**CAN UDS**

UDS - BSW Specification and Requirements

**Version 1.0**

Contents

[1. Introduction 6](#_Toc141888680)

[1.1 Abbreviations and Acronyms 6](#_Toc141888681)

[1.2 Reference Document 6](#_Toc141888682)

[1.3 Diagnostic Testing and Programming Station Set Up 6](#_Toc141888683)

[1.4 Supported CAN ID’s 6](#_Toc141888684)

[1.5 Description of Services for Application 7](#_Toc141888685)

[1.6 Service Supported in Different Sessions and Addressing 7](#_Toc141888686)

[2. BSW SERVICES 8](#_Toc141888687)

[2.1 Diagnostic Session Control (1016) service 8](#_Toc141888688)

[2.1.1 Diagnostic Session Control Request Frame Format 8](#_Toc141888689)

[2.1.2 Diagnostic Session Control Positive Response Frame Format 8](#_Toc141888690)

[2.1.3 Diagnostic Session Control Negative Response Frame Format 9](#_Toc141888691)

[2.1.4 Diagnostic Session Control Supported Negative Response Codes (NRC’s) 9](#_Toc141888692)

[2.2 ECU Reset (1116) service 10](#_Toc141888693)

[2.2.1 Request Frame Format 10](#_Toc141888694)

[2.2.2 Positive Response Frame Format 10](#_Toc141888695)

[2.2.3 Negative Response Frame Format 11](#_Toc141888696)

[2.2.4 Supported Negative Response Codes (NRC’s) 11](#_Toc141888697)

[2.3 Security Access (2716) service 12](#_Toc141888698)

[2.3.1 Request Frame Format 12](#_Toc141888699)

[2.3.2 Positive Response Frame Format 12](#_Toc141888700)

[2.3.3 Negative Response Frame Format 13](#_Toc141888701)

[2.2.4 Supported Negative Response Codes (NRC’s) 13](#_Toc141888702)

[2.4 Tester Present (3E16) service 14](#_Toc141888703)

[2.4.1 Request Frame Format 14](#_Toc141888704)

[2.4.2 Positive Response Frame Format 14](#_Toc141888705)

[2.4.3 Negative Response Frame Format 14](#_Toc141888706)

[2.4.4 Supported Negative Response Codes (NRC’s) 15](#_Toc141888707)

[2.5 Routine Control (3116) service 16](#_Toc141888708)

[2.5.1 Start Routine (0x01) - Routine Identifier (RID) – ERASE (0xFF00) 16](#_Toc141888709)

[2.5.1.1 Request Frame Format 16](#_Toc141888710)

[2.5.1.2 Positive Response Frame Format 17](#_Toc141888711)

[2.5.2 Start Routine (0x01) - Routine Identifier (RID) – CHECKSUM (0xFF01) 17](#_Toc141888712)

[2.5.2.1 Request Frame Format 17](#_Toc141888713)

[2.5.2.2 Positive Response Frame Format 17](#_Toc141888714)

[2.5.3 Stop Routine (0x02): 18](#_Toc141888715)

[2.5.3.1 Request Frame Format 18](#_Toc141888716)

[2.5.3.2 Positive Response Frame Format 18](#_Toc141888717)

[2.5.4 Request Routine Results (0x03): 18](#_Toc141888718)

[2.5.4.1 Request Frame Format 18](#_Toc141888719)

[2.5.4.2 Positive Response Frame Format 18](#_Toc141888720)

[2.5.5 Negative Response Frame Format 19](#_Toc141888721)

[2.5.6 Supported Negative Response Codes (NRC’s) 19](#_Toc141888722)

[2.6 Request Download (3416) service 20](#_Toc141888723)

[2.6.1 Request Frame Format 20](#_Toc141888724)

[2.6.2 Positive Response Frame Format 20](#_Toc141888725)

[2.6.3 Negative Response Frame Format 21](#_Toc141888726)

[2.6.4 Supported Negative Response Codes (NRC’s) 21](#_Toc141888727)

