**UDS over UART**

UDS - ASW Specification and Requirements

**Version 1.0**

**Customer: Royal Enfield**

Table of Contents

[1. Introduction 5](#_Toc148610738)

[1.1 Abbreviations and Acronyms 5](#_Toc148610739)

[1.2 Reference Document 5](#_Toc148610740)

[1.3 Diagnostic Testing and Programming Station Set Up 6](#_Toc148610741)

[1.4 Supported CAN ID’s 6](#_Toc148610742)

[1.5 Description of Services for Application 7](#_Toc148610743)

[1.6 Service Supported in Different Sessions and Addressing 7](#_Toc148610744)

[2. ASW SERVICES: 8](#_Toc148610745)

[2.1 Diagnostic session Control (0x10): 8](#_Toc148610746)

[2.1.1 Diagnostic Session Control Request Frame Format: 8](#_Toc148610747)

[2.1.2 Diagnostic Session Control Positive Response Frame Format: 9](#_Toc148610748)

[2.1.3 Diagnostic Session Control Negative Response Frame Format: 9](#_Toc148610749)

[2.1.4 Diagnostic Session Control Supported Negative Response Codes (NRC’s): 10](#_Toc148610750)

[2.2 ECU Reset service (0x11): 11](#_Toc148610751)

[2.2.1 ECU Reset Request Frame Format: 11](#_Toc148610752)

[2.2.2 ECU Reset Positive Response Frame: 13](#_Toc148610753)

[2.2.3 ECU Reset Negative Response Frame Format: 13](#_Toc148610754)

[2.2.4 ECU Reset Supported Negative Response Codes: 13](#_Toc148610755)

[2.3 Security Access service (0x27): 14](#_Toc148610756)

[2.3.1 Security Access Request Seed Frame Format: 14](#_Toc148610757)

[2.3.2 Security Access Positive Seed Response Frame Format: 14](#_Toc148610758)

[2.3.3 Security Access Send Key Request Frame Format: 15](#_Toc148610759)

[2.3.4 Security Access Positive Response Frame Format: 16](#_Toc148610760)

[2.3.5 Security Access Negative Response Frame Format: 17](#_Toc148610761)

[2.3.6 Security Service Supported Negative Response Codes (NRC’s): 17](#_Toc148610762)

[2.4 Read Data by Identifier service (0x22): 18](#_Toc148610763)

[2.4.1 Read data by identifier Request Frame Format: 18](#_Toc148610764)

[2.4.2 Read data by identifier Service Positive Response Frame Format: 19](#_Toc148610765)

[2.4.3 Read data by identifier Service Negative Response Frame Format: 19](#_Toc148610766)

[2.4.4 Read data by identifier Service Negative Response codes (NRC’s): 19](#_Toc148610767)

[2.5 Write Memory by Address (0X3D): 20](#_Toc148610768)

[2.5.1 Write Memory by Address Request Frame Format: 20](#_Toc148610769)

[2.5.2 Write Memory by Address Positive Response Frame Format: 21](#_Toc148610770)

[2.5.3 Write Memory by Address Negative Response Frame Format: 22](#_Toc148610771)

[2.5.4 Write Memory by Address Supported Negative Response Codes (NRC’s): 23](#_Toc148610772)

[2.6 Communication Control service(0x28): 24](#_Toc148610773)

[2.6.1 Communication Control Request Frame Format: 24](#_Toc148610774)

[2.6.2 Communication Control Positive Response Frame Format: 25](#_Toc148610775)

[2.6.3 Communication Control Negative Response Frame Format: 25](#_Toc148610776)

[2.6.4 Communication Control Negative Response Code: 26](#_Toc148610777)

[2.7 Write Data by Identifier (0x2E): 27](#_Toc148610778)

[2.7.1 Write Data by Identifier Request Frame Format: 27](#_Toc148610779)

[2.7.2 Write Data by Identifier Positive Response Frame Format: 27](#_Toc148610780)

[2.7.3 Write Data by Identifier Negative Response Frame Format: 28](#_Toc148610781)

[2.7.4 Write Memory by Address Negative Response Codes: 28](#_Toc148610782)

[2.8 Tester Present service (0x3E): 29](#_Toc148610783)

[2.8.1 Tester Present Request Frame Format: 29](#_Toc148610784)

[2.8.2 Tester Present Positive Response Frame Format: 29](#_Toc148610785)

[2.8.3 Tester Present Negative Response Frame Format: 30](#_Toc148610786)

[2.8.4 Tester Present Negative Response Codes (NRC’s) 30](#_Toc148610787)

[2.9 Read Memory by Address (0x23): 31](#_Toc148610788)

[2.9.1 Read memory by Address Request Frame Format: 31](#_Toc148610789)

[2.9.2 Read memory By Address Positive Response Frame Format: 32](#_Toc148610790)

[2.9.3 Read Memory By Address Negative Response Frame Format: 33](#_Toc148610791)

[2.9.4 Read memory by Address Negative Response Code: 33](#_Toc148610792)

[2.10 Clear Diagnostic Information (0x14): 34](#_Toc148610793)

[2.10.1 Clear Diagnostic Information Request Frame Format: 34](#_Toc148610794)

[2.10.2 Clear Diagnostic Information Positive Response Frame Format: 34](#_Toc148610795)

[2.10.3 Clear Diagnostic Information Negative Response Frame Format: 35](#_Toc148610796)

[2.10.4 Clear Diagnostic Information Negative Response Codes (NRC’s): 35](#_Toc148610797)

[2.11 Read DTC Information (0x19): 36](#_Toc148610798)

[2.11.1 Read DTC Information Request Frame Format: 36](#_Toc148610799)

[2.11.2 Read DTC Information Positive Response Frame Format: 38](#_Toc148610800)

[2.11.3 Read DTC Information Negative Response Frame Format: 38](#_Toc148610801)

[2.11.4 Read DTC Information Negative Response Codes (NRC’s): 38](#_Toc148610802)

[2.12 Input Output Control By Identifier (0x2F) service: 39](#_Toc148610803)

[2.12.1 Input Output Control By Identifier Request Frame Format: 39](#_Toc148610804)

[2.12.2 Input Output Control by Identifier Positive Response Frame Format: 40](#_Toc148610805)

[2.12.3 Input Output Control by Identifier Negative Response Frame Format: 40](#_Toc148610806)

[2.12.4 Input Output Control by Identifier negative response code (NRC): 41](#_Toc148610807)

[Table 1: Abbreviation 5](#_Toc146020008)

[Table 2: Supported CAN Id 6](#_Toc146020009)

**Foreword**

The document has been drafted based on ISO 14229/ISO 15765-3 which contains the Re-programming and diagnostic testing requirements for ROYAL ENFIELD UDS. The document also provides the details regarding the implementation of services and the testing methods to validate the implementation. Adherence to this document meticulously is of utmost importance in the implementation of diagnostic services in the respective ECU’s.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Version | Version Description | Date | Author | Reviewed By | Approved By |
| 1.0 | Initial Version | 22/02/2024 | Venu & Jayanth |  |  |

# Introduction

This document describes the detailed description of services for application related to UDS services to be followed during the UDS.

## Abbreviations and Acronyms

| **Acronyms / Definitions / Terms** | **Description** |
| --- | --- |
| SOM | System On Module |
| UDS | Unified Diagnostic Services |
| ASW | Application Software |
| BSW | Bootloader Software |
| vCAN | Virtual Controller Area Network |
| DID | Data Identifier |

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.

## Diagnostic Testing and Programming Station Set Up

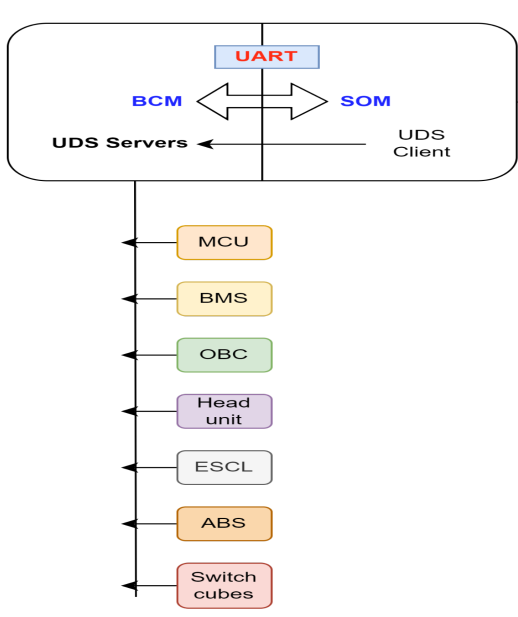


Figure 1: Testing and Programming Station Set Up

The above figure depicts the arrangement for diagnostic testing and programming of the application software (ASW) at the PC/After sales programming station. The SOM transmits the UDS commands to the BCM via UART, the BCM acts as a gateway and relays the commands over to the relevant ECUs via CAN bus.

