.13.4.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 11.13.4.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 11.13.4.4 Parameter list (Input/Output)

Inputs: T\_SPI\* SPIx

T\_UINT16 Data

Outputs: T\_SPI\* SPIx

#### 11.13.4.5 Return Value

None

#### 11.13.4.6 Other CSUs called by this CSU

assert\_param

M\_IS\_SPI\_ALL\_PERIPH\_EXT

#### 11.13.4.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to SPII2SSendData

##### 11.13.4.7.1 daulibstm32f4xxspi-SPII2SSendData-LLR-001

Requirement ID: H398-LLD-ANA-FNC-2423

The function shall perform the following:

1. Call the function assert\_param with following parameters
2. Call the function M\_IS\_SPI\_ALL\_PERIPH\_EXT with parameter SPIx.

b) Set DR of SPIx to Data.

### 11.13.5 SPII2SITConfig

Low Level Design Details about CSU SPII2SITConfig will follow in the sub sections.

#### 11.13.5.1 Brief Description

The function SPII2SITConfig Enables or disables the specified SPI/I2S interrupts.

#### 11.13.5.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

#### 11.13.5.3 List of global variables accessed and modified

Accessed : None

Modified : None

#### 11.13.5.4 Parameter list (Input/Output)

Inputs: T\_SPI\* SPIx

T\_UINT8 SPI\_I2S\_IT,

T\_FUNCTIONAL\_STATE NewState

Outputs: T\_SPI\* SPIx

#### 11.13.5.5 Return Value

None

#### 11.13.5.6 Other CSUs called by this CSU

assert\_param

M\_IS\_SPI\_ALL\_PERIPH\_EXT

IS\_FUNCTIONAL\_STATE

M\_IS\_SPI\_I2S\_CONFIG\_IT

#### 11.13.5.7 Description of list of LLRs allocated

The following section will list the LLRs allocated to SPII2SITConfig

##### 11.13.5.7.1 daulibstm32f4xxspi-SPII2SITConfig-LLR-001

Requirement ID: H398-LLD-ANA-FNC-2432

The function shall perform the following:

1. Call the function assert\_param with following parameters
2. Call the function M\_IS\_SPI\_ALL\_PERIPH\_EXT with parameter SPIx.

b) Call the function assert\_param with following parameters

ii. Call the function IS\_FUNCTIONAL\_STATE with parameter NewState.

c) Call the function assert\_param with following parameters

iii. Call the function M\_IS\_SPI\_I2S\_CONFIG\_IT with parameter SPI\_I2S\_IT.

d) Set it\_pos to (SPI\_I2S\_IT RIGHTSHIFT by four).

e) Set it\_mask to one LEFTSHIFT it\_pos.

##### 11.13.5.7.2 daulibstm32f4xxspi-SPII2SITConfig-LLR-002

Requirement ID: H398-LLD-ANA-FNC-2433

The function shall set CR2 of SPIx to BITWISE OR of CR2 of SPIx and it\_mask, when NewState is not equal to DISABLE,Otherwise set CR2 of SPIx to BITWISE AND of CR2 of SPIx and BITWISE compliment of it\_mask.

# 12 Software Low Level Requirements - MCD Software

The Software Low Level Requirements-MCD Software section specifies the Software Low Level Requirements for Analog Module MCD .

Refer section 14 Appendix A: Data Structures of H398-003-001-ANA for Data Structures and Enumerations.

## 12.1 dauanasensor

Implementation of all sensor configuration data.

### 12.1.1 Brief Description

The module dauanasensor.c contains Sensor configuration data.

### 12.1.2 List of HLRs allocated

The list of HLRs allocated to this CSU is available in the Bi-Directional Traceability Matrix from SLL to SRS/SAD (H398-003-012-ANA).

### 12.1.3 Description of list of LLRs allocated

The following section will list the LLRs allocated to dauanasensor.

#### 12.1.3.1 dauanasensor-LLR-001

Requirement ID: H398-LLD-ANA-FNC-1831

The CSC dauanasensor shall set the Analog Configuration Software Part Number to "H108E-674 -1.03 "

#### 12.1.3.2 dauanasensor-LLR-002

Requirement ID: H398-LLD-ANA-FNC-1832

The analog input channel configuration section shall have the format specified in Table: Analog input channel configuration format.