[2.7 Request Upload (3516) service 22](#_Toc141888728)

[2.7.1 Request Frame Format 22](#_Toc141888729)

[2.7.2 Positive Response Frame Format 22](#_Toc141888730)

[2.7.3 Negative Response Frame Format 23](#_Toc141888731)

[2.7.4 Supported Negative Response Codes (NRC’s) 23](#_Toc141888732)

[2.8 Transfer Data (3616) service 24](#_Toc141888733)

[2.8.1 If it is Request Download (Service 34): 24](#_Toc141888734)

[2.8.1.1 Request Frame Format 24](#_Toc141888735)

[2.8.1.2 Positive Response Frame Format 25](#_Toc141888736)

[2.8.2 If it is Request Upload (Service 35): 25](#_Toc141888737)

[2.8.2.1 Request Frame Format 25](#_Toc141888738)

[2.8.2.2 Positive Response Frame Format 25](#_Toc141888739)

[2.8.3 Negative Response Frame Format 26](#_Toc141888740)

[2.8.4 Supported Negative Response Codes (NRC’s) 26](#_Toc141888741)

[2.9 Request Transfer Exit (3716) service 27](#_Toc141888742)

[2.9.1 Request Frame Format 27](#_Toc141888743)

[2.9.2 Positive Response Frame Format 27](#_Toc141888744)

[2.9.3 Negative Response Frame Format 27](#_Toc141888745)

[2.9.4 Supported Negative Response Codes (NRC’s) 27](#_Toc141888746)

Tables

[*Table 1: Abbreviation* 6](#_Toc141888781)

[*Table 2: Supported CAN Id* 6](#_Toc141888782)

[*Table 3: Services Allowed During Default and Non-Default Diagnostic Session* 7](#_Toc141888783)

**Foreword**

The document has been drafted based on ISO 14229/ISO 15765-3 which contains the Re-programming and diagnostic testing requirements for RE CAN 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 | 10-12-2021 | Sandeep K Y | Hareesha | Muttanna |
| 1.1 | Updated Service for MPU data Transfer | 12-02-2022 | Sandeep K Y | Hareesha | Muttanna |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

# Introduction

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

## Abbreviations and Acronyms

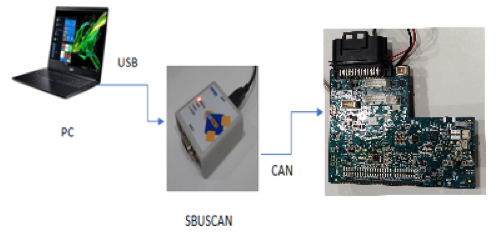
| **Acronyms / Definitions / Terms** | **Description** |
| --- | --- |
| ECU | Electronic Control Unit |
| UDS | Unified Diagnostic Services |
| ASW | Application Software |
| BSW | Bootloader Software |
| CAN | Controller Area Network |
| PID | Physical CAN ID |
| FID | Functional CAN ID |

*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. Programming is done through CAN using the services according to UDS protocol (ISO14229).

## Supported CAN ID’s

|  |  |
| --- | --- |
| **CAN Message Direction** | **ID** |
| Request ID (Physical) | 0x7F0 |
| Request ID (Functional) | N/A |
| Response ID (Physical) | 0x7F1 |

*Table 2: Supported CAN Id*

## Description of Services for Application

A Tester 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 | Tester Present | 0x3E | 🗷 | 🗷 | 🗹 | 🗹 |
| 5 | Routine  control | Routine control | 0x31 | 🗹 | 🗷 | 🗷 | 🗹 |
| 6 | Upload and download | Request Download | 0x34 | 🗹 | 🗷 | 🗷 | 🗹 |
| 7 | Request Upload | 0x35 | 🗹 | 🗷 | 🗷 | 🗹 |
| 8 | Transfer Data | 0x36 | 🗹 | 🗷 | 🗷 | 🗹 |
| 9 | Request Transfer Exit | 0x37 | 🗹 | 🗷 | 🗷 | 🗹 |

*Table 3: Services Allowed During Default and Non-Default Diagnostic Session*

# BSW SERVICES

## Diagnostic Session Control (1016) service

**SRS-BSW-10H-01**

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).

