**vCAN UDS for BCM**

**Software Design Document for Application**

**Version 1.0**

**Customer: Royal Enfield**

Contents

[1. Introduction 4](#_Toc146031498)

[1.1 Abbreviations and Acronyms: 4](#_Toc146031499)

[1.2 Reference Document 4](#_Toc146031500)

[1.3 BSW Architecture 4](#_Toc146031501)

[1.4 Memory Layout 5](#_Toc146031502)

[1.5 Supported CAN ID’s for Application 5](#_Toc146031503)

[1.6 Functions for UDS services: 6](#_Toc146031504)

[2. ASW Services 7](#_Toc146031505)

[2.1 iso14229\_serv10 7](#_Toc146031506)

[2.2 iso14229\_serv11 10](#_Toc146031507)

[2.3 iso14229\_serv27 12](#_Toc146031508)

[2.4 iso14229\_serv28 15](#_Toc146031509)

[2.5 iso14229\_serv3E 18](#_Toc146031510)

[2.6 iso14229\_serv14 19](#_Toc146031511)

[2.7 iso14229\_serv19 20](#_Toc146031512)

[2.8 iso14229\_serv22 21](#_Toc146031513)

[2.9 iso14229\_serv2E 23](#_Toc146031514)

[2.10 iso14229\_serv23 25](#_Toc146031515)

[2.11 iso14229\_serv3D 28](#_Toc146031516)

[2.12 iso14229\_serv2F 31](#_Toc146031517)

[Traceability of ASW SDD and SRS 34](#_Toc146031518)

[Table 1: Abbreviation 4](#_Toc146031472)

[Table 2: Supported CAN ID 5](#_Toc146031473)

[Table 3: Function description of 0x10 7](#_Toc146031474)

[Table 4: Supported NRC for 0x10 9](#_Toc146031475)

[Table 5: Function description of 0x11 10](#_Toc146031476)

[Table 6: Supported NRC for 0x11 11](#_Toc146031477)

[Table 7: Function description of 0x27 12](#_Toc146031478)

[Table 8: Supported NRC for 0x27 14](#_Toc146031479)

[Table 9: Function description of 0x28 15](#_Toc146031480)

[Table 10: Supported NRC for 0x28 17](#_Toc146031481)

[Table 11: Function description of 0x3E 18](#_Toc146031482)

[Table 12: Supported NRC for 0x3E 18](#_Toc146031483)

[Table 13: Function description of 0x14 19](#_Toc146031484)

[Table 14: Supported NRC for 0x14 19](#_Toc146031485)

[Table 15: Function description of 0x19 20](#_Toc146031486)

[Table 16: Supported NRC for 0x19 20](#_Toc146031487)

[Table 17: Function description of 0x22 21](#_Toc146031488)

[Table 18: Supported NRC for 0x22 22](#_Toc146031489)

[Table 19: Function description of 0x2E 23](#_Toc146031490)

[Table 20: Supported NRC for 0x2E 24](#_Toc146031491)

[Table 21: Function description of 0x23 25](#_Toc146031492)

[Table 22: Supported NRC for 0x23 27](#_Toc146031493)

[Table 23: Function description of 0x3D 28](#_Toc146031494)

[Table 24: Supported NRC for 0x3D 30](#_Toc146031495)

[Table 3: Function description of 0x2F 31](#_Toc146031496)

[Table 4: Supported NRC for 0x2F 33](#_Toc146031497)

[Figure 1: Layered Architecture of BSW 4](#_Toc144716794)

[Figure 2: Memory Mapping 5](#_Toc144716795)

[Figure 3: Interface Diagram for UDS 6](#_Toc144716796)

**Foreword**

Foreword this document has been drafted based on ISO 14229/ISO 15765 which contains the Re-programming requirements for vCAN UDS. This document also provides the details regarding the implementation of services and the testing methods to validate the implementation.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Version | **Version Description** | **Date** | **Author** | **Reviewed By** | **Approved By** |
| 1.0 | Initial Version | 19/09/2023 | Lavanya H R | Suma | Aparna |

# Introduction

This document describes the detailed programming sequence to be followed and the subsequent services in UDS to be followed during application of vCAN UDS.

## Abbreviations and Acronyms:

| **Acronyms / Definitions / Terms** | **Description** |
| --- | --- |
| ECU | Electronic Control Unit |
| UDS | Unified Diagnostic Services |
| ASW | Application Software |
| BSW | Boot Loader Software |
| UART | Universal Asynchronous Receiver/Transmitter |

Table 1: Abbreviation

## Reference Document

The services mentioned in this document are based on the ISO 14229. For details on the services, their behavior, security access levels and additional vehicle/safety conditions please refer the same.

## BSW Architecture

BSW Software Architecture consist of three layers

* Application Layer
* Configuration and Interface Layer
* Hardware Abstraction Layer

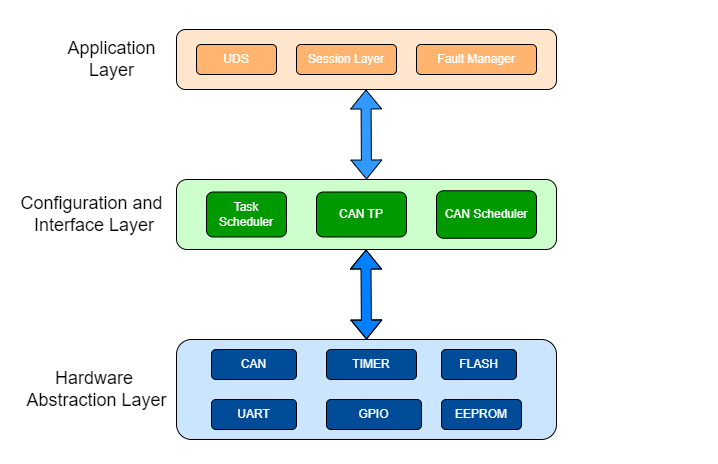


Figure 1: Layered Architecture of BSW

## Memory Layout

* Code Flash memory is of 4MB starts from **400000H** to **7FFFFFH**.
* Data Flash memory is of 128KB starts from **1000000H to 1001FFFFH**.