Table: Analog input channel configuration format

Table : Analog input channel configuration format

|  |  |  |
| --- | --- | --- |
| **Channel** | **Parameter** | **Configured Value** |
| BIT\_1 | u8\_Chan\_State | 1 - ENABLE |
| u8\_Default\_State | 0 - DISABLE |
| i16\_Default\_Val | 0 |
| b\_Calibrate | FALSE |
| i16\_Low\_Cal\_Point | 0 |
| i16\_High\_Cal\_Point | 0 |
| c\_Type | CAL\_DIS\_V |
| i16\_Low\_Cal\_Disp | 5 |
| i16\_High\_Cal\_Disp | 50 |
| i16\_Resolution | 10 |
| b\_excite\_req | FALSE |
| s\_Table | Ai16\_builtintest\_adc,  Ai16\_builtintest\_vdc |
| i16\_Min\_Range | -247 |
| i16\_Max\_Range | 252 |
| L\_SENSE\_1 SPARE | u8\_Chan\_State | 0 - DISABLE |
| u8\_Default\_State | 0 - DISABLE |
| i16\_Default\_Val | 0 |
| b\_Calibrate | FALSE |
| i16\_Low\_Cal\_Point | 0 |
| i16\_High\_Cal\_Point | 0 |
| c\_Type | CAL\_DIS\_V |
| i16\_Low\_Cal\_Disp | 0 |
| i16\_High\_Cal\_Disp | 0 |
| i16\_Resolution | 0 |
| b\_excite\_req | FALSE |
| s\_Table | Ai16\_adc\_spare  Ai16\_spare |
| i16\_Min\_Range | 0 |
| i16\_Max\_Range | 0 |
| L\_SENSE\_2 SPARE | u8\_Chan\_State | 0 - DISABLE |
| u8\_Default\_State | 0 - DISABLE |
| i16\_Default\_Val | 0 |
| b\_Calibrate | FALSE |
| i16\_Low\_Cal\_Point | 0 |
| i16\_High\_Cal\_Point | 0 |
| c\_Type | CAL\_DIS\_V |
| i16\_Low\_Cal\_Disp | 0 |
| i16\_High\_Cal\_Disp | 0 |
| i16\_Resolution | 0 |
| b\_excite\_req | FALSE |
| s\_Table | Ai16\_adc\_spare  Ai16\_spare |
| i16\_Min\_Range | 0 |
| i16\_Max\_Range | 0 |
| LL\_1 TOT1 | u8\_Chan\_State | 1 – ENABLE |
| u8\_Default\_State | 0 – DISABLE |
| i16\_Default\_Val | 0 |
| b\_Calibrate | TRUE |
| i16\_Low\_Cal\_Point | 2451 |
| i16\_High\_Cal\_Point | 29221 |
| c\_Type | CAL\_DIS\_TEMPC |
| i16\_Low\_Cal\_Disp | 1000 |
| i16\_High\_Cal\_Disp | 12000 |
| i16\_Resolution | 10 |
| b\_excite\_req | FALSE |
| s\_Table | Ai16\_typek\_adc,  Ai16\_typek\_c |
| i16\_Min\_Range | -2000 |
| i16\_Max\_Range | 12940 |
| LL\_2 TOT2 | u8\_Chan\_State | 1 - ENABLE |
| u8\_Default\_State | 0 - DISABLE |
| i16\_Default\_Val | 0 |
| b\_Calibrate | TRUE |
| i16\_Low\_Cal\_Point | 2451 |
| i16\_High\_Cal\_Point | 29221 |
| c\_Type | CAL\_DIS\_TEMPC |
| i16\_Low\_Cal\_Disp | 1000 |
| i16\_High\_Cal\_Disp | 12000 |
| i16\_Resolution | 10 |
| b\_excite\_req | FALSE |
| s\_Table | Ai16\_typek\_adc,  Ai16\_typek\_c |
| i16\_Min\_Range | -2000 |
| i16\_Max\_Range | 12940 |
| LL\_3 EOP1 | u8\_Chan\_State | 1 - ENABLE |
| u8\_Default\_State | 0 - DISABLE |
| i16\_Default\_Val | 0 |
| b\_Calibrate | TRUE |
| i16\_Low\_Cal\_Point | - 25749 |
| i16\_High\_Cal\_Point | 25748 |
| c\_Type | CAL\_DIS\_MV |
| i16\_Low\_Cal\_Disp | -25000 |
| i16\_High\_Cal\_Disp | 25000 |
| i16\_Resolution | 100 |
| b\_excite\_req | FALSE |
| s\_Table | Ai16\_adc\_vds300,  Ai16\_mv\_vds300 |
| i16\_Min\_Range | -31816 |
| i16\_Max\_Range | 31816 |
| LL\_4 EOP2 | u8\_Chan\_State | 1 - ENABLE |
| u8\_Default\_State | 0 - DISABLE |
| i16\_Default\_Val | 0 |
| b\_Calibrate | TRUE |
| i16\_Low\_Cal\_Point | -25749 |
| i16\_High\_Cal\_Point | 25748 |
| c\_Type | CAL\_DIS\_MV |
| i16\_Low\_Cal\_Disp | -25000 |
| i16\_High\_Cal\_Disp | 25000 |
| i16\_Resolution | 100 |
| b\_excite\_req | FALSE |
| s\_Table | Ai16\_adc\_vds300,  Ai16\_mv\_vds300 |
| i16\_Min\_Range | -31816 |
| i16\_Max\_Range | 31816 |
| RTD\_1 EOT1 | u8\_Chan\_State | 1 - ENABLE |
| u8\_Default\_State | 0 - DISABLE |
| i16\_Default\_Val | 0 |
| b\_Calibrate | TRUE |
| i16\_Low\_Cal\_Point | 5224 |
| i16\_High\_Cal\_Point | 26119 |
| c\_Type | CAL\_DIS\_OHM |
| i16\_Low\_Cal\_Disp | 5000 |
| i16\_High\_Cal\_Disp | 25000 |
| i16\_Resolution | 100 |
| b\_excite\_req | TRUE |
| s\_Table | Ai16\_adc\_rs300,  Ai16\_ohms\_rs300 |
| i16\_Min\_Range | 0 |
| i16\_Max\_Range | 31363 |
| BIT\_2 | u8\_Chan\_State | 1 - ENABLE |
| u8\_Default\_State | 0 - DISABLE |
| i16\_Default\_Val | 0 |
| b\_Calibrate | FALSE |
| i16\_Low\_Cal\_Point | 0 |
| i16\_High\_Cal\_Point | 0 |
| c\_Type | CAL\_DIS\_V |
| i16\_Low\_Cal\_Disp | 5 |
| i16\_High\_Cal\_Disp | 50 |
| i16\_Resolution | 10 |
| b\_excite\_req | FALSE |