## Supported CAN ID’s

|  |  |
| --- | --- |
| **CAN Message Direction** | **ID** |
| Request ID (Physical) | 0x6BB |
| Response ID (Physical) | 0x700 |

Table 2: Supported CAN Id

## Description of Services for Application

A SOM will communicate with ECU using UDS Services in a Request-Response cycle. Service Identifier (SID) is one-byte data that identifies the request-response sequence. For Every service request from tester, the ECU provides a positive response or negative response as outlined below.

## Service Supported in Different Sessions and Addressing

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **SL**  **NO** | **FUNCTIONAL UNIT** | **SERVICE** | **ID** | **SECURITY ACCESS** | **DEFAULT SESSION** | **NON-DEFAULT SESSION** | |
| **EXTENDED** | **PROGRAMMING** |
| 1 | Diagnostics and Communication Management | Diagnostics session control | 0x10 | 🗷 | 🗹 | 🗹 | 🗹 |
| 2 | Security Access | 0x27 | 🗷 | 🗷 | 🗹 | 🗹 |
| 3 | ECU Reset | 0x11 | 🗹 | 🗷 | 🗹 | 🗹 |
| 4 | Communication control | 0x28 | 🗹 | 🗷 | 🗹 | 🗹 |
| 5 | Write Data by  Identifier | 0x2E | 🗹 | 🗷 | 🗹 | 🗹 |
| 6 | Read data by Identifier | 0x22 | 🗹 | 🗷 | 🗹 | 🗹 |
| 7 | Write memory by Address | 0x3D | 🗹 | 🗷 | 🗹 | 🗹 |
| 8 | Tester Present | 0x3E | 🗹 | 🗹 | 🗹 | 🗹 |
| 9 | Read memory by Address | 0x23 | 🗹 | 🗷 | 🗹 | 🗹 |
| 10 | Clear Diagnostic  information | 0x14 | 🗷 | 🗹 | 🗹 | 🗹 |
| 11 | Read DTC Information | 0x19 | 🗷 | 🗹 | 🗹 | 🗹 |
| 12 | Input Output Control by Identifier | 0x2F | 🗹 | 🗷 | 🗹 | 🗹 |

# ASW SERVICES:

|  |  |  |
| --- | --- | --- |
| **Parameters** | **Parameter Size (#of bytes)** | **Value** |
| **Start of Frame (SOF)** | 2 bytes | 0xA5A5 |
| **Frame Length** | 1 byte | Length |
| **Source ID** | 4 bits (Lower Nibble) | 0x00: from SOM to BCM  0x01: from BCM to SOM |
| **Message Type** | 4 bits (Higher Nibble) | 0x00: Standard ID  0x01: Extended ID |
| **CAN ID** | 4 bytes | Can id |
| **Payload** | 8 bytes | Data bytes |
| **Checksum** | 1 byte | CRC8 - Checksum is calculated on payload bytes. |
| **End of Frame (EOF)** | 2 bytes | 0x5A5A |

**SOF:** SOF used to indicate the start of the frame. It is of 2 bytes. 0xA5A5 - it is user defined

**Frame Length:** Frame Length is used to indicate the length of the source id, Message type, CAN id & payload. It is of 1 byte.

It is used to define number of bytes present in the Payload.

**Source Id:** It is 4 bits field; it is used to indicate the node id.

0x00: SOM to BCM

0x01: BCM to SOM

**Message Type:** It is 4 bits field. It is used to indicate the message id type.0x00: Standard ID

0x01: Extended ID

**Payload:** The Payload contains the Data Record. Size is 8 bytes.

**Checksum:** Checksum is used to check the correctness of the data transferred. The checksum is calculated on Payload.

**End Of Frame:** EOF used to indicate the End of the frame. It is of 2 bytes. 0x5A5A - it is user defined.

## Diagnostic session Control (0x10):

**SRS-ASW-10H-001**

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(s). This service provides the capability that the server(s) can report data link layer specific parameter values valid for the enabled diagnostic session (eg: timing parameter values).

There shall always be exactly one diagnostic session active in a server. A server shall always start the default diagnostic session when powered up. If no other diagnostic session is started, then the default diagnostic session shall be running as long as the server is powered.

### Diagnostic Session Control Request Frame Format:

**SRS-ASW-10H-002**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SOF** | **Frame Length** | **Message**  **Type** | | **ID** | **Payload Data Frame** | | | | | | | | **Checksum** | **EOF** |
| **0xA5**  **0xA5** | 0x0C | SRC ID | MSG Type | $ID | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | CRC8 |  |
| 0 | 0 | **Data**  **length** | **SID**  **REQ** | **SF** | **Data** | **Data** | **Data** | **Data** | **Data** | **0x5A**  **0x5A** |
| 0x02 | 0x10 | XX | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 |  |

Table:1 Diagnostic Session Control Request Frame Format

**$ID** – CAN ID

**XX –** Sub function.

1. **Default Session (0x01): SRS-ASW-10H-003**

This diagnostic session enables the default diagnostic session in the server(s).

1. **Programming Session(0x02): SRS-ASW-10H-004**

This diagnostic Session enables all diagnostic services required to support the memory programming of a server.

1. **Extended Diagnostic Session(0x03): SRS-ASW-10H-005**

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.

1. **Suppressed positive Request Message (0x81): SRS-ASW-10H-006**

This diagnostic session enables the default diagnostic session in the server(s).

1. **Suppressed positive Request Message (0x82): SRS-ASW-10H-007**

This diagnostic Session enables all diagnostic services required to support the memory programming of a server.

1. **Suppressed positive Request Message (0x83): SRS-ASW-10H-008**

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.

**Notes:** The suppressed positive Request frame, there is no response frame.

### Diagnostic Session Control Positive Response Frame Format:

**SRS-ASW-10H-009**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SOF** | **Frame Length** | **Message**  **Type** | | **ID** | **Payload Data Frame** | | | | | | | | **Checksum** | **EOF** |
| **0xA5**  **0xA5** | 0x0C | SRC ID | MSG Type | $ID | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |  | **0x5A**  **0x5A** |
| 1 | 0 | **Data**  **length** | **SID**  **RESP** | **SF** | **Data** | **Data** | **Data** | **Data** | **Data** | CRC8 |
| 0x02 | 0x50 | XX | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 |  |

Table:2 Diagnostic Session Control Positive Response Frame Format

**XX** – Sub function.

### Diagnostic Session Control Negative Response Frame Format:

**SRS-ASW-10H-010**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SOF** | **Frame Length** | **Message**  **Type** | | **ID** | **Payload Data Frame** | | | | | | | | **Checksum** | **EOF** |
| **0xA5**  **0xA5** | 0x0C | SRC ID | MSG Type | $ID | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |  | **0x5A**  **0x5A** |
| 1 | 0 | **Data**  **length** | **SID**  **RESP** | **SF** | **NRC** | **Data** | **Data** | **Data** | **Data** | CRC8 |
| 0x03 | 0x7F | 0x10 | XX | 0x00 | 0x00 | 0x00 | 0x00 |  |

Table:3 Diagnostic Session Control Negative Response Frame Format.

**XX –** Negative Response Code.

### Diagnostic Session Control Supported Negative Response Codes (NRC’s):

**SRS-ASW-10H-011**

|  |  |  |
| --- | --- | --- |
| **Tags** | **NRC** | **Description** |
| **SRS-ASW-10H-012** | **0X12** | Sub Function Not Supported |
| **SRS-ASW-10H-013** | **0x13** | Incorrect Message length or Invalid Format |
| **SRS-ASW-10H-014** | **0x22** | Condition Not Correct |
| **SRS-ASW-10H-015** | **0x7F** | Service not support active session. |

Table:4 Diagnostic Session Control Negative Response Code.

## ECU Reset service (0x11):

**SRS-ASW-11H-01**

The function of the ECU Reset Service Identifier (0x11) is to reset the ECU/Server in a different format according to the problem requirement.