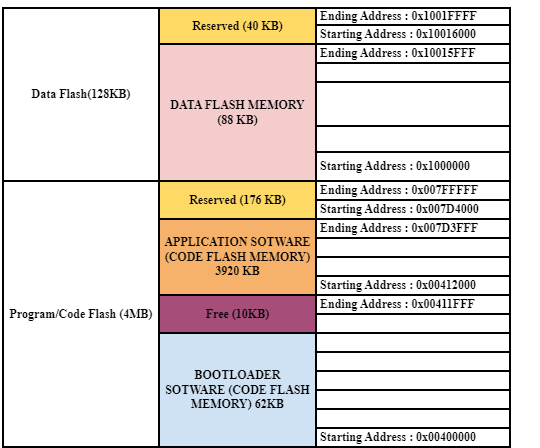


Figure : Memory Mapping

* In flash memory (4M bytes) is divided into two sections, ASW area and BSW area.
* Boot loader will start from the address 0x00400000 and it will end at 0x00411FFF.
* There will be an FW0 & FW1 Flag which will stored in external EEPROM with 1 byte each in length, and this section is used for the validation of the ASW Section.
* ASW will start after the FW1 & FW0 Flag will be true. That is from 0x00412000till 0x007FFFFF.
* FW1 Flag shall be clear when routine control request is received on the firmware.

## Supported CAN ID’s for Application

|  |  |
| --- | --- |
| **CAN Message Direction** | **ID** |
| Request ID (Physical) | 0x7DA |
| Request ID (Functional) | N/A |
| Response ID (Physical) | 0x7DB |

Table 2: Supported CAN ID

## Functions for UDS services:

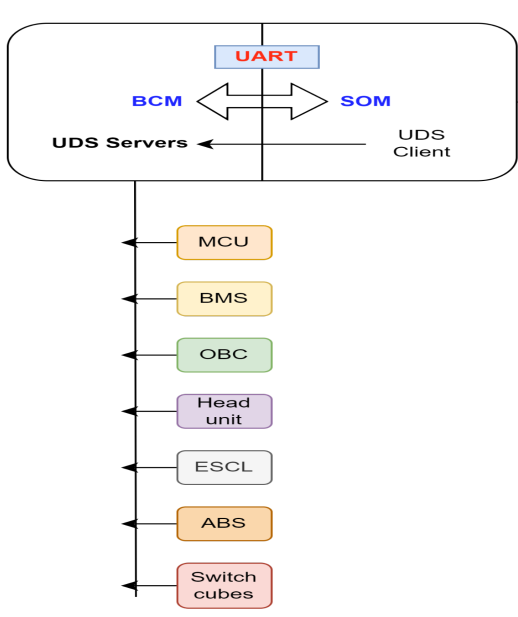


Figure 3: Interface Diagram for UDS

# ASW Services

## iso14229\_serv10

[SDD\_ASW\_10H\_001]

**iso14229\_serv10 shall be the service interpreter for service 0x10.**

| **Function** | **iso14229\_serv10** |
| --- | --- |
| Module | iso14229\_serv10.c |
| Prototype | UDS\_Serv\_resptype\_En\_t iso14229\_serv10(UDS\_Serv\_St\_t\* UDS\_Serv\_pSt); |
| Description | Service interpreter function for service 0x10. |
| Parameter | UDS\_Serv\_St\_t\* UDS\_Serv\_pSt - request data buffer form distributor function. |
| Return | UDS\_Serv\_resptype\_En\_t Serv\_resptype\_En - Returns the type of response. |
| Algorithm | The Diagnostic Session Control service is used to enable different diagnostic sessions in the server(s).A diagnostic session enables a specific set of diagnostic services and/or functionality in the server.  Sub functions supported by Diagnostic Session Control  Default Session(0x01): This diagnostic session enables the default diagnostic session in the server(s) and does not support any diagnostic application timeout handling provisions.  Programming Session(0x02): This diagnostic Session enables all diagnostic services required to support the memory programming of a server. When the client requests this service ASW start and end enable pattern (0x55AA55AA) which is programmed is erased then it will jump to Boot loader.  Extended Diagnostic Session(0x03): This diagnostic Session can be used to enable all diagnostic services required to support the adjustment of functions like "Idle Speed, CO Value, etc." in the server's memory. |
| Error handling | Check if the request contains a valid SID (0x10) |

Table : Function description of 0x10

[SDD\_ASW\_10H\_002]

|  |  |
| --- | --- |
| **Function** | **iso14229\_serv10** |
| Module | iso14229\_serv10.c |
| Prototype | void UDS\_Default\_Diag\_Init (); |
| Description | This Function is used to initialize default diagnostic session. |
| Parameter | void |
| Return | void |

[SDD\_ASW\_10H\_003]

|  |  |
| --- | --- |
| **Function** | **iso14229\_serv10** |
| Module | iso14229\_serv10.c |
| Prototype | void UDS\_Programming\_Session\_Init(); |
| Description | This Function is used to initialize Programming session. |
| Parameter | void |
| Return | void |

[SDD\_ASW\_10H\_004]

|  |  |
| --- | --- |
| **Function** | **iso14229\_serv10** |
| Module | iso14229\_serv10.c |
| Prototype | void UDS\_Extended\_Diag\_Init (); |
| Description | This Function is used to initialize Extended session. |
| Parameter | void |
| Return | void |

[SDD\_ASW\_10H\_005]

|  |  |
| --- | --- |
| **Function** | **iso14229\_serv10** |
| Module | iso14229\_serv10.c |
| Prototype | void UDS\_Serv10\_Timeout (); |
| Description | This Function is used to set the current session as Default Diagnostic Session. |
| Parameter | void |
| Return | void |

[SDD\_ASW\_10H\_006]

|  |  |
| --- | --- |
| **Function** | **iso14229\_serv10** |
| Module | iso14229\_serv10.c |
| Prototype | Uint8\_tUDS\_GetCurrentSession(); |
| Description | This Function is used to get the current Session. |
| Parameter | void |
| Return | uint8\_t |

**NRC handling:**

|  |  |  |
| --- | --- | --- |
| **Tags** | **NRC** | **Description** |
| [SDD\_ASW\_10H\_007] | 0x12 | **sub-function Not Supported:**  This NRC shall be sent if the sub-function parameter is not supported. |
| [SDD\_ASW\_10H\_008] | 0x13 | **Incorrect Message Length or Invalid Format:**  This NRC shall be sent if the length of the message is wrong. |
| [SDD\_ASW\_10H\_009] | 0x22 | **Conditions Not Correct**  This NRC shall be returned if the Security is not accessed and sub function is not supported in active session. |

Table : Supported NRC for 0x10

## iso14229\_serv11

[SDD\_ASW\_11H\_001]