| s\_Table | Ai16\_builtintest\_adc,  Ai16\_builtintest\_vdc |
| i16\_Min\_Range | -247 |
| i16\_Max\_Range | 252 |
| RTD\_2 EOT2 | u8\_Chan\_State | 1 - ENABLE |
| u8\_Default\_State | 0 - DISABLE |
| i16\_Default\_Val | 0 |
| b\_Calibrate | TRUE |
| i16\_Low\_Cal\_Point | 5224 |
| i16\_High\_Cal\_Point | 26119 |
| c\_Type | CAL\_DIS\_OHM |
| i16\_Low\_Cal\_Disp | 5000 |
| i16\_High\_Cal\_Disp | 25000 |
| i16\_Resolution | 100 |
| b\_excite\_req | TRUE |
| s\_Table | Ai16\_adc\_rs300,  Ai16\_ohms\_rs300 |
| i16\_Min\_Range | 0 |
| i16\_Max\_Range | 31363 |
| RTD\_3 SPARE | u8\_Chan\_State | 0 - DISABLE |
| u8\_Default\_State | 0 - DISABLE |
| i16\_Default\_Val | 0 |
| b\_Calibrate | FALSE |
| i16\_Low\_Cal\_Point | 0 |
| i16\_High\_Cal\_Point | 0 |
| c\_Type | CAL\_DIS\_V |
| i16\_Low\_Cal\_Disp | 0 |
| i16\_High\_Cal\_Disp | 0 |
| i16\_Resolution | 0 |
| b\_excite\_req | FALSE |
| s\_Table | Ai16\_adc\_spare,  Ai16\_spare |
| i16\_Min\_Range | 0 |
| i16\_Max\_Range | 0 |
| RTD\_4 SPARE | u8\_Chan\_State | 0 - DISABLE |
| u8\_Default\_State | 0 - DISABLE |
| i16\_Default\_Val | 0 |
| b\_Calibrate | FALSE |
| i16\_Low\_Cal\_Point | 0 |
| i16\_High\_Cal\_Point | 0 |
| c\_Type | CAL\_DIS\_V |
| i16\_Low\_Cal\_Disp | 0 |
| i16\_High\_Cal\_Disp | 0 |
| i16\_Resolution | 0 |
| b\_excite\_req | FALSE |
| s\_Table | Ai16\_adc\_spare,  Ai16\_spare |
| i16\_Min\_Range | 0 |
| i16\_Max\_Range | 0 |
| RTD\_5 Spare | u8\_Chan\_State | 0 - DISABLE |
| u8\_Default\_State | 0 - DISABLE |
| i16\_Default\_Val | 0 |
| b\_Calibrate | FALSE |
| i16\_Low\_Cal\_Point | 0 |
| i16\_High\_Cal\_Point | 0 |
| c\_Type | CAL\_DIS\_V |
| i16\_Low\_Cal\_Disp | 0 |
| i16\_High\_Cal\_Disp | 0 |
| i16\_Resolution | 0 |
| b\_excite\_req | FALSE |
| s\_Table | Ai16\_adc\_spare,  Ai16\_spare |
| i16\_Min\_Range | 0 |
| i16\_Max\_Range | 0 |
| RTD\_6 TOT1\_R RS1000 | u8\_Chan\_State | 1 - ENABLE |
| u8\_Default\_State | 0 - DISABLE |
| i16\_Default\_Val | 0 |
| b\_Calibrate | TRUE |
| i16\_Low\_Cal\_Point | 2823 |
| i16\_High\_Cal\_Point | 28236 |
| c\_Type | CAL\_DIS\_OHM |
| i16\_Low\_Cal\_Disp | 1000 |
| i16\_High\_Cal\_Disp | 10000 |
| i16\_Resolution | 10 |
| b\_excite\_req | TRUE |
| s\_Table | Ai16\_adc\_rs1000,  Ai16\_ohms\_rs1000 |
| i16\_Min\_Range | 0 |
| i16\_Max\_Range | 11605 |
| RTD\_7 TOT2\_R RS1000 | u8\_Chan\_State | 1 - ENABLE |
| u8\_Default\_State | 0 - DISABLE |
| i16\_Default\_Val | 0 |
| b\_Calibrate | TRUE |
| i16\_Low\_Cal\_Point | 2823 |
| i16\_High\_Cal\_Point | 28236 |
| c\_Type | CAL\_DIS\_OHM |
| i16\_Low\_Cal\_Disp | 1000 |
| i16\_High\_Cal\_Disp | 10000 |
| i16\_Resolution | 10 |
| b\_excite\_req | TRUE |
| s\_Table | Ai16\_adc\_rs1000,  Ai16\_ohms\_rs1000 |
| i16\_Min\_Range | 0 |
| i16\_Max\_Range | 11605 |
| RTD\_8 Spare | u8\_Chan\_State | 0 - DISABLE |
| u8\_Default\_State | 0 - DISABLE |
| i16\_Default\_Val | 0 |
| b\_Calibrate | FALSE |
| i16\_Low\_Cal\_Point | 0 |
| i16\_High\_Cal\_Point | 0 |
| c\_Type | CAL\_DIS\_V |
| i16\_Low\_Cal\_Disp | 0 |
| i16\_High\_Cal\_Disp | 0 |
| i16\_Resolution | 0 |
| b\_excite\_req | FALSE |
| s\_Table | Ai16\_adc\_spare,  Ai16\_spare |
| i16\_Min\_Range | 0 |
| i16\_Max\_Range | 0 |
| BIT\_3 | u8\_Chan\_State | 1 - ENABLE |
| u8\_Default\_State | 0 - DISABLE |
| i16\_Default\_Val | 0 |
| b\_Calibrate | FALSE |
| i16\_Low\_Cal\_Point | 0 |
| i16\_High\_Cal\_Point | 0 |
| c\_Type | CAL\_DIS\_V |
| i16\_Low\_Cal\_Disp | 5 |
| i16\_High\_Cal\_Disp | 50 |
| i16\_Resolution | 10 |
| b\_excite\_req | FALSE |
| s\_Table | Ai16\_builtintest\_adc,  Ai16\_builtintest\_vdc |
| i16\_Min\_Range | -247 |
| i16\_Max\_Range | 252 |
| RTD\_9 MGBT RS1000 | u8\_Chan\_State | 1 - ENABLE |
| u8\_Default\_State | 0 - DISABLE |
| i16\_Default\_Val | 0 |
| b\_Calibrate | TRUE |
| i16\_Low\_Cal\_Point | 2823 |
| i16\_High\_Cal\_Point | 28236 |
| c\_Type | CAL\_DIS\_OHM |
| i16\_Low\_Cal\_Disp | 1000 |
| i16\_High\_Cal\_Disp | 10000 |
| i16\_Resolution | 10 |
| b\_excite\_req | TRUE |
| s\_Table | Ai16\_adc\_rs1000,  Ai16\_ohms\_rs1000 |
| i16\_Min\_Range | 0 |
| i16\_Max\_Range | 11605 |
| HL\_1 VD12 TRQ1 | u8\_Chan\_State | 1 - ENABLE |