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.

### 2.1.1 Diagnostic Session Control Request Frame Format

**SRS-BSW-10H-02**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Client → Server** | **ID** | **DLC** | **CAN Data Field** | | | | | | | |
| **Physical Request** | **0x7F0** | **8 Byte** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| **Data Length** | **SID**  **REQ** | **SF**  **REQ** | **Data** | **Data** | **Data** | **Data** | **Data** |
| **0x02** | **0x10** | **XX** | **0x00** | **0x00** | **0x00** | **0x00** | **0x00** |

**SF-REQ-XX: -**

1. **Default Session (0x01): SRS-BSW-10H-03**

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

1. **Programming Session(0x02): SRS-BSW-10H-04**

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

1. **Extended Diagnostic Session(0x03): SRS-BSW-10H-05:**

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

### 2.1.2 Diagnostic Session Control Positive Response Frame Format

**SRS-BSW-10H-06**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Server → Client** | **ID** | **DLC** | **CAN Data Field** | | | | | | | |
| **Physical Request** | **0x7F1** | **8 Byte** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| **Data Length** | **SID**  **REQ** | **SF**  **REQ** | **Data** | **Data** | **Data** | **Data** | **Data** |
| **0x02** | **0x50** | **XX** | **0x00** | **0x00** | **0x00** | **0x00** | **0x00** |

### 2.1.3 Diagnostic Session Control Negative Response Frame Format

**SRS-BSW-10H-07**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Server → Client** | **ID** | **DLC** | **CAN Data Field** | | | | | | | |
| **Physical Request** | **0x7F1** | **8 Byte** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| **Data Length** | **Neg**  **Response** | **SID**  **REQ** | **NRC** | **Data** | **Data** | **Data** | **Data** |
| **0x03** | **0x7F** | **0x10** | **XX** | **0x00** | **0x00** | **0x00** | **0x00** |

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

|  |  |  |
| --- | --- | --- |
| **Tags** | **NRC** | **Description** |
| **SRS-BSW-10H-08** | **0x12** | **Sub Function Not Supported** |
| **SRS-BSW-10H-09** | **0x13** | **Incorrect Message Length or Invalid Format** |
| **SRS-BSW-10H-10** | **0x22** | **Conditions Not Correct** |
| **SRS-BSW-10H-11** | **0x7F** | **Service No Support In Active Session** |
| **SRS-BSW-10H-12** | **0x11** | **Service Not Supported** |

## ECU Reset (1116) service

**SRS-BSW-11H-01**

The ECU Reset service is used by the client to request a server reset.

The main purpose of ECU reset is to recover the malfunctioned ECU from its non-working condition or hanged state or from any non-working condition but it should be able to communicate with the external diagnostic computer.

### 2.2.1 Request Frame Format

**SRS-BSW-11H-02**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Client → Server** | **ID** | **DLC** | **CAN Data Field** | | | | | | | |
| **Physical Request** | **0x7F0** | **8 Byte** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| **Data Length** | **SID**  **REQ** | **SF**  **REQ** | **Data** | **Data** | **Data** | **Data** | **Data** |
| **0x02** | **0x11** | **XX** | **0x00** | **0x00** | **0x00** | **0x00** | **0x00** |

**SF-REQ-XX: -**

1. **Hard Reset (0x01): SRS-BSW-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). It might result in the re-initialization of both volatile memory and non-volatile memory locations to predetermined values.

1. **Key Off On Reset (0x02): SRS-BSW -11H-04**

This Sub Function identifies a condition similar to the driver turning the ignition key off and back on. This reset condition should simulate a key-off-on sequence (i.e. interrupting the switched power supply). Typically the values of non-volatile memory locations are preserved; volatile memory will be initialized

1. **Soft Reset (0x03): SRS-BSW-11H-05**

This Sub Function identifies a "soft reset" condition, which causes the server to immediately restart the application program if applicable. A typical action is to restart the application without reinitializing of previously learned configuration data, adaptive factors and other long-term adjustments.