**iso14229\_serv11 shall be the service interpreter for service 0x11.**

| **Function** | **iso14229\_serv11** |
| --- | --- |
| Module | iso14229\_serv11.c |
| Prototype | UDS\_Serv\_resptype\_En\_t iso14229\_serv11(UDS\_Serv\_St\_t\* UDS\_Serv\_pSt); |
| Description | Service interpreter function for service 0x11. |
| Parameter | UDS\_Serv\_St\_t\* UDS\_Serv\_pSt - request data buffer form distributor function. |
| Return | UDS\_Serv\_resptype\_En\_t Serv\_resptype\_En - Returns the type of response. |
| Algorithm | The ECUReset Control service is used to enable different Reset control in the server(s). A ECUReset enables a specific set of Reset services and/or functionality in the server.  Sub functions supported by ECUReset Control  Hard Reset(0x01): This Sub Function identifies a "hard reset" condition which simulates the power-on/start-up sequence typically performed after a server has been previously disconnected from its power supply. The performed action is implementation specific and not defined by this document.  KeyOffOn Reset(0x02): When you will do IGN off then immediately ECU’s power down will not happen, it will first store all the data into Non-volatile memory of the processor and then de-initialized all the programs and then it’ll go into power down mode without losing any data. We can say that this is nothing but proper reset of the ECU.  **Soft Reset(0x03):** In simple term the soft reset is equivalent to restart your main application program, means your stack pointer of the microcontroller will point to the address of main function |
| Error handling | Check if the request contains a valid SID (0x11) |

Table : Function description of 0x11

**NRC handling:**

|  |  |  |
| --- | --- | --- |
| **Tags** | **NRC** | **Description** |
| [SDD\_ASW\_11H\_002] | 0x12 | **sub-function Not Supported:**  This NRC shall be sent if the sub-function parameter is not supported. |
| [SDD\_ASW\_11H\_003] | 0x13 | **Incorrect Message Length or Invalid Format:**  This NRC shall be sent if the length of the message is wrong. |
| [SDD\_ASW\_11H\_004] | 0x33 | **Security Access Denied**  This NRC shall be returned if the Security is not accessed. |

Table : Supported NRC for 0x11

## iso14229\_serv27

[SDD\_ASW\_27H\_001]

**iso14229\_serv27 shall be the service interpreter for service 0x27.**

| **Function** | **iso14229\_serv27** |
| --- | --- |
| Module | iso14229\_serv27.c |
| Prototype | UDS\_Serv\_resptype\_En\_t iso14229\_serv27(UDS\_Serv\_St\_t\* UDS\_Serv\_pSt); |
| Description | Service interpreter function for service 0x27. |
| Parameter | UDS\_Serv\_St\_t\* UDS\_Serv\_pSt - request data buffer form distributor function. |
| Return | UDS\_Serv\_resptype\_En\_t Serv\_resptype\_En - Returns the type of response. |
| Algorithm | The Security Access Service is used to modify the ECU data stored in memory, before that the user first has to grant access through this service. The purpose of this service is to supply a method to access information and/or diagnostic services that have restricted access for security, emissions, or safety reasons.  Seed Request – (0x01)  The client sends a request for a “seed” to the server that it wants to unlock.  The server replies by sending the “seed” back to the client.  Send Key – (0x02)  The client then generates a “key” based on the “seed” and sends the key to the server.  If the client-generated the “key” with the correct algorithm the server will respond that the “key” was valid and that it will unlock itself. |
| Error handling | Check if the request contains a valid SID (0x27) |

Table : Function description of 0x27

[SDD\_ASW\_27H\_002]

| **Function** | **iso14229\_serv27** |
| --- | --- |
| Module | iso14229\_serv27.c |
| Prototype | void Generate\_Seed\_Frame (uint8\_t \*databuff\_pu8); |
| Description | This function used to generate the Seed. |
| Parameter | uint8\_t \*databuff\_pu8- data buffer used to store the seed value. |
| Return | void. |

[SDD\_ASW\_27H\_003]

| **Function** | **iso14229\_serv27** |
| --- | --- |
| Module | iso14229\_serv27.c |
| Prototype | bool GenerateKeyAndVerify (uint8\_t \*databuff\_pu8); |
| Description | This function used to Generate and Verify the key value. |
| Parameter | uint8\_t \*databuff\_pu8- data buffer used to store the client enter key value. |
| Return | bool |

[SDD\_ASW\_27H\_004]

| **Function** | **iso14229\_serv27** |
| --- | --- |
| Module | iso14229\_serv27.c |
| Prototype | void Generate\_NullSeed (uint8\_t \*databuff\_pu8); |
| Description | This function used to generate null Seed. |
| Parameter | uint8\_t \*databuff\_pu8- data buffer used to store the Null Seed  value |
| Return | void. |

[SDD\_ASW\_27H\_005]

| **Function** | **iso14229\_serv27** |
| --- | --- |
| Module | iso14229\_serv27.c |
| Prototype | void UDS\_Serv27\_timeout (); |
| Description | This function is used to clear the security flag. |
| Parameter | null |
| Return | void. |

**NRC handling:**

|  |  |  |
| --- | --- | --- |
| **Tags** | **NRC** | **Description** |
| [SDD\_ASW\_27H\_006] | 0x12 | **sub-function Not Supported:**  This NRC shall be sent if the sub-function parameter is not supported. |
| [SDD\_ASW\_27H\_007] | 0x13 | **Incorrect Message Length or Invalid Format:**  This NRC shall be sent if the length of the message is wrong. |
| [SDD\_ASW\_27H\_008] | 0x24 | **Request Sequence Error:**  This NRC shall be returned if Without getting Seed enter into send key sub function. |
| [SDD\_ASW\_27H\_009] | 0x35 | **Invalid Key:**  This NRC shall be returned if enters Invalid key |
| [SDD\_ASW\_27H\_010] | 0x36 | **Exceeded Number of Attempt:**  This NRC shall be returned if Client enters invalid key for 3 times. |
| [SDD\_ASW\_27H\_011] | 0x37 | **Required time delay not expired:**  This NRC shall be returned if enter invalid key for 3 times then timeout function will activate for 20 seconds. |

Table : Supported NRC for 0x27

## iso14229\_serv28

[SDD\_ASW\_28H\_001]

**iso14229\_serv28 shall be the service interpreter for service 0x28.**

| **Function** | **iso14229\_serv28** |
| --- | --- |
| Module | iso14229\_serv28.c |
| Prototype | UDS\_Serv\_resptype\_En\_t iso14229\_serv28(UDS\_Serv\_St\_t\* UDS\_Serv\_pSt); |
| Description | Service interpreter function for service 0x28. |
| Parameter | UDS\_Serv\_St\_t\* UDS\_Serv\_pSt - request data buffer from distributor function. |
| Return | UDS\_Serv\_resptype\_En\_t Serv\_resptype\_En - Returns the type of response. |
| Algorithm | The purpose of this service is to switch on/off the transmission and/or the reception of certain messages of (a) server(s).  Sub functions supported by Communication Control service:  **Enable Rx and Tx(0x00):** This value indicates that the reception and transmission of messages shall be enabled for the specified communication Type.  **Enable Rx and Disable Tx(0x01):** This value indicates that the reception of messages shall be enabled and the transmission shall be disabled for the specified communication Type.  **Disable Rx and Enable Tx(0x02):** This value indicates that the reception of messages shall be disabled and the transmission shall be enabled for the specified communication Type.  **Disable Rx and Tx(0x03):** This value indicates that the reception and transmission of messages shall be disabled for the specified communication Type. |
| Error handling | Check if the request contains a valid SID (0x28) |