### 2.2.1 ECU Reset Request Frame Format:

**SRS-ASW-11H-02**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SOF** | **Frame Length** | **Message**  **Type** | | **ID** | **Payload Data Frame** | | | | | | | | **Checksum** | **EOF** |
| **0xA5**  **0xA5** | 0x0C | SRC ID | MSG Type | $ID | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |  | **0x5A**  **0x5A** |
| 0 | 0 | **Data**  **length** | **SID**  **REQ** | **SF** | **Data** | **Data** | **Data** | **Data** | **Data** | CRC8 |
| 0x02 | 0x11 | XX | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 |  |

Table:1 ECU Reset Request Frame Format.

**XX –** Sub function

**SF REQ XX**

* **Hard Reset:(0x01) SRS-ASW-11H-03**

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 (i.e battery). The performed action is implementation specific and not defined by this document. It might result in the re-initialization of both volatile memory and non-volatile memory locations to predetermined values.

* **Key off-on Reset:(0x02) SRS-ASW-11H-04**

In simple terms key OFF – ON Resetis equivalent to Ignition OFF – ON process in the ECU reset Service Identifier(0x11).

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) SRS-ASW-11H-05**

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.

Soft reset is an application software it’ll restart ECU immediately, this is a very common reset functionality which is normally used to restart the ECU.

* **Suppressed Positive Request Message :(0x81) SRS-ASW-11H-06**

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 (i.e battery). The performed action is implementation specific and not defined by this document. It might result in the re-initialization of both volatile memory and non-volatile memory locations to predetermined values.

* **Suppressed Positive Request Message:(0x82) SRS-ASW-11H-07**

In simple terms key OFF – ON Resetis equivalent to Ignition OFF – ON process in the ECU reset Service Identifier(0x11).

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.

* **Suppressed Positive Request Message:(0x83) SRS-ASW-11H-08**

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.

Soft reset is an application software it’ll restart ECU immediately, this is a very common reset functionality which is normally used to restart the ECU.

**Notes:** The suppressed positive Request frame, there is no response frame.

### ECU Reset Positive Response Frame:

**SRS-ASW-11H-09**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SOF** | **Frame Length** | **Message**  **Type** | | **ID** | **Payload Data Frame** | | | | | | | | **Checksum** | **EOF** |
| **0xA5**  **0xA5** | 0x0C | SRC ID | MSG Type | $ID | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |  | **0x5A**  **0x5A** |
| 1 | 0 | **Data**  **length** | **SID**  **RESP** | **SF** | **Data** | **Data** | **Data** | **Data** | **Data** | CRC8 |
| 0x02 | 0x51 | XX | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 |  |

Table:2 ECU Reset Request Positive Response Frame Format.

**XX – Sub function**

### ECU Reset Negative Response Frame Format:

**SRS-ASW-11H-10**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SOF** | **Frame Length** | **Message**  **Type** | | **ID** | **Payload Data Frame** | | | | | | | | **Checksum** | **EOF** |
| **0xA5**  **0xA5** | 0x0C | SRC ID | MSG Type | $ID | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |  | **0x5A**  **0x5A** |
| 1 | 0 | **Data**  **length** | **SID**  **RESP** | **SID** | **NRC** | **Data** | **Data** | **Data** | **Data** | CRC8 |
| 0x03 | 0x7F | 0x11 | XX | 0x00 | 0x00 | 0x00 | 0x00 |  |

Table:3 ECU Reset Negative Response Frame Format.

**XX –** Negative Response Code**.**

### ECU Reset Supported Negative Response Codes:

**SRS-ASW-11H-11**

|  |  |  |
| --- | --- | --- |
| Tags | NRC | Description |
| **SRS-ASW-11H-12** | 0x12 | Sub function is not supported |
| **SRS-ASW-11H-13** | 0x13 | Invalid message length |
| **SRS-ASW-11H-14** | 0x22 | Condition not supported |
| **SRS-ASW-11H-15** | 0x33 | Security access denied |
| **SRS-ASW-11H-16** | 0x7F | Service not supported in active session |
| **SRS-ASW-11H-17** | 0x11 | Service is not supported |

## Security Access service (0x27):

**SRS-ASW-27H-01**

The purpose of this service is to provide a means to access data and/or diagnostic services, which have restricted access for security, emissions, or safety reasons. Diagnostic services for downloading/uploading routines or data into a server and reading specific memory locations from a server are situations where security access may be required. Improper routines or data downloaded into a server could potentially damage the electronics or other vehicle components or risk the vehicle’s compliance to emission, safety, or security standards. The security concept uses a seed and key relationship.

A typical example of the use of this service is as follows:

— client requests the “Seed”,

— server sends the “Seed”,

— client sends the “Key” (appropriate for the Seed received),

— server responds that the “Key” was valid and that it will unlock itself.

### Security Access Request Seed Frame Format:

**SRS-ASW-27H-02**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SOF** | **Frame Length** | **Message**  **Type** | | **ID** | **Payload Data Frame** | | | | | | | | **Checksum** | **EOF** |
| **0xA5**  **0xA5** | 0x0C | SRC ID | MSG Type | $ID | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |  | **0x5A**  **0x5A** |
| 0 | 0 | **Data**  **length** | **SID**  **REQ** | **SF** | **Data** | **Data** | **Data** | **Data** | **Data** | CRC8 |
| 0x02 | 0x27 | YY | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 |  |

Table:1 Security Seed Request Frame Format.

**YY** – Request Seed (0x01, 0x03)

### Security Access Positive Seed Response Frame Format:

**SRS-ASW-27H-03**

**First frame from Server to client:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SOF** | **Frame Length** | **Message**  **Type** | | **ID** | **Payload Data Frame** | | | | | | | | **Checksum** | **EOF** |
| **0xA5**  **0xA5** | 0x0C | SRC ID | MSG Type | $ID | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |  | **0x5A**  **0x5A** |
| 1 | 0 | **First**  **Frame** | **Data Length** | **SID**  **RESP** | **Sub id** | **Data** | **Data** | **Data** | **Data** | CRC8 |
| 0x10 | 0x12 | 0x67 | YY | XX | XX | XX | XX |  |

Table:2 Security Seed Response Frame Format.

**XX** – 16 bytes of Seed Generated by Server.

**Flow control Response frame:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SOF** | **Frame Length** | **Message**  **Type** | | **ID** | **Payload Data Frame** | | | | | | | | **Checksum** | **EOF** |
| **0xA5**  **0xA5** | 0x0C | SRC ID | MSG Type | $ID | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |  | **0x5A**  **0x5A** |
| 0 | 0 | **FC**  **Frame** | **Data** | **Data** | **Data** | **Data** | **Data** | **Data** | **Data** | CRC8 |
| 0x30 | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 |  |

**Consecutive Request frame:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SOF** | **Frame Length** | **Message**  **Type** | | **ID** | **Payload Data Frame** | | | | | | | | **Checksum** | **EOF** |
| **0xA5**  **0xA5** | 0x0C | SRC ID | MSG Type | $ID | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |  | **0x5A**  **0x5A** |
| 1 | 0 | **CS**  **Frame** | **Data** | **Data** | **Data** | **Data** | **Data** | **Data** | **Data** | CRC8 |
| 0x2S | XX | XX | XX | XX | XX | XX | XX |  |

Table:3 Security Seed Response Frame Format.

**S**: Sequence Counter

**XX**: Seed Value

### Security Access Send Key Request Frame Format:

**SRS-ASW-27H-04**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SOF** | **Frame Length** | **Message**  **Type** | | **ID** | **Payload Data Frame** | | | | | | | | **Checksum** | **EOF** |
| **0xA5**  **0xA5** | 0x0C | SRC ID | MSG Type | $ID | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |  | **0x5A**  **0x5A** |
| 0 | 0 | **First**  **Frame** | **Data Length** | **SID**  **REQ** | **Sub id** | **Data** | **Data** | **Data** | **Data** | CRC8 |
| 0x10 | 0x12 | 0x27 | ZZ | YY | YY | YY | YY |  |

Table:4 Security Send Key Request Frame Format.