### 2.2.2 Positive Response Frame Format

**SRS-BSW-11H-08**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Server → Client** | **ID** | **DLC** | **CAN Data Field** | | | | | | | |
| **Physical Request** | **0x7F1** | **8 Byte** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| **Data Length** | **SID**  **REQ** | **SF**  **REQ** | **Data** | **Data** | **Data** | **Data** | **Data** |
| **0x02** | **0x51** | **XX** | **0x00** | **0x00** | **0x00** | **0x00** | **0x00** |

### 2.2.3 Negative Response Frame Format

**SRS-BSW-11H-09**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Server → Client** | **ID** | **DLC** | **CAN Data Field** | | | | | | | |
| **Physical Request** | **0x7F1** | **8 Byte** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| **Data Length** | **Neg**  **Response** | **SID**  **REQ** | **NRC** | **Data** | **Data** | **Data** | **Data** |
| **0x03** | **0x7F** | **0x11** | **XX** | **0x00** | **0x00** | **0x00** | **0x00** |

### 2.2.4 Supported Negative Response Codes (NRC’s)

|  |  |  |
| --- | --- | --- |
| **Tags** | **NRC** | **Description** |
| **SRS-BSW-11H-10** | **0x12** | **Sub Function Not Supported** |
| **SRS-BSW-11H-11** | **0x13** | **Incorrect Message Length or Invalid Format** |
| **SRS-BSW-11H-12** | **0x22** | **Conditions Not Correct** |
| **SRS-BSW-11H-13** | **0x33** | **Security Access Denied** |

## Security Access (2716) service

**SRS-BSW-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.

### 2.3.1 Request Frame Format

1. **Request Seed (0x01): SRS-BSW-27H-02**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Client → Server** | **ID** | **DLC** | **CAN Data Field** | | | | | | | |
| **Physical Request** | **0x7F0** | **8 Byte** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| **Data Length** | **SID**  **REQ** | **SF**  **REQ** | **Data** | **Data** | **Data** | **Data** | **Data** |
| **0x02** | **0x27** | **01** | **0x00** | **0x00** | **0x00** | **0x00** | **0x00** |

1. **Send key (0x02): SRS-BSW-27H-03**

Based on the random key, client have to send the keys (4 Byte) with same logic.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Client → Server** | **ID** | **DLC** | **CAN Data Field** | | | | | | | |
| **Physical Request** | **0x7F0** | **8 Byte** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| **Data Length** | **SID**  **REQ** | **SF**  **REQ** | **Data** | **Data** | **Data** | **Data** | **Data** |
| **0x06** | **0x27** | **02** | **YY** | **YY** | **YY** | **YY** | **0x00** |

YY – Send Keys based on received Seed value.

### 2.3.2 Positive Response Frame Format

1. **Request Seed (0x01): SRS-BSW-27H-04**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Server → Client** | **ID** | **DLC** | **CAN Data Field** | | | | | | | |
| **Physical Request** | **0x7F1** | **8 Byte** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| **Data Length** | **SID**  **REQ** | **SF**  **REQ** | **Data** | **Data** | **Data** | **Data** | **Data** |
| **0x06** | **0x67** | **01** | **XX** | **XX** | **XX** | **XX** | **0x00** |

XX – 4 bytes of Random Number Generated by Server.