| u8\_Default\_State | 0 - DISABLE |
| i16\_Default\_Val | 0 |
| b\_Calibrate | TRUE |
| i16\_Low\_Cal\_Point | 1342 |
| i16\_High\_Cal\_Point | 32211 |
| c\_Type | CAL\_DIS\_V |
| i16\_Low\_Cal\_Disp | 500 |
| i16\_High\_Cal\_Disp | 12000 |
| i16\_Resolution | 1000 |
| b\_excite\_req | FALSE |
| s\_Table | Ai16\_adc\_vd12,  Ai16\_v\_vd12 |
| i16\_Min\_Range | -12207 |
| i16\_Max\_Range | 12207 |
| HL\_2 VD12 TRQ2 | u8\_Chan\_State | 1 - ENABLE |
| u8\_Default\_State | 0 - DISABLE |
| i16\_Default\_Val | 0 |
| b\_Calibrate | TRUE |
| i16\_Low\_Cal\_Point | 1342 |
| i16\_High\_Cal\_Point | 32211 |
| c\_Type | CAL\_DIS\_V |
| i16\_Low\_Cal\_Disp | 500 |
| i16\_High\_Cal\_Disp | 12000 |
| i16\_Resolution | 1000 |
| b\_excite\_req | FALSE |
| s\_Table | Ai16\_adc\_vd12,  Ai16\_v\_vd12 |
| i16\_Min\_Range | -12207 |
| i16\_Max\_Range | 12207 |
| HL\_RMS\_01 EOP1 REF | u8\_Chan\_State | 1 - ENABLE |
| u8\_Default\_State | 0 - DISABLE |
| i16\_Default\_Val | 0 |
| b\_Calibrate | FALSE |
| i16\_Low\_Cal\_Point | 1342 |
| i16\_High\_Cal\_Point | 32211 |
| c\_Type | CAL\_DIS\_V |
| i16\_Low\_Cal\_Disp | 500 |
| i16\_High\_Cal\_Disp | 12000 |
| i16\_Resolution | 1000 |
| b\_excite\_req | FALSE |
| s\_Table | Ai16\_adc\_vd12,  Ai16\_v\_vd12 |
| i16\_Min\_Range | -12207 |
| i16\_Max\_Range | 12207 |
| HL\_RMS\_02 SPARE | u8\_Chan\_State | 0 - DISABLE |
| u8\_Default\_State | 0 - DISABLE |
| i16\_Default\_Val | 0 |
| b\_Calibrate | FALSE |
| i16\_Low\_Cal\_Point | 0 |
| i16\_High\_Cal\_Point | 0 |
| c\_Type | CAL\_DIS\_V |
| i16\_Low\_Cal\_Disp | 0 |
| i16\_High\_Cal\_Disp | 0 |
| i16\_Resolution | 0 |
| b\_excite\_req | FALSE |
| s\_Table | Ai16\_adc\_spare,  Ai16\_spare |
| i16\_Min\_Range | 0 |
| i16\_Max\_Range | 0 |
| HL\_RMS\_03 EOP2 REF | u8\_Chan\_State | 1 - ENABLE |
| u8\_Default\_State | 0 - DISABLE |
| i16\_Default\_Val | 0 |
| b\_Calibrate | FALSE |
| i16\_Low\_Cal\_Point | 1342 |
| i16\_High\_Cal\_Point | 32211 |
| c\_Type | CAL\_DIS\_V |
| i16\_Low\_Cal\_Disp | 500 |
| i16\_High\_Cal\_Disp | 12000 |
| i16\_Resolution | 1000 |
| b\_excite\_req | FALSE |
| s\_Table | Ai16\_adc\_vd12,  Ai16\_v\_vd12 |
| i16\_Min\_Range | -12207 |
| i16\_Max\_Range | 12207 |
| HL\_RMS\_04 SPARE | u8\_Chan\_State | 0 - DISABLE |
| u8\_Default\_State | 0 - DISABLE |
| i16\_Default\_Val | 0 |
| b\_Calibrate | FALSE |
| i16\_Low\_Cal\_Point | 0 |
| i16\_High\_Cal\_Point | 0 |
| c\_Type | CAL\_DIS\_V |
| i16\_Low\_Cal\_Disp | 0 |
| i16\_High\_Cal\_Disp | 0 |
| i16\_Resolution | 0 |
| b\_excite\_req | FALSE |
| s\_Table | Ai16\_adc\_spare,  Ai16\_spare |
| i16\_Min\_Range | 0 |
| i16\_Max\_Range | 0 |
| BIT\_4 | u8\_Chan\_State | 1 - ENABLE |
| u8\_Default\_State | 0 - DISABLE |
| i16\_Default\_Val | 0 |
| b\_Calibrate | FALSE |
| i16\_Low\_Cal\_Point | 0 |
| i16\_High\_Cal\_Point | 0 |
| c\_Type | CAL\_DIS\_V |
| i16\_Low\_Cal\_Disp | 5 |
| i16\_High\_Cal\_Disp | 50 |
| i16\_Resolution | 10 |
| b\_excite\_req | FALSE |
| s\_Table | Ai16\_builtintest\_adc, Ai16\_builtintest\_vdc |
| i16\_Min\_Range | -247 |
| i16\_Max\_Range | 252 |
| HL\_RMS\_05 FP1 REF | u8\_Chan\_State | 1 - ENABLE |
| u8\_Default\_State | 0 - DISABLE |
| i16\_Default\_Val | 0 |
| b\_Calibrate | FALSE |
| i16\_Low\_Cal\_Point | 1342 |
| i16\_High\_Cal\_Point | 32211 |
| c\_Type | CAL\_DIS\_V |
| i16\_Low\_Cal\_Disp | 500 |
| i16\_High\_Cal\_Disp | 12000 |
| i16\_Resolution | 1000 |
| b\_excite\_req | FALSE |
| s\_Table | Ai16\_adc\_vd12,  Ai16\_v\_vd12 |
| i16\_Min\_Range | -12207 |
| i16\_Max\_Range | 12207 |
| HL\_RMS\_06 FP1 | u8\_Chan\_State | 1 - ENABLE |
| u8\_Default\_State | 0 - DISABLE |
| i16\_Default\_Val | 0 |
| b\_Calibrate | TRUE |
| i16\_Low\_Cal\_Point | 1342 |
| i16\_High\_Cal\_Point | 32211 |
| c\_Type | CAL\_DIS\_V |
| i16\_Low\_Cal\_Disp | 500 |
| i16\_High\_Cal\_Disp | 12000 |
| i16\_Resolution | 1000 |
| b\_excite\_req | FALSE |
| s\_Table | Ai16\_adc\_vd12,  Ai16\_v\_vd12 |
| i16\_Min\_Range | -12207 |
| i16\_Max\_Range | 12207 |
| HL\_RMS\_07 FP2 REF | u8\_Chan\_State | 1 - ENABLE |
| u8\_Default\_State | 0 - DISABLE |
| i16\_Default\_Val | 0 |