Table : Function description of 0x28

[SDD\_ASW\_28H\_002]

| **Function** | **iso14229\_serv28** |
| --- | --- |
| Module | iso14229\_serv28.c |
| Prototype | void Comm\_Ctrl (uint8\_t Com\_Type\_u8, bool Rx\_Status\_b, bool Tx\_Status\_b) |
| Description | Enable and/or disable the Tx and Tx messages of the communication type. |
| Parameter | uint8\_t Com\_Type\_u8: request Communication type.  bool Rx\_Status\_b: to enable(true) or Disable(false) the reception of the message.  bool Tx\_Status\_b: to Enable(true) or Disable(false) the transmission of the message. |
| Return | void. |

[SDD\_ASW\_28H\_003]

| **Function** | **iso14229\_serv28** |
| --- | --- |
| Module | Comm\_cntrl\_adapt.c |
| Prototype | void NormalCommMsg\_Adapt (bool Rx\_Status, bool Tx\_Status) |
| Description | Enable and/or disable the Tx and Tx messages of normal communication type. |
| Parameter | bool Rx\_Status: to enable(true) or disable(false) the reception of the message.  bool Tx\_Status: to enable(true) or disable(false) the transmission of the message. |
| Return | void. |

[SDD\_ASW\_28H\_004]

| **Function** | **iso14229\_serv28** |
| --- | --- |
| Module | Comm\_cntrl\_adapt.c |
| Prototype | void NWMngmntCommMsg\_Adapt (bool Rx\_Status, bool Tx\_Status) |
| Description | Enable and/or disable the Tx and Tx messages of network communication type. |
| Parameter | bool Rx\_Status: to enable(true) or disable(false) the reception of the message.  bool Tx\_Status: to enable(true) or disable(false) the transmission of the message. |
| Return | void. |

[SDD\_ASW\_28H\_005]

| **Function** | **iso14229\_serv28** |
| --- | --- |
| Module | Comm\_cntrl\_adapt.c |
| Prototype | void NWMngmnt \_NormalCommMsg\_Adapt (bool Rx\_Status, bool Tx\_Status) |
| Description | Enable and/or disable the Tx and Tx messages of normal and network communication type. |
| Parameter | bool Rx\_Status: to enable(true) or disable(false) the reception of the message.  bool Tx\_Status: to enable(true) or disable(false) the transmission of the message. |
| Return | void. |

**NRC handling:**

|  |  |  |
| --- | --- | --- |
| **Tags** | **NRC** | **Description** |
| [SDD\_ASW\_28H\_006] | 0x12 | **sub-function Not Supported:**  This NRC shall be sent if the Sub Function parameter is not supported. |
| [SDD\_ASW\_28H\_007] | 0x13 | **Incorrect Message Length or Invalid Format:**  This NRC shall be sent if the length of the message is wrong. |
| [SDD\_ASW\_28H\_008] | 0x31 | **Request Out of Range:**  The server shall use this response code, if it detects an error in the communication Type or node Identification Number parameter. |

Table : Supported NRC for 0x28

## iso14229\_serv3E

[SDD\_ASW\_3EH\_001]

**iso14229\_serv3E shall be the service interpreter for service 0x3E.**

| **Function** | **iso14229\_serv3E** |
| --- | --- |
| Module | iso14229\_serv3E.c |
| Prototype | UDS\_Serv\_resptype\_En\_t iso14229\_serv3E (UDS\_Serv\_St\_t\* UDS\_Serv\_pSt); |
| Description | Service interpreter function for service 0x3E. |
| Parameter | UDS\_Serv\_St\_t\* UDS\_Serv\_pSt - request data buffer form distributor function. |
| Return | UDS\_Serv\_resptype\_En\_t Serv\_resptype\_En - Returns the type of response. |
| Algorithm | This service is used to indicate to a server (or servers) that a client is still connected to the vehicle and that certain diagnostic services and/or communication that have been previously activated are to remain active.  Zero Sub Function – (0x00)  This parameter value is used to indicate that no Sub Function value beside the suppressPosRspMsgIndicationBit is supported by this service. |
| Error handling | Check if the request contains a valid SID (0x3E) |

Table : Function description of 0x3E

**NRC handling:**

[SDD\_ASW\_3EH\_002]

|  |  |  |
| --- | --- | --- |
| **Tags** | **NRC** | **Description** |
| [SDD\_ASW\_3EH\_003] | 0x12 | **sub-function Not Supported:**  This NRC shall be sent if the sub-function parameter is not supported |
| [SDD\_ASW\_3EH\_004] | 0x13 | **Incorrect Message Length or Invalid Format:**  This NRC shall be sent if the length of the message is wrong. |

Table : Supported NRC for 0x3E

## iso14229\_serv14

[SDD\_ASW\_14H\_001]

**iso14229\_serv14 shall be the service interpreter for service 0x14.**

| **Function** | **iso14229\_serv14** |
| --- | --- |
| Module | iso14229\_serv14.c |
| Prototype | UDS\_Serv\_resptype\_En\_t iso14229\_serv14 (UDS\_Serv\_St\_t\* UDS\_Serv\_pSt); |
| Description | Service interpreter function for service 0x14 |
| Parameter | UDS\_Serv\_St\_t\* UDS\_Serv\_pSt - request data buffer form distributor function. |
| Return | UDS\_Serv\_resptype\_En\_t Serv\_resptype\_En - Returns the type of response. |
| Algorithm | The Clear Diagnostic Information service is used by the client to clear diagnostic information in one or multiple servers’ memory.  The server shall send a positive response when the ClearDiagnosticInformation service is completelyprocessed. The server shall send a positive response even if no DTCs are stored. |
| Error handling | Check if the request contains a valid SID (0x14) |