**YY** - Send Keys based on received Seed value.

**ZZ** – Send Key (0x02, 0x04)

**Flow control Response frame:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SOF** | **Frame Length** | **Message**  **Type** | | **ID** | **Payload Data Frame** | | | | | | | | **Checksum** | **EOF** |
| **0xA5**  **0xA5** | 0x0C | SRC ID | MSG Type | $ID | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |  | **0x5A**  **0x5A** |
| 1 | 0 | **FC**  **Frame** | **Data** | **Data** | **Data** | **Data** | **Data** | **Data** | **Data** | CRC8 |
| 0x30 | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 |  |

**Consecutive Request frame:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SOF** | **Frame Length** | **Message**  **Type** | | **ID** | **Payload Data Frame** | | | | | | | | **Checksum** | **EOF** |
| **0xA5**  **0xA5** | 0x0C | SRC ID | MSG Type | $ID | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |  | **0x5A**  **0x5A** |
| 0 | 0 | **CS**  **Frame** | **Data** | **Data** | **Data** | **Data** | **Data** | **Data** | **Data** | CRC8 |
| 0x2S | YY | YY | YY | YY | YY | YY | YY |  |

Table:5 Security Key Request Frame Format.

**S**: Sequence Counter

**YY** - Send Keys based on received Seed value.

### Security Access Positive Response Frame Format:

**SRS-ASW-27H-05**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SOF** | **Frame Length** | **Message**  **Type** | | **ID** | **Payload Data Frame** | | | | | | | | **Checksum** | **EOF** |
| **0xA5**  **0xA5** | 0x0C | SRC ID | MSG Type | $ID | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |  | **0x5A**  **0x5A** |
| 1 | 0 | **Data**  **length** | **SID**  **RESP** | **SF** | **Data** | **Data** | **Data** | **Data** | **Data** | CRC8 |
| 0x02 | 0x67 | XX | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 |  |

Table:6 Security Positive Response Frame Format.

**XX** – Send Key (0x02, 0x04).

### Security Access Negative Response Frame Format:

**SRS-ASW-27H-06**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SOF** | **Frame Length** | **Message**  **Type** | | **ID** | **Payload Data Frame** | | | | | | | | **Checksum** | **EOF** |
| **0xA5**  **0xA5** | 0x0C | SRC ID | MSG Type | $ID | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |  | **0x5A**  **0x5A** |
| 1 | 0 | **Data**  **length** | **SID**  **RESP** | **SID** | **NRC** | **Data** | **Data** | **Data** | **Data** | CRC8 |
| 0x03 | 0x7F | 0x27 | XX | 0x00 | 0x00 | 0x00 | 0x00 |  |

Table:7 Security Negative Response Frame Format.

**XX** – Negative Response Code

### Security Service Supported Negative Response Codes (NRC’s):

|  |  |  |
| --- | --- | --- |
| **Tags** | **NRC** | **Description** |
| **SRS-ASW-27H-07** | 0x12 | Sub Function Not Supported |
| **SRS-ASW-27H-08** | 0x11 | Service not Supported |
| **SRS-ASW-27H-09** | 0x24 | Request Sequence Error |
| **SRS-ASW-27H-10** | 0x35 | Invalid Key |
| **SRS-ASW-27H-11** | 0x36 | Exceeded number of attempts |
| **SRS-ASW-27H-12** | 0x13 | Incorrect Message Length |
| **SRS-ASW-27H-13** | 0x37 | Required Time Delay Not Expired |
| **SRS-ASW-27H-14** | 0x7F | Service Inactive session |

## Read Data by Identifier service (0x22):

**SRS-ASW-22H-00**1

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 format and definition of the data Record shall be vehicle manufacturer or system supplier specific, and may include analog input and output signals, digital input and output signals, internal data, and system status information if supported by the server.

The server may limit the number of data Identifiers that can be simultaneously requested as agreed upon by the vehicle manufacturer and system supplier. Upon receiving a Read Data by Identifier request, the server shall access the data elements of the records specified by the data Identifier parameter(s) and transmit their value in one single Read Data by Identifier positive response containing the associated data Record parameter(s). 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.

### Read data by identifier Request Frame Format:

**SRS-ASW-22H-002**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SOF** | **Frame Length** | **Message**  **Type** | | **ID** | **Payload Data Frame** | | | | | | | | **Checksum** | **EOF** |
| **0xA5**  **0xA5** | 0x0C | SRC ID | MSG Type | $ID | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |  | **0x5A**  **0x5A** |
| 0 | 0 | **Data**  **length** | **SID**  **REQ** | **DID** | **DID** | **Data** | **Data** | **Data** | **Data** | CRC8 |
| YY | 0x22 | XX | XX | 0x00 | 0x00 | 0x00 | 0x00 |  |

Table:1 Read data by identifier request frame format.

**XX –** Data Identifier

**YY-** Data Length Depend on the Request DID

### Read data by identifier Service Positive Response Frame Format:

**SRS-ASW-22H-003**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SOF** | **Frame Length** | **Message**  **Type** | | **ID** | **Payload Data Frame** | | | | | | | | **Checksum** | **EOF** |
| **0xA5**  **0xA5** | 0x0C | SRC ID | MSG Type | $ID | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |  | **0x5A**  **0x5A** |
| 1 | 0 | **Data**  **length** | **SID**  **RESP** | **DID** | **DID** | **Data** | **Data** | **Data** | **Data** | CRC8 |
| YY | 0x62 | XX | XX | DR | DR | 0x00 | 0x00 |  |

**XX** – Data Identifier

**YY-** Data Length Depends on the data Present in the DID.

**DR –** Data Record

### 2.4.3 Read data by identifier Service Negative Response Frame Format:

**SRS-ASW-22H-004**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SOF** | **Frame Length** | **Message**  **Type** | | **ID** | **Payload Data Frame** | | | | | | | | **Checksum** | **EOF** |
| **0xA5**  **0xA5** | 0x0C | SRC ID | MSG Type | $ID | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |  | **0x5A**  **0x5A** |
| 1 | 0 | **Data**  **length** | **Neg Resp** | **SID** | **NRC** | **Data** | **Data** | **Data** | **Data** | CRC8 |
| 0x03 | 0x7F | 0x22 | XX | 0x00 | 0x00 | 0x00 | 0x00 |  |

Table:3 Read data by identifier positive response frame format.

**XX – Negative response code.**

### Read data by identifier Service Negative Response codes (NRC’s):

**SRS-ASW-22H-005**

|  |  |  |
| --- | --- | --- |
| **Tags** | **NRC** | **Description** |
| **SRS-ASW-22H-006** | **0x13** | Invalid message length |
| **SRS-ASW-22H-007** | **0x31** | Request out of range |
| **SRS-ASW-22H-008** | **0x33** | Security access denied |
| **SRS-ASW-22H-009** | **0x22** | Condition not correct |
| **SRS-ASW-22H-010** | **0x14** | Response length exceed |

## Write Memory by Address (0X3D):

**SRS-ASW-3DH-001**

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.

The format and definition of the data Record shall be vehicle manufacturer specific and may or may not be secured. It is the vehicle manufacturer's responsibility to assure that the server conditions are met when performing this service. Possible uses for this service are:

— clear non-volatile memory.

— change calibration values.

### Write Memory by Address Request Frame Format:

**SRS-ASW-3DH-002**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SOF** | **Frame Length** | **Message**  **Type** | | **ID** | **Payload Data Frame** | | | | | | | | **Checksum** | **EOF** | |
| **0xA5**  **0xA5** | 0x0C | SRC ID | MSG Type | $ID | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |  | **0x5A**  **0x5A** |
| 0 | 0 | **First**  **Frame** | **Data Length** | **SID**  **REQ** | **Add**  **Len** | **Mem**  **Addr** | **Mem**  **Addr** | **Mem**  **Addr** | **Mem**  **Addr** | CRC8 |
| 0x10 | $ZZ | 0x3D | 0x24 | XX | XX | XX | XX |  |

Table:1 Write memory by address request frame format.

$ZZ – Data Length will be varied based on size.

ADD Len- Address Length Format identifier.

XX – Memory Address.