1. **Send key (0x02): SRS-BSW-27H-05**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Server → Client** | **ID** | **DLC** | **CAN Data Field** | | | | | | | |
| **Physical Request** | **0x7F1** | **8 Byte** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| **Data Length** | **SID**  **REQ** | **SF**  **REQ** | **Data** | **Data** | **Data** | **Data** | **Data** |
| **0x02** | **0x67** | **0x02** | **0x00** | **0x00** | **0x00** | **0x00** | **0x00** |

### 2.3.3 Negative Response Frame Format

**SRS-BSW-27H-06**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Server → Client** | **ID** | **DLC** | **CAN Data Field** | | | | | | | |
| **Physical Request** | **0x7F1** | **8 Byte** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| **Data Length** | **Neg**  **Response** | **SID**  **REQ** | **NRC** | **Data** | **Data** | **Data** | **Data** |
| **0x03** | **0x7F** | **0x27** | **XX** | **0x00** | **0x00** | **0x00** | **0x00** |

### 2.2.4 Supported Negative Response Codes (NRC’s)

|  |  |  |
| --- | --- | --- |
| **Tags** | **NRC** | **Description** |
| **SRS-BSW-27H-07** | **0x12** | **Sub Function Not Supported** |
| **SRS-BSW-27H-08** | **0x13** | **Incorrect Message Length or Invalid Format** |
| **SRS-BSW-27H-09** | **0x22** | **Conditions Not Correct** |
| **SRS-BSW-27H-10** | **0x24** | **Request Sequence Error** |
| **SRS-BSW-27H-11** | **0x31** | **Request Out of Range** |
| **SRS-BSW-27H-12** | **0x35** | **Invalid Key** |
| **SRS-BSW-27H-13** | **0x36** | **Exceeded Number of Attempts** |
| **SRS-BSW-27H-14** | **0x37** | **Required Time Delay Not Expired** |

## Tester Present (3E16) service

**SRS-BSW-3EH-01**

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. The detailed session requirements that apply to the use of this service when keeping a single server or multiple servers in a diagnostic session other than the default Session can be found in the implementation specifications of ISO 14229.

### 2.4.1 Request Frame Format

**SRS-BSW-3EH-02**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Client → Server** | **ID** | **DLC** | **CAN Data Field** | | | | | | | |
| **Physical Request** | **0x7F0** | **8 Byte** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| **Data Length** | **SID**  **REQ** | **SF**  **REQ** | **Data** | **Data** | **Data** | **Data** | **Data** |
| **0x02** | **0x3E** | **XX** | **0x00** | **0x00** | **0x00** | **0x00** | **0x00** |

**SF-REQ-XX:**

1. **Zero Sub Function (0x00): SRS-BSW-3EH-02**

This parameter value is used to indicate that no Sub Function value beside the suppress Positive Response Message Indication Bit is supported by this service.

### 2.4.2 Positive Response Frame Format

**SRS-BSW-3EH-03**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Server → Client** | **ID** | **DLC** | **CAN Data Field** | | | | | | | |
| **Physical Request** | **0x7F1** | **8 Byte** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| **Data Length** | **SID**  **REQ** | **SF**  **REQ** | **Data** | **Data** | **Data** | **Data** | **Data** |
| **0x02** | **0x7E** | **XX** | **0x00** | **0x00** | **0x00** | **0x00** | **0x00** |

### 2.4.3 Negative Response Frame Format

**SRS-BSW-3EH-04**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Server → Client** | **ID** | **DLC** | **CAN Data Field** | | | | | | | |
| **Physical Request** | **0x7F1** | **8 Byte** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| **Data Length** | **Neg**  **Response** | **SID**  **REQ** | **NRC** | **Data** | **Data** | **Data** | **Data** |
| **0x03** | **0x7F** | **0x3E** | **XX** | **0x00** | **0x00** | **0x00** | **0x00** |

### 2.4.4 Supported Negative Response Codes (NRC’s)

|  |  |  |
| --- | --- | --- |
| **Tags** | **NRC** | **Description** |
| **SRS-BSW-3EH-05** | **0x12** | **Sub Function Not Supported** |
| **SRS-BSW-3EH-06** | **0x13** | **Incorrect Message Length or Invalid Format** |

## Routine Control (3116) service

**SRS-BSW-31H-01**

The Routine Control service is used by the client to execute a defined sequence of steps and obtain any relevant results. There is a lot of flexibility with this service, but typical usage may include functionality such as erasing memory, resetting or learning adaptive data, running a self-test, overriding the normal server control strategy, and controlling a server value to change over time including predefined sequences (e.g. close convertible roof) to name a few. In general, when used to control outputs this service is used for more complex type control whereas input Output Control By Identifier is used for relatively simple (e.g. static) output control.