Table : Function description of 0x14

**NRC handling:**

|  |  |  |
| --- | --- | --- |
| **Tags** | **NRC** | **Description** |
| [SDD\_ASW\_14H\_002] | 0x13 | **IncorrectMessageLengthOrInvalidFormat:**  This NRC shall be sent if the length of the message is wrong. |
| [SDD\_ASW\_14H\_003] | 0x31 | **RequestOutOfRange**  This NRC shall be returned if the specified groupOfDTC parameter is not supported. |
| [SDD\_ASW\_14H\_004] | 0x72 | **GeneralProgrammingFailure**  This NRC shall be returned if the server detects an error when writing to a memory location. |

Table : Supported NRC for 0x14

## iso14229\_serv19

[SDD\_ASW\_19H\_001]

**iso14229\_serv14 shall be the service interpreter for service 0x19.**

| **Function** | **iso14229\_serv19** |
| --- | --- |
| Module | iso14229\_serv19.c |
| Prototype | UDS\_Serv\_resptype\_En\_t iso14229\_serv19(UDS\_Serv\_St\_t\* UDS\_Serv\_pSt); |
| Description | Service interpreter function for service 19 |
| Parameter | UDS\_Serv\_St\_t\* UDS\_Serv\_pSt - request data buffer form distributor function. |
| Return | UDS\_Serv\_resptype\_En\_t Serv\_resptype\_En - Returns the type of response. |
| Algorithm | This service allows a client to read the status of server resident Diagnostic Trouble Code (DTC) information from any server, or group of servers within a vehicle. |
| Error handling | Check if the request contains a valid SID (0x19) |

Table : Function description of 0x19

**NRC handling:**

|  |  |  |
| --- | --- | --- |
| **Tags** | **NRC** | **Description** |
| [SDD\_ASW\_19H\_002] | 0x12 | **sub-functionNotSupported:**  This NRC shall be sent if the sub-function parameter is not supported. |
| [SDD\_ASW\_19H\_003] | 0x13 | **IncorrectMessageLengthOrInvalidFormat**:  This NRC shall be sent if the length of the message is wrong. |
| [SDD\_ASW\_19H\_004] | 0x31 | **RequestOutOfRange:**  The server shall use this response code if it detects an error in the Communication Type parameter. |

Table : Supported NRC for 0x19

## iso14229\_serv22

[SDD\_ASW\_22H\_001]

**iso14229\_serv22 shall be the service interpreter for service 0x22.**

| **Function** | **iso14229\_serv22** |
| --- | --- |
| Module | iso14229\_serv22.c |
| Prototype | UDS\_Serv\_resptype\_En\_t iso14229\_serv22(UDS\_Serv\_St\_t\* UDS\_Serv\_pSt); |
| Description | Service interpreter function for service 0x22. |
| Parameter | UDS\_Serv\_St\_t\* UDS\_Serv\_pSt - request data buffer form distributor function. |
| Return | UDS\_Serv\_resptype\_En\_t Serv\_resptype\_En - Returns the type of response. |
| Algorithm | The Read Data by Identifier service allows the client to request data record values from the server identified by one or more data Identifiers. The client request message contains one or more two bytes data Identifier values that identify data record(s) maintained by the server.  The request message may contain the same data Identifier multiple times. The server shall treat each data Identifier as a separate parameter and respond with data for each data Identifier as often as requested. |
| Error handling | Check if the request contains a valid SID (0x22) |

Table : Function description of 0x22

[SDD\_ASW\_22H\_002]

|  |  |
| --- | --- |
| **Function** | **iso14229\_serv22** |
| Module | iso14229\_serv22.c |
| Prototype | bool DID\_check(uint16\_t arr[] , uint8\_t size) |
| Description | This Function is used to check the Data Identifier (DID) is valid or not. |
| Parameter | uint16\_t arr[], uint8\_t size |
| Return | bool |

[SDD\_ASW\_22H\_003]

|  |  |
| --- | --- |
| **Function** | **iso14229\_serv22** |
| Module | iso14229\_serv22.c |
| Prototype | bool check\_session (uint16\_t array[] , uint8\_t index) |
| Description | This Function is used to check the Current active Session. |
| Parameter | uint16\_t arr[], uint8\_t size |
| Return | bool |

**NRC handling:**

|  |  |  |
| --- | --- | --- |
| **Tags** | **NRC** | **Description** |
| [SDD\_ASW\_22H\_004] | 0x22 | **Condition not Corrected**  This NRC shall be sent if the condition is not corrected. |
| [SDD\_ASW\_22H\_005] | 0x13 | **Incorrect Message Length**  This NRC shall be sent if the length of the message is wrong. |
| [SDD\_ASW\_22H\_006] | 0x14 | **Response length Exceeded**  This NRC shall be sent if the response length is exceeded. |
| [SDD\_ASW\_22H\_007] | 0x33 | **Security Access Denied**  This NRC Shall be sent if the Security is not Gained. |
| [SDD\_ASW\_22H\_008] | 0x31 | **Request out of Range**  This NRC Shall be send if the DID is Not Correct and Current Session is not Supported. |

Table : Supported NRC for 0x22

## iso14229\_serv2E

[SDD\_ASW\_2EH\_001]

**iso14229\_serv2E shall be the service interpreter for service 0x2E.**

| **Function** | **iso14229\_serv2E** |
| --- | --- |
| Module | iso14229\_serv2E.c |
| Prototype | UDS\_Serv\_resptype\_En\_t iso14229\_serv2E (UDS\_Serv\_St\_t\* UDS\_Serv\_pSt); |
| Description | Service interpreter function for service 0x2E. |
| Parameter | UDS\_Serv\_St\_t\* UDS\_Serv\_pSt - request data buffer form distributor function. |
| Return | UDS\_Serv\_resptype\_En\_t Serv\_resptype\_En - Returns the type of response. |
| Algorithm | The Write Data by Identifier service allows the client to write information into the server at an internal location specified by the provided data identifier. The Write Data by Identifier service is used by the client to write a data Record to a server. The data is identified by a data Identifier and may or may not be secured. |
| Error handling | Check if the request contains a valid SID (0x2E) |

Table : Function description of 0x2E

[SDD\_ASW\_2EH\_002]

| **Function** | **iso14229\_serv2E** |
| --- | --- |
| Module | iso14229\_serv2E.c |
| Prototype | bool check\_Max\_len (); |
| Description | This function used to check DID’s Maximum length |
| Parameter | null |
| Return | bool |

[SDD\_ASW\_2EH\_003]

| **Function** | **iso14229\_serv2E** |
| --- | --- |
| Module | iso14229\_serv2E.c |
| Prototype | Bool\_Check\_Active\_fun\_2E (); |
| Description | This function used to check the Current active session. |
| Parameter | null |
| Return | bool |