| b\_Calibrate | FALSE |
| i16\_Low\_Cal\_Point | 1342 |
| i16\_High\_Cal\_Point | 32211 |
| c\_Type | CAL\_DIS\_V |
| i16\_Low\_Cal\_Disp | 500 |
| i16\_High\_Cal\_Disp | 12000 |
| i16\_Resolution | 1000 |
| b\_excite\_req | FALSE |
| s\_Table | Ai16\_adc\_vd12,  Ai16\_v\_vd12 |
| i16\_Min\_Range | -12207 |
| i16\_Max\_Range | 12207 |
| HL\_RMS\_08 FP2 | u8\_Chan\_State | 1 - ENABLE |
| u8\_Default\_State | 0 - DISABLE |
| i16\_Default\_Val | 0 |
| b\_Calibrate | TRUE |
| i16\_Low\_Cal\_Point | 1342 |
| i16\_High\_Cal\_Point | 32211 |
| c\_Type | CAL\_DIS\_V |
| i16\_Low\_Cal\_Disp | 500 |
| i16\_High\_Cal\_Disp | 12000 |
| i16\_Resolution | 1000 |
| b\_excite\_req | FALSE |
| s\_Table | Ai16\_adc\_vd12,  Ai16\_v\_vd12 |
| i16\_Min\_Range | -12207 |
| i16\_Max\_Range | 12207 |
| HL\_RMS\_09 LHP REF | u8\_Chan\_State | 1 - ENABLE |
| u8\_Default\_State | 0 - DISABLE |
| i16\_Default\_Val | 0 |
| b\_Calibrate | FALSE |
| i16\_Low\_Cal\_Point | 1342 |
| i16\_High\_Cal\_Point | 32211 |
| c\_Type | CAL\_DIS\_V |
| i16\_Low\_Cal\_Disp | 500 |
| i16\_High\_Cal\_Disp | 12000 |
| i16\_Resolution | 1000 |
| b\_excite\_req | FALSE |
| s\_Table | Ai16\_adc\_vd12,  Ai16\_v\_vd12 |
| i16\_Min\_Range | -12207 |
| i16\_Max\_Range | 12207 |
| HL\_RMS\_10 LHP | u8\_Chan\_State | 1 - ENABLE |
| u8\_Default\_State | 0 - DISABLE |
| i16\_Default\_Val | 0 |
| b\_Calibrate | TRUE |
| i16\_Low\_Cal\_Point | 1342 |
| i16\_High\_Cal\_Point | 32211 |
| c\_Type | CAL\_DIS\_V |
| i16\_Low\_Cal\_Disp | 500 |
| i16\_High\_Cal\_Disp | 12000 |
| i16\_Resolution | 1000 |
| b\_excite\_req | FALSE |
| s\_Table | Ai16\_adc\_vd12,  Ai16\_v\_vd12 |
| i16\_Min\_Range | -12207 |
| i16\_Max\_Range | 12207 |
| HL\_RMS\_11 RHP REF | u8\_Chan\_State | 1 - ENABLE |
| u8\_Default\_State | 0 - DISABLE |
| i16\_Default\_Val | 0 |
| b\_Calibrate | FALSE |
| i16\_Low\_Cal\_Point | 1342 |
| i16\_High\_Cal\_Point | 32211 |
| c\_Type | CAL\_DIS\_V |
| i16\_Low\_Cal\_Disp | 500 |
| i16\_High\_Cal\_Disp | 12000 |
| i16\_Resolution | 1000 |
| b\_excite\_req | FALSE |
| s\_Table | Ai16\_adc\_vd12,  Ai16\_v\_vd12 |
| i16\_Min\_Range | -12207 |
| i16\_Max\_Range | 12207 |
| BIT\_5 | u8\_Chan\_State | 1 - ENABLE |
| u8\_Default\_State | 0 - DISABLE |
| i16\_Default\_Val | 0 |
| b\_Calibrate | FALSE |
| i16\_Low\_Cal\_Point | 0 |
| i16\_High\_Cal\_Point | 0 |
| c\_Type | CAL\_DIS\_V |
| i16\_Low\_Cal\_Disp | 5 |
| i16\_High\_Cal\_Disp | 50 |
| i16\_Resolution | 10 |
| b\_excite\_req | FALSE |
| s\_Table | Ai16\_builtintest\_adc, Ai16\_builtintest\_vdc |
| i16\_Min\_Range | -247 |
| i16\_Max\_Range | 252 |
| HL\_RMS\_12 RHP | u8\_Chan\_State | 1 - ENABLE |
| u8\_Default\_State | 0 - DISABLE |
| i16\_Default\_Val | 0 |
| b\_Calibrate | TRUE |
| i16\_Low\_Cal\_Point | 1342 |
| i16\_High\_Cal\_Point | 32211 |
| c\_Type | CAL\_DIS\_V |
| i16\_Low\_Cal\_Disp | 500 |
| i16\_High\_Cal\_Disp | 12000 |
| i16\_Resolution | 1000 |
| b\_excite\_req | FALSE |
| s\_Table | Ai16\_adc\_vd12,  Ai16\_v\_vd12 |
| i16\_Min\_Range | -12207 |
| i16\_Max\_Range | 12207 |
| HL\_RMS\_13 MGBOP REF | u8\_Chan\_State | 1 - ENABLE |
| u8\_Default\_State | 0 - DISABLE |
| i16\_Default\_Val | 0 |
| b\_Calibrate | FALSE |
| i16\_Low\_Cal\_Point | 1342 |
| i16\_High\_Cal\_Point | 32211 |
| c\_Type | CAL\_DIS\_V |
| i16\_Low\_Cal\_Disp | 500 |
| i16\_High\_Cal\_Disp | 12000 |
| i16\_Resolution | 1000 |
| b\_excite\_req | FALSE |
| s\_Table | Ai16\_adc\_vd12,  Ai16\_v\_vd12 |
| i16\_Min\_Range | -12207 |
| i16\_Max\_Range | 12207 |
| HL\_RMS\_14 MGBOP | u8\_Chan\_State | 1 - ENABLE |
| u8\_Default\_State | 0 - DISABLE |
| i16\_Default\_Val | 0 |
| b\_Calibrate | TRUE |
| i16\_Low\_Cal\_Point | 1342 |
| i16\_High\_Cal\_Point | 32211 |
| c\_Type | CAL\_DIS\_V |
| i16\_Low\_Cal\_Disp | 500 |
| i16\_High\_Cal\_Disp | 12000 |
| i16\_Resolution | 1000 |
| b\_excite\_req | FALSE |
| s\_Table | Ai16\_adc\_vd12,  Ai16\_v\_vd12 |
| i16\_Min\_Range | -12207 |
| i16\_Max\_Range | 12207 |
| HL\_RMS\_15 SPARE | u8\_Chan\_State | 0 - DISABLE |
| u8\_Default\_State | 0 - DISABLE |
| i16\_Default\_Val | 0 |
| b\_Calibrate | FALSE |