**Flow control Response frame:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SOF** | **Frame Length** | **Message**  **Type** | | **ID** | **Payload Data Frame** | | | | | | | | **Checksum** | **EOF** |
| **0xA5**  **0xA5** | 0x0C | SRC ID | MSG Type | $ID | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |  | **0x5A**  **0x5A** |
| 1 | 0 | **FC**  **Frame** | **Data** | **Data** | **Data** | **Data** | **Data** | **Data** | **Data** | CRC8 |
| 0x30 | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 |  |

**Consecutive Request frame:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SOF** | **Frame Length** | **Message**  **Type** | | **ID** | **Payload Data Frame** | | | | | | | | **Checksum** | **EOF** |
| **0xA5**  **0xA5** | 0x0C | SRC ID | MSG Type | $ID | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |  | **0x5A**  **0x5A** |
| 0 | 0 | **CS**  **Frame** | **Mem**  **Size** | **Mem**  **Size** | **Data** | **Data** | **Data** | **Data** | **Data** | CRC8 |
| 0x2S | ZZ | ZZ | YY | YY | YY | YY | YY |  |

**ZZ-** Number of bytes write in the Flash memory

**YY-** Data Send from Client to Server**.**

### Write Memory by Address Positive Response Frame Format:

**SRS-ASW-3DH-003**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SOF** | **Frame Length** | **Message**  **Type** | | **ID** | **Payload Data Frame** | | | | | | | | **Checksum** | **EOF** |
| **0xA5**  **0xA5** | 0x0C | SRC ID | MSG Type | $ID | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |  | **0x5A**  **0x5A** |
| 1 | 0 | **First**  **Frame** | **Data Length** | **SID**  **RESP** | **Add**  **Len** | **Mem**  **Addr** | **Mem**  **Addr** | **Mem**  **Addr** | **Mem**  **Addr** | CRC8 |
| 0x10 | $YY | 0x7D | 0x24 | XX | XX | XX | XX |  |

Table:2 Write memory by address positive Response frame format.

**XX –** Memory Starting Address.

**YY**- Number of bytes write in Memory (0x08).

**0x24 –** Address Length Format Identifier.

**Flow control Response frame:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SOF** | **Frame Length** | **Message**  **Type** | | **ID** | **Payload Data Frame** | | | | | | | | **Checksum** | **EOF** |
| **0xA5**  **0xA5** | 0x0C | SRC ID | MSG Type | $ID | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |  | **0x5A**  **0x5A** |
| 0 | 0 | **FC**  **Frame** | **Data** | **Data** | **Data** | **Data** | **Data** | **Data** | **Data** | CRC8 |
| 0x30 | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 |  |

**Consecutive Request frame:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SOF** | **Frame Length** | **Message**  **Type** | | **ID** | **Payload Data Frame** | | | | | | | | **Checksum** | **EOF** |
| **0xA5**  **0xA5** | 0x0C | SRC ID | MSG Type | $ID | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |  | **0x5A**  **0x5A** |
| 1 | 0 | **CS**  **Frame** | **Mem**  **Size** | **Mem**  **Size** | **Data** | **Data** | **Data** | **Data** | **Data** | CRC8 |
| 0x2S | ZZ | ZZ | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 |  |

**S** - Sequence Counter.

**ZZ-** Number of bytes write in the Flash memory

### Write Memory by Address Negative Response Frame Format:

**SRS-ASW-3DH-004**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SOF** | **Frame Length** | **Message**  **Type** | | **ID** | **Payload Data Frame** | | | | | | | | **Checksum** | **EOF** |
| **0xA5**  **0xA5** | 0x0C | SRC ID | MSG Type | $ID | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |  | **0x5A**  **0x5A** |
| 1 | 0 | **Data**  **length** | **Neg Resp** | **SID** | **NRC** | **Data** | **Data** | **Data** | **Data** | CRC8 |
| 0x03 | 0x7F | 0x3D | XX | 0x00 | 0x00 | 0x00 | 0x00 |  |

Table:3 Write memory by address Negative Response frame format.

**XX-**Negative Response code**.**

### Write Memory by Address Supported Negative Response Codes (NRC’s):

**SRS-ASW-3DH-005**

|  |  |  |
| --- | --- | --- |
| **Tags** | **NRC** | **Description** |
| **SRS-ASW-3DH-006** | 0X31 | Request Out of Range. |
| **SRS-ASW-3DH-007** | 0x13 | Incorrect Message length or Invalid Format. |
| **SRS-ASW-3DH-008** | 0x22 | Condition Not Correct. |
| **SRS-ASW-3DH-009** | 0x72 | General Program Failure. |
| **SRS-ASW-3DH-010** | 0x33 | Security Access Denied. |
| **SRS-ASW-3DH-011** | 0x7F | Service not support active session. |

Table: 4 Write memory by address Negative Response code.

## Communication Control service(0x28):

**SRS-ASW-28H-001**

The purpose of this service is to switch on/off the transmission and/or the reception of certain messages of (a) server(s) (eg: application communication messages).

The server shall still send a positive response if the service is supported in the active session with a requested Sub Function even if the requested Sub Function state is already active.

IMPORTANT — The server and the client shall meet the request and response message.

### Communication Control Request Frame Format:

**SRS-ASW-28H-002**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SOF** | **Frame Length** | **Message**  **Type** | | **ID** | **Payload Data Frame** | | | | | | | | **Checksum** | **EOF** |
| **0xA5**  **0xA5** | 0x0C | SRC ID | MSG Type | $ID | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |  | **0x5A**  **0x5A** |
| 0 | 0 | **Data**  **length** | **SID**  **REQ** | **Sub**  **ID** | **Com**  **Type** | **Data** | **Data** | **Data** | **Data** | CRC8 |
| 0x03 | 0x28 | XX | YY | 0x00 | 0x00 | 0x00 | 0x00 |  |

Table:1 Communication Control Service Request Frame Format

**XX-**Sub Function

**YY-** Communication type

**SF-REQ-XX: -**

1. **Enable Rx and Tx (0x00)**: **SRS-ASW-28H-003**

This value indicates that the reception and transmission of messages shall be enabled for the specified communication Type.

1. **Enable Rx and Disable Tx (0x01): SRS-ASW-28H-004**

This value indicates that the reception of messages shall be enabled and the transmission shall be disabled for the specified communication Type.

1. **Disable Rx and Enable Tx (0x02): SRS-ASW-28H-005**

This value indicates that the reception of messages shall be disabled and the transmission shall be enabled for the specified communication Type.

1. **Disable Rx and Tx(0x03): SRS-ASW-28H-006**

This value indicates that the reception and transmission of messages shall be disabled for the specified communication Type.

**COMM type – YY: -**

1. **Normal Communication Message(0x01): SRS-ASW-28H-007**

This value references all application-related communication (inter-application signal exchange between multiple in-vehicle servers).

1. **Network Management Communication Message(0x02): SRS-ASW-28H-008**

This value references all network management related communication.

1. **Network Management Communication Message and Normal Communication Message (0x03): SRS-ASW-28H-009**

This value references all network management and application-related communication.

### Communication Control Positive Response Frame Format:

**SRS-ASW-28H-010**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SOF** | **Frame Length** | **Message**  **Type** | | **ID** | **Payload Data Frame** | | | | | | | | **Checksum** | **EOF** |
| **0xA5**  **0xA5** | 0x0C | SRC ID | MSG Type | $ID | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |  | **0x5A**  **0x5A** |
| 1 | 0 | **Data**  **length** | **SID**  **RESP** | **Sub**  **ID** | **Data** | **Data** | **Data** | **Data** | **Data** | CRC8 |
| 0x02 | 0x68 | XX | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 |  |

Table:2 Communication Control Service Positive Response Frame Format

**XX-Sub Function**

### Communication Control Negative Response Frame Format:

**SRS-ASW-28H-011**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SOF** | **Frame Length** | **Message**  **Type** | | **ID** | **Payload Data Frame** | | | | | | | | **Checksum** | **EOF** |
| **0xA5**  **0xA5** | 0x0C | SRC ID | MSG Type | $ID | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |  | **0x5A**  **0x5A** |
| 1 | 0 | **Data**  **length** | **Neg Resp** | **SID** | **NRC** | **Data** | **Data** | **Data** | **Data** | CRC8 |
| 0x03 | 0x7F | 0x28 | XX | 0x00 | 0x00 | 0x00 | 0x00 |  |

Table:3 Communication Control Service Negative Response Frame Format

### Communication Control Negative Response Code:

**SRS-ASW-28H-012**

|  |  |  |
| --- | --- | --- |
| **Tags** | **NRC** | **Description** |
| **SRS-ASW-28H-013** | 0x12 | Sub Function Not Supported |
| **SRS-ASW-28H-014** | 0x13 | Incorrect Message Length |
| **SRS-ASW-28H-015** | 0x11 | Service Not Supported |
| **SRS-ASW-28H-016** | 0x7F | Service not supported in active session |
| **SRS-ASW-28H-017** | 0x31 | Request Out of Range |

## Write Data by Identifier (0x2E):

**SRS-ASW-2EH-001**

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.