NRC handling:

|  |  |  |
| --- | --- | --- |
| **Tags** | **NRC** | **Description** |
| [SDD\_ASW\_2EH\_004] | 0x72 | **General programming failure**  This NRC shall be returned if the server detects an error when writing to a memory location. |
| [SDD\_ASW\_2EH\_005] | 0x13 | **Invalid message length**  Mnemonic This NRC shall be sent if the length of the message is wrong. |
| [SDD\_ASW\_2EH\_006] | 0x22 | **Condition not correct**  This NRC shall be sent if the operating conditions of the server are not met to perform the required action |
| [SDD\_ASW\_2EH\_007] | 0x31 | **Request out of range**  the dataIdentifier in the request message is not supported in the server or the dataIdentifier is supported for read only purpose |
| [SDD\_ASW\_2EH\_008] | 0x33 | **Security Access Denied**  This NRC shall be sent if the dataIdentifier, which reference a specific address, is secured and the server is not in an unlocked state. |

Table : Supported NRC for 0x2E

## iso14229\_serv23

[SDD\_ASW\_23H\_001]

**iso14229\_serv23 shall be the service interpreter for service 0x23.**

| **Function** | **iso14229\_serv23** |
| --- | --- |
| Module | iso14229\_serv23.c |
| Prototype | UDS\_Serv\_resptype\_En\_t iso14229\_serv28(UDS\_Serv\_St\_t\* UDS\_Serv\_pSt); |
| Description | Service interpreter function for service 0x23. |
| Parameter | UDS\_Serv\_St\_t\* UDS\_Serv\_pSt - request data buffer from distributor function. |
| Return | UDS\_Serv\_resptype\_En\_t Serv\_resptype\_En - Returns the type of response. |
| Algorithm | The Read Memory By Address service allows the client to request memory data from the server via provided starting address and size of memory to be read. |
| Error handling | Check if the request contains a valid SID (0x23) |

Table : Function description of 0x23

[SDD\_ASW\_23H\_002]

| **Function** | **iso14229\_serv23** |
| --- | --- |
| Module | iso14229\_serv23.c |
| Prototype | bool addressSizeAndSessionCheck(uint32\_t MemomryStartingAddress\_u32,uint16\_t NumOfBytes\_u16, uint8\_t \*MemoryTableIndex\_pu8) |
| Description | This function is used to check starting address, memory size and current session is supported or not. |
| Parameter | uint32\_t MemomryStartingAddress\_u32: request starting memory address from iso14229\_serv23 function.  uint16\_t NumOfBytes\_u16: request memory size from iso14229\_serv23 function.  uint8\_t \*MemoryTableIndex\_pu8: This is used to get the predefined memory table index. |
| Return | bool |

[SDD\_ASW\_23H\_003]

| **Function** | **iso14229\_serv23** |
| --- | --- |
| Module | iso14229\_serv23.c |
| Prototype | status\_t DataReadWriteFromMemory(uint32\_t StartingMemoryAddress\_u32, uint16\_t MemorySize\_u16, uint8\_t ReadWrite\_u8, uint8\_t \*DataBuff\_u8, uint8\_t \*MemoryTableIndex\_u8) |
| Description | This function is used check the read/write access and call function to read data from memory. |
| Parameter | uint32\_t StartingMemoryAddress\_u32: request Starting memory address from service 0x23 function.  uint16\_t MemorySize\_u16: request memory size from service 23 function.  uint8\_t ReadWrite\_u8: requested to perform a read or write operation from service 0x23 function.  uint8\_t \*DataBuff\_Pu8: Used to store the read data.  Uint8\_t \*MemoryTableIndex\_u8: Reference to memory table index. |
| Return | status\_t |

[SDD\_ASW\_23H\_004]

| **Function** | **iso14229\_serv23** |
| --- | --- |
| Module | iso14229\_serv23.c |
| Prototype | void ReadDataFromMemory(uint8\_t \*TxBuff\_Pu8, uint32\_t MemoryStratingAddress\_u32, uint16\_t NumOfBytes\_u16) |
| Description | This function is used to read data from memory. |
| Parameter | uint16\_t \*TxLen\_pu16: request to update the transmit length.  uint32\_t MemoryStratingAddress\_u32: requested to starting address of memory to read the data.  uint16\_t NumOfBytes\_u16: requested to number of bytes to read the data from starting address. |
| Return | void. |

**NRC handling:**

|  |  |  |
| --- | --- | --- |
| **Tags** | **NRC** | **Description** |
| [SDD\_ASW\_23H\_005] | 0x13 | **Incorrect Message Length or Invalid Format:**  This NRC shall be sent if the length of the message is wrong. |
| [SDD\_ASW\_23H\_006] | 0x14 | **Response Too Long:**  This NRC shall be sent if the total length of the response message exceeds the limit of the underlying transport protocol or response buffer size. |
| [SDD\_ASW\_23H\_007] | 0x31 | **Request Out of Range:**  This NRC shall be sent if:  — Any memory address within the interval is invalid;  — Any memory address within the interval is restricted;  — The memory Size parameter value in the request message is not supported by the server;  — The specified address And Length Format Identifier is not valid;  — The memory Size parameter value in the request message is zero. |
| [SDD\_ASW\_23H\_008] | 0x33 | **Security Access Denied:**  This NRC shall be sent if any memory address within the interval [MA16, (MA16 + MS16 -116)] is secure and the server is locked. |

Table : Supported NRC for 0x23

## iso14229\_serv3D

[SDD\_ASW\_3DH\_001]

**iso14229\_serv3D shall be the service interpreter for service 0x3D.**

| **Function** | **iso14229\_serv3D** |
| --- | --- |
| Module | iso14229\_serv3D.c |
| Prototype | UDS\_Serv\_resptype\_En\_t iso14229\_serv3D(UDS\_Serv\_St\_t\* UDS\_Serv\_pSt); |
| Description | Service interpreter function for service 0x3D. |
| Parameter | UDS\_Serv\_St\_t\* UDS\_Serv\_pSt - request data buffer form distributor function. |
| Return | UDS\_Serv\_resptype\_En\_t Serv\_resptype\_En - Returns the type of response. |
| Algorithm | The Write Memory by Address service allows the client to write information into the server at one or more contiguous memory locations.  The Write Memory by Address request message writes information specified by the parameter data Record into the server at memory locations specified by parameters memory Address and memory Size. The number of bytes used for the memory Address and memory Size parameter is defined by address And Length Format Identifier (low and high nibble). It is also possible to use a fixed address And Length Format Identifier and unused bytes within the memory Address or memory Size parameter are padded with the value 0016 in the higher range address locations |
| Error handling | Check if the request contains a valid SID (0x3D) |