| i16\_Low\_Cal\_Point | 0 |
| i16\_High\_Cal\_Point | 0 |
| c\_Type | CAL\_DIS\_V |
| i16\_Low\_Cal\_Disp | 0 |
| i16\_High\_Cal\_Disp | 0 |
| i16\_Resolution | 0 |
| b\_excite\_req | FALSE |
| s\_Table | Ai16\_adc\_spare,  Ai16\_spare |
| i16\_Min\_Range | 0 |
| i16\_Max\_Range | 0 |
| HL\_RMS\_16 SPARE | u8\_Chan\_State | 0 - DISABLE |
| u8\_Default\_State | 0 - DISABLE |
| i16\_Default\_Val | 0 |
| b\_Calibrate | FALSE |
| i16\_Low\_Cal\_Point | 0 |
| i16\_High\_Cal\_Point | 0 |
| c\_Type | CAL\_DIS\_V |
| i16\_Low\_Cal\_Disp | 100 |
| i16\_High\_Cal\_Disp | 3500 |
| i16\_Resolution | 100 |
| b\_excite\_req | FALSE |
| s\_Table | Ai16\_adc\_spare,  Ai16\_spare |
| i16\_Min\_Range | 0 |
| i16\_Max\_Range | 0 |
| HL\_RMS\_17 SPARE | u8\_Chan\_State | 0 - DISABLE |
| u8\_Default\_State | 0 - DISABLE |
| i16\_Default\_Val | 0 |
| b\_Calibrate | FALSE |
| i16\_Low\_Cal\_Point | 0 |
| i16\_High\_Cal\_Point | 0 |
| c\_Type | CAL\_DIS\_V |
| i16\_Low\_Cal\_Disp | 0 |
| i16\_High\_Cal\_Disp | 0 |
| i16\_Resolution | 0 |
| b\_excite\_req | FALSE |
| s\_Table | Ai16\_adc\_spare,  Ai16\_spare |
| i16\_Min\_Range | 0 |
| i16\_Max\_Range | 0 |
| HL\_RMS\_18 SPARE | u8\_Chan\_State | 0 - DISABLE |
| u8\_Default\_State | 0 - DISABLE |
| i16\_Default\_Val | 0 |
| b\_Calibrate | FALSE |
| i16\_Low\_Cal\_Point | 0 |
| i16\_High\_Cal\_Point | 0 |
| c\_Type | CAL\_DIS\_V |
| i16\_Low\_Cal\_Disp | 0 |
| i16\_High\_Cal\_Disp | 0 |
| i16\_Resolution | 0 |
| b\_excite\_req | FALSE |
| s\_Table | Ai16\_adc\_spare,  Ai16\_spare |
| i16\_Min\_Range | 0 |
| i16\_Max\_Range | 0 |
| Tach 1 | u8\_Chan\_State | 1 - ENABLE |
| u8\_Default\_State | 1 - ENABLE |
| i16\_Default\_Val | 0 |
| i16\_Reference\_Frequency | 7785.0f |
| i16\_Scaled\_Reading | 1000 |
| i16\_Min\_Range | 0 |
| i16\_Max\_Range | 1250 |
| i16\_Resolution | 1 |
| Tach 2 | u8\_Chan\_State | 1 - ENABLE |
| u8\_Default\_State | 1 - ENABLE |
| i16\_Default\_Val | 0 |
| i16\_Reference\_Frequency | 7785.0f |
| i16\_Scaled\_Reading | 1000 |
| i16\_Min\_Range | 0 |
| i16\_Max\_Range | 1250 |
| i16\_Resolution | 1 |
| Tach 3 | u8\_Chan\_State | 1 - ENABLE |
| u8\_Default\_State | 1 - ENABLE |
| i16\_Default\_Val | 0 |
| i16\_Reference\_Frequency | 70.0f |
| i16\_Scaled\_Reading | 1000 |
| i16\_Min\_Range | 0 |
| i16\_Max\_Range | 1250 |
| i16\_Resolution | 10 |
| Tach 4 | u8\_Chan\_State | 1 - ENABLE |
| u8\_Default\_State | 1 - ENABLE |
| i16\_Default\_Val | 0 |
| i16\_Reference\_Frequency | 70.0f |
| i16\_Scaled\_Reading | 1000 |
| i16\_Min\_Range | 0 |
| i16\_Max\_Range | 1250 |
| i16\_Resolution | 10 |
| Tach 5 | u8\_Chan\_State | 1 - ENABLE |
| u8\_Default\_State | 1 - ENABLE |
| i16\_Default\_Val | 0 |
| i16\_Reference\_Frequency | 70.0f |
| i16\_Scaled\_Reading | 1000 |
| i16\_Min\_Range | 0 |
| i16\_Max\_Range | 1250 |
| i16\_Resolution | 10 |
| Tach 6 | u8\_Chan\_State | 1 - ENABLE |
| u8\_Default\_State | 1 - ENABLE |
| i16\_Default\_Val | 0 |
| i16\_Reference\_Frequency | 70.0f |
| i16\_Scaled\_Reading | 1000 |
| i16\_Min\_Range | 0 |
| i16\_Max\_Range | 1250 |
| i16\_Resolution | 10 |
| PWM 1 | u8\_Chan\_State | 1 - ENABLE |
| u8\_Default\_State | 1 - ENABLE |
| i16\_Default\_Val | 0 |
| i16\_Reference\_Frequency | 100.0f |
| i16\_Scaled\_Reading | 1000 |
| i16\_Min\_Range | 0 |
| i16\_Max\_Range | 1250 |
| i16\_Resolution | 10 |
| PWM 2 | u8\_Chan\_State | 1 - ENABLE |
| u8\_Default\_State | 1 - ENABLE |
| i16\_Default\_Val | 0 |
| i16\_Reference\_Frequency | 100.0f |
| i16\_Scaled\_Reading | 1000 |
| i16\_Min\_Range | 0 |
| i16\_Max\_Range | 1250 |
| i16\_Resolution | 10 |
| PWM 3 | u8\_Chan\_State | 1 - ENABLE |
| u8\_Default\_State | 1 - ENABLE |
| i16\_Default\_Val | 0 |
| i16\_Reference\_Frequency | 100.0f |
| i16\_Scaled\_Reading | 1000 |
| i16\_Min\_Range | 0 |
| i16\_Max\_Range | 1250 |
| i16\_Resolution | 10 |
| PWM 4 | u8\_Chan\_State | 1 - ENABLE |
| u8\_Default\_State | 1 - ENABLE |
| i16\_Default\_Val | 0 |
| i16\_Reference\_Frequency | 100.0f |
| i16\_Scaled\_Reading | 1000 |
| i16\_Min\_Range | 0 |
| i16\_Max\_Range | 1250 |
| i16\_Resolution | 10 |