### Write Data by Identifier Request Frame Format:

**SRS-ASW-2EH-002**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SOF** | **Frame Length** | **Message**  **Type** | | **ID** | **Payload Data Frame** | | | | | | | | **Checksum** | **EOF** |
| **0xA5**  **0xA5** | 0x0C | SRC ID | MSG Type | $ID | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |  | **0x5A**  **0x5A** |
| 0 | 0 | **Data**  **length** | **SID**  **REQ** | **DID** | **DID** | **Data** | **Data** | **Data** | **Data** | CRC8 |
| YY | 0x2E | XX | XX | RD | RD | RD | RD |  |

Table:1 Write Data Identifier Request Frame Format.

* **DID**: Data Identifier.

This parameter identifies the server data record(s) that are being requested by the client.

* **RD**: Record Data (Clint sends to server).

### Write Data by Identifier Positive Response Frame Format:

**SRS-ASW-2EH-003**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SOF** | **Frame Length** | **Message**  **Type** | | **ID** | **Payload Data Frame** | | | | | | | | **Checksum** | **EOF** |
| **0xA5**  **0xA5** | 0x0C | SRC ID | MSG Type | $ID | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |  | **0x5A**  **0x5A** |
| 1 | 0 | **Data**  **length** | **SID**  **RESP** | **DID** | **DID** | **Data** | **Data** | **Data** | **Data** | CRC8 |
| 0x03 | 0x6E | XX | XX | 0x00 | 0x00 | 0x00 | 0x00 |  |

Table:2 Write Data Identifier Positive Response Frame Format

### Write Data by Identifier Negative Response Frame Format:

**SRS-ASW-2EH-004**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SOF** | **Frame Length** | **Message**  **Type** | | **ID** | **Payload Data Frame** | | | | | | | | **Checksum** | **EOF** |
| **0xA5**  **0xA5** | 0x0C | SRC ID | MSG Type | $ID | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |  | **0x5A**  **0x5A** |
| 1 | 0 | **Data**  **length** | **Neg Resp** | **SID** | **NRC** | **Data** | **Data** | **Data** | **Data** | CRC8 |
| 0x03 | 0x7F | 0x2E | XX | 0x00 | 0x00 | 0x00 | 0x00 |  |

Table:3 Write Data Identifier Positive Response Frame Format

XX-Negative Response Code.

### Write Memory by Address Negative Response Codes:

**SRS-ASW-2EH-005**

|  |  |  |
| --- | --- | --- |
| Tags | NRC | Description |
| **SRS-ASW-2EH-006** | 0x72 | General programming failure |
| **SRS-ASW-2EH-007** | 0x13 | Invalid message length |
| **SRS-ASW-2EH-008** | 0x22 | Condition not correct |
| **SRS-ASW-2EH-009** | 0x31 | Request out of range |
| **SRS-ASW-2EH-010** | 0x33 | Security Access Denied |
| **SRS-ASW-2EH-011** | 0x7F | Service not support in active session |

## Tester Present service (0x3E):

**SRS-ASW-3EH-001**

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. This service is used to keep one or multiple servers in a diagnostic session other than the default Session. This can either be done by transmitting the Tester Present request message periodically or in case of the absence of other diagnostic services to prevent the server(s) from automatically returning to the default Session.

### Tester Present Request Frame Format:

**SRS-ASW-3EH-002**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SOF** | **Frame Length** | **Message**  **Type** | | **ID** | **Payload Data Frame** | | | | | | | | **Checksum** | **EOF** |
| **0xA5**  **0xA5** | 0x0C | SRC ID | MSG Type | $ID | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |  | **0x5A**  **0x5A** |
| 0 | 0 | **Data**  **length** | **SID**  **REQ** | **Sub**  **ID** | **Data** | **Data** | **Data** | **Data** | **Data** | CRC8 |
| 0x02 | 0x3E | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 |  |

Table:1 Tester Present Request Frame Format

**SF REQ –**

1. **Zero Sub Function(0x00):**

**-**This parameter value is used to indicate that no Sub Function value beside the suppressPosRspMsg Indication Bit is supported by this service.

### Tester Present Positive Response Frame Format:

**SRS-ASW-3EH-003**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SOF** | **Frame Length** | **Message**  **Type** | | **ID** | **Payload Data Frame** | | | | | | | | **Checksum** | **EOF** |
| **0xA5**  **0xA5** | 0x0C | SRC ID | MSG Type | $ID | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |  | **0x5A**  **0x5A** |
| 1 | 0 | **Data**  **length** | **SID**  **RESP** | **Sub**  **ID** | **Data** | **Data** | **Data** | **Data** | **Data** | CRC8 |
| 0x02 | 0x7E | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 |  |

### Tester Present Negative Response Frame Format:

**SRS-ASW-3EH-004**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SOF** | **Frame Length** | **Message**  **Type** | | **ID** | **Payload Data Frame** | | | | | | | | **Checksum** | **EOF** |
| **0xA5**  **0xA5** | 0x0C | SRC ID | MSG Type | $ID | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |  | **0x5A**  **0x5A** |
| 1 | 0 | **Data**  **length** | **Neg Resp** | **SID** | **NRC** | **Data** | **Data** | **Data** | **Data** | CRC8 |
| 0x03 | 0x7F | 0x3E | XX | 0x00 | 0x00 | 0x00 | 0x00 |  |

Table:2 Tester Present Request Frame Format

XX-Negative Response code.

### Tester Present Negative Response Codes (NRC’s)

|  |  |  |
| --- | --- | --- |
| **Tags** | **NRC** | **Description** |
| **SRS-ASW-3EH-005** | 0x12 | Sub Function Not Supported |
| **SRS-ASW-3EH-006** | 0x13 | Incorrect Message Length or Invalid Format |

## Read Memory by Address (0x23):

**SRS-ASW-23H-01**

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. The server sends data record values via the Read Memory By Address positive response message. The format and definition of the data record parameter shall be vehicle manufacturer specific. The data record parameter may include analog input and output signals, digital input and output signals, internal data and system status information if supported by the server.

### Read memory by Address Request Frame Format:

**SRS-ASW-23H-02**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SOF** | **Frame Length** | **Message**  **Type** | | **ID** | **Payload Data Frame** | | | | | | | | **Checksum** | **EOF** |
| **0xA5**  **0xA5** | 0x0C | SRC ID | MSG Type | $ID | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |  | **0x5A**  **0x5A** |
| 0 | 0 | **First**  **Frame** | **Data Length** | **SID**  **REQ** | **Add**  **Len** | **Mem**  **Addr** | **Mem**  **Addr** | **Mem**  **Addr** | **Mem**  **Addr** | CRC8 |
| 0x10 | 0x08 | 0x23 | 0x24 | AA | AA | AA | AA |  |

**AA:** Memory starting address.

**Flow Control from the Server to client:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SOF** | **Frame Length** | **Message**  **Type** | | **ID** | **Payload Data Frame** | | | | | | | | **Checksum** | **EOF** |
| **0xA5**  **0xA5** | 0x0C | SRC ID | MSG Type | $ID | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |  | **0x5A**  **0x5A** |
| 1 | 0 | **FC**  **Frame** | **Data** | **Data** | **Data** | **Data** | **Data** | **Data** | **Data** | CRC8 |
| 0x30 | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 |  |

**Consecutive Frame:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SOF** | **Frame Length** | **Message**  **Type** | | **ID** | **Payload Data Frame** | | | | | | | | **Checksum** | **EOF** |
| **0xA5**  **0xA5** | 0x0C | SRC ID | MSG Type | $ID | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |  | **0x5A**  **0x5A** |
| 0 | 0 | **CS**  **Frame** | **Mem**  **Size** | **Mem**  **Size** | **Data** | **Data** | **Data** | **Data** | **Data** | CRC8 |
| 0x2S | YY | YY | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 |  |

**YY:** Memory size.

### Read memory By Address Positive Response Frame Format:

**SRS-ASW-23H-03**

**First Frame from Server to Client:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SOF** | **Frame Length** | **Message**  **Type** | | **ID** | **Payload Data Frame** | | | | | | | | **Checksum** | **EOF** |
| **0xA5**  **0xA5** | 0x0C | SRC ID | MSG Type | $ID | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |  | **0x5A**  **0x5A** |
| 1 | 0 | **First**  **Frame** | **Data Length** | **SID**  **RESP** | **Data** | **Data** | **Data** | **Data** | **Data** | CRC8 |
| 0x10 | $YY | 0x63 | XX | XX | XX | XX | XX |  |