Table : Function description of 0x3D

[SDD\_ASW\_3DH\_002]

|  |  |
| --- | --- |
| **Function** | **iso14229\_serv3D** |
| Module | uds\_conf.c |
| Prototype | Status\_t DataReadWriteFromMemory(uint32\_t StartingMemoryAddress\_u32, uint16\_t MemorySize\_u16, uint8\_t ReadWrite\_u8,uint8\_t \*DataBuff\_Pu8, uint8\_t MemoryTableIndex\_u8) |
| Description | This Function is used to check the read/write access and call function to write data into memory. |
| Parameter | uint32\_t StartingMemoryAddress\_u32: request Starting memory address from service 0x23 function.  uint16\_t MemorySize\_u16: request memory size from service 23 function.  uint8\_t ReadWrite\_u8: requested to perform a read or write operation from service 0x23 function.  uint8\_t \*DataBuff\_Pu8: Used to store the read data.  Uint8\_t \*MemoryTableIndex\_u8: Reference to memory table index. |
| Return | Status\_t |

[SDD\_ASW\_3DH\_003]

|  |  |
| --- | --- |
| **Function** | **iso14229\_serv3D** |
| Module | iso14229\_serv3D.c |
| Prototype | Status\_t Writedatabyaddress\_fn(uint32\_t Starting\_address, UDS\_Srv\_St\_t\* data, uint16\_t size) |
| Description | This Function is used to write data into memory. |
| Parameter | uint32\_t Starting\_address: request Starting memory address from service 0x3D function.  uint16\_t Size: request memory size from service 0x3D function.  UDS\_Srv\_St\_t \*data: Used to store the write data. |
| Return | Status\_t |

**NRC handling:**

[SDD\_ASW\_3DH\_004]

|  |  |  |
| --- | --- | --- |
| **Tags** | **NRC** | **Description** |
| SDD\_ASW\_3DH\_004 | 0X31 | Request Out of Range.  — any memory address within the interval is invalid;  — any memory address within the interval is restricted;  — the memorySize parameter value in the request message is not supported by the server;  the specified addressAndLengthFormatIdentifier is not valid;  — the memorySize parameter value in the request message is zero. |
| SDD\_ASW\_3DH\_005 | 0x13 | Incorrect Message length or Invalid Format.  This NRC shall be sent if the length of the message is wrong. |
| SDD\_ASW\_3DH\_006 | 0x72 | General Program Failure.  This NRC shall be returned if the server detects an error when writing to a memory location. |
| SDD\_ASW\_3DH\_007 | 0x33 | Security Access Denied.  This NRC shall be sent if any memory address within the interval [MA16, (MA16 + MS16 116)] is secure and the server is locked. |

Table : Supported NRC for 0x3D

## iso14229\_serv2F

[SDD\_ASW\_2FH\_001]

**iso14229\_serv2F shall be the service interpreter for service 0x2F.**

| **Function** | **iso14229\_serv2F** |
| --- | --- |
| Module | iso14229\_serv2F.c |
| Prototype | UDS\_Serv\_resptype\_En\_t iso14229\_serv2F (UDS\_Serv\_St\_t\* UDS\_Serv\_pSt); |
| Description | Service interpreter function for service 0x2F. |
| Parameter | UDS\_Serv\_St\_t\* UDS\_Serv\_pSt - request data buffer form distributor function. |
| Return | UDS\_Serv\_resptype\_En\_t Serv\_resptype\_En - Returns the type of response. |
| Algorithm | The Input Output Control By Identifier service is used by the client to substitute a value for an input signal, internal server function and/or force control to a value for an output (actuator) of an electronic system. |
| Error handling | Check if the request contains a valid SID (0x2F) |

Table 3: Function description of 0x2F

[SDD\_ASW\_2FH\_002]

|  |  |
| --- | --- |
| **Function** | **iso14229\_serv2F** |
| Module | iso14229\_serv2F.c |
| Prototype | bool IS\_Request\_outof\_range(uint16\_t DID\_id\_u16) |
| Description | This Function is used to check the Data Identifier is valid or not. |
| Parameter | DID\_id\_u16 |
| Return | bool |

[SDD\_ASW\_2FH\_003]

|  |  |
| --- | --- |
| **Function** | **iso14229\_serv2F** |
| Module | iso14229\_serv2F.c |
| Prototype | void UDS\_Serv2F\_Timeout(void) |
| Description | This Function is called when there is a timeout. |
| Parameter | void |
| Return | void |

[SDD\_ASW\_2FH\_004]

|  |  |
| --- | --- |
| **Function** | **iso14229\_serv2F** |
| Module | iso14229\_serv2F.c |
| Prototype | void UDS\_Serv2F\_Init(void) |
| Description | This Function is used to Reset all the Data Identifier mode to false. |
| Parameter | void |
| Return | void |

[SDD\_ASW\_2FH\_005]

|  |  |
| --- | --- |
| **Function** | **iso14229\_serv2F** |
| Module | iso14229\_serv2F.c |
| Prototype | void UDS\_ResetStimulus(ISO14229\_DidList\_En\_t DidList\_En ); |
| Description | This Function is used to Reset the passed DID to the previous value or default value. |
| Parameter | ISO14229\_DidList\_En\_t DidList\_En |
| Return | void |

[SDD\_ASW\_2FH\_006]

|  |  |
| --- | --- |
| **Function** | **iso14229\_serv2F** |
| Module | iso14229\_serv2F.c |
| Prototype | void UDS SetStimulus(ISO14229\_DidList\_En\_t DidList\_En, uint8\_t \*ForceValue\_pu8) |
| Description | This Function is used to Set the DID with the value passed in Short\_term\_adjustment subfunction. |
| Parameter | ISO14229\_DidList\_En\_t DidList\_En,  uint8\_t \*ForceValue\_pu8 |
| Return | void |

**NRC handling:**

[SDD\_ASW\_22H\_007]

|  |  |  |
| --- | --- | --- |
| **Tags** | **NRC** | **Description** |
| SRS-ASW-2FH-008 | 0x22 | **Condition not Corrected**  This NRC shall be sent if the condition is not corrected. |
| SRS-ASW-2FH-009 | 0x13 | **Incorrect Message Length or Invalid Format:**  This NRC shall be sent if the length of the message is wrong. |
| SRS-ASW-2FH-010 | 0x31 | **Request out of Range**  This NRC Shall be send if the DID is Not Correct and Current Session is not Supported. |
| SRS-ASW-2FH-011 | 0x7F | **Service not supported in active Session.**  This NRC shall be sent if the DID is Not Supported in the current Session . |
| SRS-ASW-2FH-012 | 0x33 | **Security Access Denied**  This NRC Shall be sent if the Security is not Gained. |
| SRS-ASW-2FH-013 | 0x34 | **Authentication required**  This NRC shall be sent if the dataIdentifier is secured and the client has insufficient rights based on its Authentication state. |