#### 12.1.3.3 dauanasensor-LLR-003

Requirement ID: H398-LLD-ANA-FNC-1833

The Cold Junction signal configuration section shall have the format specified in Table: Cold Junction signal configuration format.

Table: Cold Junction signal configuration format

Table : Cold Junction signal configuration format

|  |  |  |
| --- | --- | --- |
| **Channel** | **Parameter** | **Configured Value** |
| Cold Junction signal | u8\_Chan\_State | 1 - ENABLE |
| u8\_Default\_State | 0 - DISABLE |
| i16\_Default\_Val | 0 |
| b\_Calibrate | FALSE |
| i16\_Low\_Cal\_Point | 0 |
| i16\_High\_Cal\_Point | 0 |
| c\_Type | CAL\_DIS\_TEMPC |
| i16\_Low\_Cal\_Disp | 1000 |
| i16\_High\_Cal\_Disp | 200 |
| i16\_Resolution | 1 |
| b\_excite\_req | FALSE |
| s\_Table | Ai16\_coldjunction\_adc,  Ai16\_coldjunction\_c |
| i16\_Min\_Range | -732 |
| i16\_Max\_Range | 1058 |

#### 12.1.3.4 dauanasensor-LLR-004

Requirement ID: H398-LLD-ANA-FNC-2448

The DAC signal configuration section shall have the format specified in Table: DAC signal configuration format.