**XX:** Data record.

**$YY**: Data length depends on the data record.

**Flow Control Frame:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SOF** | **Frame Length** | **Message**  **Type** | | **ID** | **Payload Data Frame** | | | | | | | | **Checksum** | **EOF** |
| **0xA5**  **0xA5** | 0x0C | SRC ID | MSG Type | $ID | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |  | **0x5A**  **0x5A** |
| 0 | 0 | **FC**  **Frame** | **Data** | **Data** | **Data** | **Data** | **Data** | **Data** | **Data** | CRC8 |
| 0x30 | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 |  |

**Consecutive Frame:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SOF** | **Frame Length** | **Message**  **Type** | | **ID** | **Payload Data Frame** | | | | | | | | **Checksum** | **EOF** |
| **0xA5**  **0xA5** | 0x0C | SRC ID | MSG Type | $ID | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |  | **0x5A**  **0x5A** |
| 1 | 0 | **CS**  **Frame** | **Data** | **Data** | **Data** | **Data** | **Data** | **Data** | **Data** | CRC8 |
| 0x2S | XX | XX | XX | XX | XX | XX | XX |  |

**XX:** Data Record.

**S:** Sequence Count.

### Read Memory By Address Negative Response Frame Format:

**SRS-ASW-23H-04**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SOF** | **Frame Length** | **Message**  **Type** | | **ID** | **Payload Data Frame** | | | | | | | | **Checksum** | **EOF** |
| **0xA5**  **0xA5** | 0x0C | SRC ID | MSG Type | $ID | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |  | **0x5A**  **0x5A** |
| 1 | 0 | **Data**  **length** | **Neg Resp** | **SID** | **NRC** | **Data** | **Data** | **Data** | **Data** | CRC8 |
| 0x03 | 0x7F | 0x23 | XX | 0x00 | 0x00 | 0x00 | 0x00 |  |

### Read memory by Address Negative Response Code:

**SRS-ASW-23H-05**

|  |  |  |
| --- | --- | --- |
| **Tags** | **NRC** | **Description** |
| **SRS-ASW-23H-06** | 0x13 | Incorrect Message Length or Invalid format |
| **SRS-ASW-23H-07** | 0x14 | Response Too Long |
| **SRS-ASW-23H-08** | 0x22 | Conditions Not Correct |
| **SRS-ASW-23H-09** | 0x31 | Request Out of Range |
| **SRS-ASW-23H-10** | 0x33 | Security Access Denied |

## Clear Diagnostic Information (0x14):

**SRS-ASW-14H-001**

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 Clear Diagnostic Information service is completely processed. The server shall send a positive response even if no DTCs are stored.

### Clear Diagnostic Information Request Frame Format:

**SRS-ASW-14H-002**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SOF** | **Frame Length** | **Message**  **Type** | | **ID** | **Payload Data Frame** | | | | | | | | **Checksum** | **EOF** |
| **0xA5**  **0xA5** | 0x0C | SRC ID | MSG Type | $ID | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |  | **0x5A**  **0x5A** |
| 0 | 0 | **Data**  **length** | **SID**  **REQ** | **DTC** | **DTC** | **DTC** | **Data** | **Data** | **Data** | CRC8 |
| 0x04 | 0x14 | $XX | $XX | $XX | 0x00 | 0x00 | 0x00 |  |

Table:1 Clear Diagnostic Information Request Frame Format

$XX **– Group of DTC (3 bytes).**

### Clear Diagnostic Information Positive Response Frame Format:

**SRS-ASW-14H-003**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SOF** | **Frame Length** | **Message**  **Type** | | **ID** | **Payload Data Frame** | | | | | | | | **Checksum** | **EOF** |
| **0xA5**  **0xA5** | 0x0C | SRC ID | MSG Type | $ID | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |  | **0x5A**  **0x5A** |
| 1 | 0 | **Data**  **length** | **SID**  **RESP** | **Data** | **Data** | **Data** | **Data** | **Data** | **Data** | CRC8 |
| 0x01 | 0x54 | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 |  |

Table:2 Clear Diagnostic Information Positive Response Frame Format

### Clear Diagnostic Information Negative Response Frame Format:

**SRS-ASW-14H-004**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SOF** | **Frame Length** | **Message**  **Type** | | **ID** | **Payload Data Frame** | | | | | | | | **Checksum** | **EOF** |
| **0xA5**  **0xA5** | 0x0C | SRC ID | MSG Type | $ID | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |  | **0x5A**  **0x5A** |
| 1 | 0 | **Data**  **length** | **Neg Resp** | **SID** | **NRC** | **Data** | **Data** | **Data** | **Data** | CRC8 |
| 0x03 | 0x7F | 0x14 | XX | 0x00 | 0x00 | 0x00 | 0x00 |  |

Table:3 Clear Diagnostic Information Negative Response Frame Format.

**$XX –** Negative Response Code.

### Clear Diagnostic Information Negative Response Codes (NRC’s):

**SRS-ASW-14H-00**5

|  |  |  |
| --- | --- | --- |
| **Tags** | **NRC** | **Description** |
| **SRS-ASW-14H-006** | **0x13** | Incorrect Message length or Invalid Format |
| **SRS-ASW-14H-007** | **0x22** | Condition Not Correct |
| **SRS-ASW-14H-008** | **0x31** | Request Out of Range |
| **SRS-ASW-14H-009** | **0x72** | General Programming Failure |

Table:4 Clear Diagnostic Information Negative Response Code.

## Read DTC Information (0x19):

**SRS-ASW-19H-001**

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.

### Read DTC Information Request Frame Format:

**SRS-ASW-19H-002**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SOF** | **Frame Length** | **Message**  **Type** | | **ID** | **Payload Data Frame** | | | | | | | | **Checksum** | **EOF** |
| **0xA5**  **0xA5** | 0x0C | SRC ID | MSG Type | $ID | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |  | **0x5A**  **0x5A** |
| 0 | 0 | **Data**  **length** | **SID**  **REQ** | **Sub ID(n)** | **Data** | **Data** | **Data** | **Data** | **Data**  **(m)** | CRC8 |
| 2+n+m | 0x19 | $YY | $ZZ | $ZZ | $ZZ | $ZZ | $ZZ |  |

Table:1 Read DTC Information Request Frame Format

**$ZZ: Data Parameters (It will be changed for every subfunction)**

**$YY:** Sub function ID

* **Report Number of DTC By Status Mask (0x01) SRS-ASW-19H-003**

A client can retrieve a count of the number of DTCs matching a client defined status mask by sending a request for this service with the Sub Function set to report Number of DTC By Status Mask.

* **Report DTC By Status Mask(0x02) SRS-ASW-19H-004**

The client can retrieve a list of DTCs, which satisfy a client defined status mask by sending a request with the Sub-Function byte set to report DTC By Status Mask. This Sub-Function allows the client to request the server to report all DTCs that are “test Failed” OR “confirmed” OR “etc.”.

* **Report DTC Snapshot record identification (0x03) SRS-ASW-19H-005**

A client can retrieve DTC Snapshot record identification information for all captured DTC Snapshot records by sending a request for this service with the Sub-Function set to report DTC Snapshot Identification.

* **Report DTC Snapshot Record by DTC Number (0x04) SRS-ASW-19H-006**

A client can retrieve captured DTC Snapshot record data for a client defined DTC Mask Record in conjunction with a DTC Snapshot record number by sending a request for this service with the Sub-Function set to report DTC Snapshot Record by DTC Number.

* **Report Number of DTC By Severity Mask Record (0x07) SRS-ASW-19H-007**

A client can retrieve a count of the number of DTCs matching a client defined severity status mask record by sending a request for this service with the Sub-Function set to report Number Of DTC By Severity Mask Record.

* **Report DTC By Severity Mask Record (0x08) SRS-ASW-19H-008**

The client can retrieve a list of DTC severity and functional unit information, which satisfy a client defined severity mask record by sending a request with the Sub-Function byte set to report DTC By Severity Mask Record.