Table 4: Supported NRC for 0x2F

# Traceability of ASW SDD and SRS

|  |  |
| --- | --- |
| **SDD Tags** | **SRS Tags** |
| SDD\_ASW\_10H\_001 | SRS\_ASW\_10H\_001 |
| SDD\_ASW\_10H\_002 | SRS\_ASW\_10H\_003 |
| SDD\_ASW\_10H\_003 | SRS\_ASW\_10H\_004 |
| SDD\_ASW\_10H\_004 | SRS\_ASW\_10H\_005 |
| SDD\_ASW\_10H\_007 | SRS\_ASW\_10H\_008 |
|  |  |
| SDD\_ASW\_11H\_001 | SRS\_ASW\_11H\_001 |
| SDD\_ASW\_11H\_002 | SRS\_ASW\_11H\_011 |
| SDD\_ASW\_11H\_003 | SRS\_ASW\_11H\_012 |
| SDD\_ASW\_11H\_004 | SRS\_ASW\_11H\_013 |
| SDD\_ASW\_11H\_007 | SRS\_ASW\_11H\_016 |
|  |  |
| SDD\_ASW\_27H\_001 | SRS\_ASW\_27H\_001 |
| SDD\_ASW\_27H\_007 | SRS\_ASW\_27H\_012 |
| SDD\_ASW\_27H\_008 | SRS\_ASW\_27H\_013 |
| SDD\_ASW\_27H\_010 | SRS\_ASW\_27H\_010 |
| SDD\_ASW\_27H\_011 | SRS\_ASW\_27H\_011 |
| SDD\_ASW\_27H\_012 | SRS\_ASW\_27H\_012 |
| SDD\_ASW\_27H\_013 | SRS\_ASW\_27H\_009 |
| SDD\_ASW\_27H\_014 | SRS\_ASW\_27H\_013 |
| SDD\_ASW\_27H\_015 | SRS\_ASW\_27H\_014 |
|  |  |
| SDD\_ASW\_3EH\_001 | SRS\_ASW\_3EH\_001 |
| SDD\_ASW\_3EH\_003 | SRS\_ASW\_3EH\_005 |
| SDD\_ASW\_3EH\_004 | SRS\_ASW\_3EH\_006 |
|  |  |
| SDD\_ASW\_28H\_001 | SRS\_ASW\_28H\_001 |
| SDD\_ASW\_28H\_003 | SRS\_ASW\_28H\_007 |
| SDD\_ASW\_28H\_001 | SRS\_ASW\_28H\_008 |
| SDD\_ASW\_28H\_004 | SRS\_ASW\_28H\_009 |
| SDD\_ASW\_28H\_006 | SRS\_ASW\_28H\_012 |
| SDD\_ASW\_28H\_007 | SRS\_ASW\_28H\_013 |
| SDD\_ASW\_28H\_009 | SRS\_ASW\_28H\_016 |
|  |  |
| SDD\_ASW\_22H\_001 | SRS\_ASW\_22H\_001 |
| SDD\_ASW\_22H\_006 | SRS\_ASW\_22H\_006 |
| SDD\_ASW\_22H\_007 | SRS\_ASW\_22H\_007 |
| SDD\_ASW\_22H\_008 | SRS\_ASW\_22H\_008 |
| SDD\_ASW\_22H\_009 | SRS\_ASW\_22H\_009 |
| SDD\_ASW\_2EH\_010 | SRS\_ASW\_2EH\_010 |
| SDD\_ASW\_2EH\_011 | SRS\_ASW\_2EH\_011 |
|  |  |
| SDD\_ASW\_2EH\_001 | SRS\_ASW\_2EH\_001 |
| SDD\_ASW\_2EH\_006 | SRS\_ASW\_2EH\_006 |
| SDD\_ASW\_2EH\_007 | SRS\_ASW\_2EH\_007 |
| SDD\_ASW\_2EH\_008 | SRS\_ASW\_2EH\_008 |
| SDD\_ASW\_2EH\_009 | SRS\_ASW\_2EH\_009 |
| SDD\_ASW\_2EH\_010 | SRS\_ASW\_2EH\_010 |
| SDD\_ASW\_2EH\_011 | SRS\_ASW\_2EH\_011 |
|  |  |
| SDD\_ASW\_23H\_001 | SRS\_ASW\_23H\_001 |
| SDD\_ASW\_23H \_005 | SRS\_ASW\_23H\_009 |
| SDD\_ASW\_23H \_006 | SRS\_ASW\_23H\_010 |
| SDD\_ASW\_23H \_007 | SRS\_ASW\_23H\_011 |
| SDD\_ASW\_23H \_008 | SRS\_ASW\_23H\_012 |
| SDD\_ASW\_23H \_009 | SRS\_ASW\_23H\_013 |
|  |  |
| SDD\_ASW\_3DH\_001 | SRS\_ASW\_3DH\_001 |
| SDD\_ASW\_3DH\_002 | SRS\_ASW\_3DH\_001 |
| SDD\_ASW\_3DH\_003 | SRS\_ASW\_3DH\_001 |
| SDD\_ASW\_3DH\_004 | SRS\_ASW\_3DH\_006 |
| SDD\_ASW\_3DH\_005 | SRS\_ASW\_3DH\_007 |
| SDD\_ASW\_3DH\_006 | SRS\_ASW\_3DH\_008 |
| SDD\_ASW\_3DH\_007 | SRS\_ASW\_3DH\_009 |
| SDD\_ASW\_3DH\_008 | SRS\_ASW\_3DH\_010 |
|  |  |
| SDD\_ASW\_2FH\_001 | SRS-ASW-2FH-001 |
| SDD\_ASW\_2FH\_002 | SRS-ASW-2FH-011 |
| SDD\_ASW\_2FH\_005 | SRS\_ASW\_2FH\_004 |
| SDD\_ASW\_2FH\_006 | SRS\_ASW\_2FH\_006 |
| SDD\_ASW\_2FH\_008 | SDD\_ASW\_2FH\_009 |
| SDD\_ASW\_2FH\_009 | SDD\_ASW\_2FH\_010 |
| SDD\_ASW\_2FH\_010 | SDD\_ASW\_2FH\_011 |
| SDD\_ASW\_2FH\_011 | SDD\_ASW\_2FH\_012 |
| SDD\_ASW\_2FH\_012 | SDD\_ASW\_2FH\_013 |
| SDD\_ASW\_2FH\_013 | SDD\_ASW\_2FH\_014 |