Table: DAC signal configuration format

|  |  |  |
| --- | --- | --- |
| **Channel** | **Parameter** | **Configured Value** |
| DAC1 TRQ1 OUT | u8\_Chan\_State | 1 - ENABLE |
| u8\_Default\_State | 0 - DISABLE |
| i16\_Default\_Val | 0 |
| b\_Calibrate | TRUE |
| i16\_Low\_Cal\_Point | 33392 |
| i16\_High\_Cal\_Point | 63975 |
| c\_Type | CAL\_DIS\_V |
| i16\_Low\_Cal\_Disp | 1 |
| i16\_High\_Cal\_Disp | 50 |
| i16\_Resolution | 10 |
| b\_excite\_req | FALSE |
| s\_Table | Ai16\_dac\_adc,  Ai16\_dac\_v |
| i16\_Min\_Range | -5250 |
| i16\_Max\_Range | 5250 |
| DAC2 TRQ2 OUT | u8\_Chan\_State | 1 - ENABLE |
| u8\_Default\_State | 0 - DISABLE |
| i16\_Default\_Val | 0 |
| b\_Calibrate | TRUE |
| i16\_Low\_Cal\_Point | 33392 |
| i16\_High\_Cal\_Point | 63975 |
| c\_Type | CAL\_DIS\_V |
| i16\_Low\_Cal\_Disp | 1 |
| i16\_High\_Cal\_Disp | 50 |
| i16\_Resolution | 10 |
| b\_excite\_req | FALSE |
| s\_Table | Ai16\_dac\_adc,  Ai16\_dac\_v |
| i16\_Min\_Range | -5250 |
| i16\_Max\_Range | 5250 |
| DAC3 TRQ1+2 OUT | u8\_Chan\_State | 1 - ENABLE |
| u8\_Default\_State | 0 - DISABLE |
| i16\_Default\_Val | 0 |
| b\_Calibrate | TRUE |
| i16\_Low\_Cal\_Point | 33392 |
| i16\_High\_Cal\_Point | 63975 |
| c\_Type | CAL\_DIS\_V |
| i16\_Low\_Cal\_Disp | 1 |
| i16\_High\_Cal\_Disp | 50 |
| i16\_Resolution | 10 |
| b\_excite\_req | FALSE |
| s\_Table | Ai16\_dac\_adc,  Ai16\_dac\_v |
| i16\_Min\_Range | -5250 |
| i16\_Max\_Range | 5250 |
| DAC4 TOT1 OUT | u8\_Chan\_State | 1 - ENABLE |
| u8\_Default\_State | 0 - DISABLE |
| i16\_Default\_Val | 0 |
| b\_Calibrate | TRUE |
| i16\_Low\_Cal\_Point | 33392 |
| i16\_High\_Cal\_Point | 63975 |
| c\_Type | CAL\_DIS\_V |
| i16\_Low\_Cal\_Disp | 1 |
| i16\_High\_Cal\_Disp | 50 |
| i16\_Resolution | 10 |
| b\_excite\_req | FALSE |
| s\_Table | Ai16\_dac\_tot\_adc,  Ai16\_dac\_tot\_degc |
| i16\_Min\_Range | -5250 |
| i16\_Max\_Range | 5250 |
| DAC5 TOT2 OUT | u8\_Chan\_State | 1 - ENABLE |
| u8\_Default\_State | 0 - DISABLE |
| i16\_Default\_Val | 0 |
| b\_Calibrate | TRUE |
| i16\_Low\_Cal\_Point | 33392 |
| i16\_High\_Cal\_Point | 63975 |
| c\_Type | CAL\_DIS\_V |
| i16\_Low\_Cal\_Disp | 1 |
| i16\_High\_Cal\_Disp | 50 |
| i16\_Resolution | 10 |
| b\_excite\_req | FALSE |
| s\_Table | Ai16\_dac\_tot\_adc,  Ai16\_dac\_tot\_degc |
| i16\_Min\_Range | -5250 |
| i16\_Max\_Range | 5250 |
| DAC6 MGBT OUT | u8\_Chan\_State | 1 - ENABLE |
| u8\_Default\_State | 0 - DISABLE |
| i16\_Default\_Val | 0 |
| b\_Calibrate | TRUE |
| i16\_Low\_Cal\_Point | 33392 |
| i16\_High\_Cal\_Point | 63975 |
| c\_Type | CAL\_DIS\_V |
| i16\_Low\_Cal\_Disp | 1 |
| i16\_High\_Cal\_Disp | 50 |
| i16\_Resolution | 10 |
| b\_excite\_req | FALSE |
| s\_Table | Ai16\_dac\_mgbt\_adc,  Ai16\_dac\_mgbt\_ohms |
| i16\_Min\_Range | -5250 |
| i16\_Max\_Range | 5250 |
| DAC7 MGBOP OUT | u8\_Chan\_State | 1 - ENABLE |
| u8\_Default\_State | 0 - DISABLE |
| i16\_Default\_Val | 0 |
| b\_Calibrate | TRUE |
| i16\_Low\_Cal\_Point | 33392 |
| i16\_High\_Cal\_Point | 63975 |
| c\_Type | CAL\_DIS\_V |
| i16\_Low\_Cal\_Disp | 1 |
| i16\_High\_Cal\_Disp | 50 |
| i16\_Resolution | 10 |
| b\_excite\_req | FALSE |
| s\_Table | Ai16\_dac\_mgbop\_adc,  Ai16\_dac\_mgbop\_mv |
| i16\_Min\_Range | -5250 |
| i16\_Max\_Range | 5250 |
| DAC8 | u8\_Chan\_State | 1 - ENABLE |
| u8\_Default\_State | 0 - DISABLE |
| i16\_Default\_Val | 0 |
| b\_Calibrate | TRUE |
| i16\_Low\_Cal\_Point | 33392 |
| i16\_High\_Cal\_Point | 63975 |
| c\_Type | CAL\_DIS\_V |
| i16\_Low\_Cal\_Disp | 1 |
| i16\_High\_Cal\_Disp | 50 |
| i16\_Resolution | 10 |
| b\_excite\_req | FALSE |
| s\_Table | Ai16\_dac\_adc,  Ai16\_dac\_v |
| i16\_Min\_Range | -5250 |
| i16\_Max\_Range | 5250 |

# 13 Appendix A : Data Dictionary

The data dictionary for the Analog Module Application Software :



The constant data table for the Analog Module MCD Software.



# 14 Appendix B: Data Constants

This section contains the data constants for the Analog Module Application Software:



This section contains the data constants for the Analog Module MCD Software.