* **Report Severity Information Of DTC(0x09) SRS-ASW-19H-009**

A client can retrieve severity and functional unit information for a client defined DTC Mask Record by sending a request for this service with the Sub-Function set to report Severity Information Of DTC.

* **Report Supported DTC (0x0A) SRS-ASW-19H-010**

A client can retrieve the status of all DTCs supported by the server by sending a request for this service with the Sub-Function set to report Supported DTCs.

* **Report First Test Failed DTC (0x0B) SRS-ASW-19H-011**

The client can retrieve the first failed DTC from the server by sending a request with the Sub-Function byte set to “report First Test Failed DTC”.

* **Report First Confirmed DTC (0x0C) SRS-ASW-19H-012**

The client can retrieve the first confirmed DTC from the server by sending a request with the Sub-Function byte set to “report First Confirmed DTC”.

* **Report DTC Fault Detection Counter(0x14) SRS-ASW-19H-013**

The client can retrieve a list of all current "pre-failed" DTCs which have or have not yet been detected as "pending" or "confirmed" at the time of the client's request. The intention of the DTC Fault Detection Counter is a simple method to identify a growing or intermittent problem which cannot be identified/read by the status Of DTC byte of a particular DTC.

* **Report DTC With Permanent Status(0x15) SRS-ASW-19H-014**

The Sub-Function 15 will be replaced by Sub-Function 5516 in the future. In case there is a need for Permanent DTC implementation it is recommended to use the 5516 Sub-Function.

### Read DTC Information Positive Response Frame Format:

**SRS-ASW-19H-015**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SOF** | **Frame Length** | **Message**  **Type** | | **ID** | **Payload Data Frame** | | | | | | | | **Checksum** | **EOF** |
| **0xA5**  **0xA5** | 0x0C | SRC ID | MSG Type | $ID | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |  | **0x5A**  **0x5A** |
| 1 | 0 | **Data**  **length** | **SID**  **RESP** | **Sub ID(n)** | **Data** | **Data** | **Data** | **Data** | **Data** | CRC8 |
| 2+n+m | 0x59 | $YY | $ZZ | $ZZ | $ZZ | $ZZ | $ZZ |  |

Table:2 Read DTC Information Positive Response Frame Format

### Read DTC Information Negative Response Frame Format:

**SRS-ASW-19H-016**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SOF** | **Frame Length** | **Message**  **Type** | | **ID** | **Payload Data Frame** | | | | | | | | **Checksum** | **EOF** |
| **0xA5**  **0xA5** | 0x0C | SRC ID | MSG Type | $ID | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |  | **0x5A**  **0x5A** |
| 1 | 0 | **Data**  **length** | **Neg Resp** | **SID** | **NRC** | **Data** | **Data** | **Data** | **Data** | CRC8 |
| 0x03 | 0x7F | 0x19 | XX | 0x00 | 0x00 | 0x00 | 0x00 |  |

Table:3 Read DTC Information Negative Response Frame Format.

**$XX –** Negative Response Code**.**

### Read DTC Information Negative Response Codes (NRC’s):

**SRS-ASW-19H-017**

|  |  |  |
| --- | --- | --- |
| **Tags** | **NRC** | **Description** |
| **SRS-ASW-19H-018** | **0x13** | Incorrect Message length or Invalid Format |
| **SRS-ASW-19H-019** | **0x12** | Sub Function not supported. |
| **SRS-ASW-19H-020** | **0x31** | Request out of range |

Table:4 Read DTC Information Negative Response Code.

## Input Output Control By Identifier (0x2F) service:

**SRS-ASW-2FH-001**

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. In general, this service is used for relatively simple (e.g. static) input substitution/output control whereas routine Control is used if more complex input substitution/output control is necessary.

The client request message contains a data Identifier to reference the input signal, internal server function, and/or output signal(s). The control Option Record parameter shall include all information required by the server's input signal(s), internal function(s) and/or output signal(s).

The server shall send a positive response message if the request control was successfully started or has reached its desired state. The service allows the control of a single data Identifier and its corresponding parameter(s) in a single request message. Doing so, the server will respond with a single response message including the data Identifier of the request message plus control Status information.

### Input Output Control By Identifier Request Frame Format:

**SRS-ASW-2FH-002**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SOF** | **Frame Length** | **Message**  **Type** | | **ID** | **Payload Data Frame** | | | | | | | | **Checksum** | **EOF** |
| **0xA5**  **0xA5** | 0x0C | SRC ID | MSG Type | $ID | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |  | **0x5A**  **0x5A** |
| 0 | 0 | **Data**  **length** | **SID**  **REQ** | **DID** | **DID** | **COR** | **Data** | **Data** | **Data** | CRC8 |
| YY | 0x2F | $ZZ | $ZZ | $XX | 0x00 | 0x00 | 0x00 |  |

Table:1 Input Output Control by Identifier Request Frame Format

**$ZZ - Data identifiers**

**$YY -Data Length**: Here the length may vary depending upon the COR.

**$XX- Control Option Record:** Input output control parameters definition.

The control Option Record consists of one or multiple bytes (input Output Control Parameter and controlState#1 to control State #m). The control Option Record parameter details shall be implemented as defined below.

1. **Return Control To ECU (0x00): SRS\_ASW\_2FH\_003**

**-** This value shall indicate to the server that the client no longer has control about the input signal(s), internal parameter(s) and/or output signal(s) referenced by the Data Identifier.

1. **Reset To Default (0x01): SRS\_ASW\_2FH\_004**

**-** This value shall indicate to the server that it is requested to reset the input signal(s), internal parameter(s) and/or output signal(s) referenced by the data Identifier to its default state.

1. **Freeze Current State (0x02): SRS\_ASW\_2FH\_005**

**-** his value shall indicate to the server that it is requested to freeze the current state of the input signal(s), internal parameter(s) and/or output signal referenced by the data Identifier.

1. **Short Term Adjustment(0x03): SRS\_ASW\_2FH\_006**

**-** This value shall indicate to the server that it is requested to adjust the input signal(s), internal parameter(s) and/or controlled output signal(s) referenced by the data Identifier in RAM to the value(s) included in the control Option parameter(s) (e.g. set Idle Air Control Valve to a specific step number, set pulse width of valve to a specific value/duty cycle).

### Input Output Control by Identifier Positive Response Frame Format:

**SRS-ASW-2FH\_007**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SOF** | **Frame Length** | **Message**  **Type** | | **ID** | **Payload Data Frame** | | | | | | | | **Checksum** | **EOF** |
| **0xA5**  **0xA5** | 0x0C | SRC ID | MSG Type | $ID | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |  | **0x5A**  **0x5A** |
| 1 | 0 | **Data**  **length** | **SID**  **RESP** | **DID** | **DID** | **COR** | **Data** | **Data** | **Data** | CRC8 |
| YY | 0x6F | $ZZ | $ZZ | $XX | $RR | $RR | $RR |  |

Table:2 Input Output Control by Identifier Positive Response Frame Format

**$ZZ -**Data identifiers.

**$YY-**Data Length.

**$XX-**Control Status Record.

**$RR-** Data Record**.**

### Input Output Control by Identifier Negative Response Frame Format:

**SRS-ASW-2FH-008**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SOF** | **Frame Length** | **Message**  **Type** | | **ID** | **Payload Data Frame** | | | | | | | | **Checksum** | **EOF** |
| **0xA5**  **0xA5** | 0x0C | SRC ID | MSG Type | $ID | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |  | **0x5A**  **0x5A** |
| 1 | 0 | **Data**  **length** | **Neg Resp** | **SID** | **NRC** | **Data** | **Data** | **Data** | **Data** | CRC8 |
| 0x03 | 0x7F | 0x2F | XX | 0x00 | 0x00 | 0x00 | 0x00 |  |

Table:3 Input Output Control by Identifier Negative Response Frame Format.

### Input Output Control by Identifier negative response code (NRC):

|  |  |  |
| --- | --- | --- |
| **Tags** | **NRC** | **Description** |
| **SRS-ASW-2FH-009** | 0x22 | Condition not correct |
| **SRS-ASW-2FH-010** | 0x13 | Incorrect Message Length or Invalid Format |
| **SRS-ASW-2FH-011** | 0x31 | Request Out of range |
| **SRS-ASW-2FH-012** | 0x7F | Service not supported in active Session. |
| **SRS-ASW-2FH-013** | 0x33 | Security access denied |
| **SRS-ASW-2FH-014** | 0x34 | Authentication required |