The Routine Control service is used by the client to:

— start a routine;

— stop a routine; and

— request routine results.

**SF-REQ:**

1. **Start Routine (0x01): SRS-BSW-31H-02**

This parameter specifies that the server shall start the routine specified by the routine Identifier.

1. **Stop Routine (0x02): SRS-BSW-31H-03**

This parameter specifies that the server shall stop the routine specified by the routine Identifier.

1. **Request Routine Results (0x03): SRS-BSW-31H-04**

This parameter specifies that the server shall return result values of the routine specified by the routine Identifier.

**List of Routine Identifier (RID’s):-**

1. **ERASE (0xFF00)**
2. **CHECKSUM (0xFF01)**

### 2.5.1 Start Routine (0x01) - Routine Identifier (RID) – ERASE (0xFF00)

### 2.5.1.1 Request Frame Format

**SRS-BSW-31H-05**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Client → Server** | **ID** | **DLC** | **CAN Data Field** | | | | | | | |
| **Physical Request** | **0x7F0** | **8 Byte** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| **Data Length** | **SID**  **REQ** | **SF**  **REQ** | **RID**  **(HB)** | **RID**  **(LB)** | **Erase**  **Block**  **No** | **Data** | **Data** |
| **0x05** | **0x31** | **0x01** | **0xFF** | **0x00** | **0x00** | **0x00** | **0x00** |

**Erase Block No: -**

|  |  |
| --- | --- |
| **Erase Block Number** | **Start Address** |
| 0x00 | 0x10000 |
| 0x01 | 0x18000 |

### 2.5.1.2 Positive Response Frame Format

**SRS-BSW-31H-06**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Server → Client** | **ID** | **DLC** | **CAN Data Field** | | | | | | | |
| **Physical Request** | **0x7F1** | **8 Byte** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| **Data Length** | **SID**  **REQ** | **SF**  **REQ** | **RID**  **(HB)** | **RID**  **(LB)** | **Data** | **Data** | **Data** |
| **0x04** | **0x71** | **0x01** | **0xFF** | **0x00** | **0x00** | **0x00** | **0x00** |

### 2.5.2 Start Routine (0x01) - Routine Identifier (RID) – CHECKSUM (0xFF01)

### 2.5.2.1 Request Frame Format

**SRS-BSW-31H-07**

Sending a First Frame like as mentioned below,

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Client → Server** | **ID** | **DLC** | **CAN Data Field** | | | | | | | |
| **Physical Request** | **0x7F0** | **8 Byte** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| **Data Length** | **Data Length** | **SID**  **REQ** | **SF**  **REQ** | **RID**  **(HB)** | **RID**  **(LB)** | **Data** | **Data** |
| **0x10** | **0x08** | **0x31** | **0x01** | **0xFF** | **0x01** | **0xYY** | **0xYY** |

YY – 4 bytes of Check Sum Value

Receiving a Flow Control Frame like as mentioned below,

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Server → Client** | **ID** | **DLC** | **CAN Data Field** | | | | | | | |
| **Physical Request** | **0x7F1** | **8 Byte** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| **Data** | **Data** | **Data** | **Data** | **Data** | **Data** | **Data** | **Data** |
| **0x30** | **0x00** | **0x00** | **0x00** | **0x00** | **0x00** | **0x00** | **0x00** |

Sending the Consecutive Frame like as mentioned below,

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Client → Server** | **ID** | **DLC** | **CAN Data Field** | | | | | | | |
| **Physical Request** | **0x7F0** | **8 Byte** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| **Data** | **Data** | **Data** | **Data** | **Data** | **Data** | **Data** | **Data** |
| **0x21** | **0xYY** | **0xYY** | **0x00** | **0x00** | **0x00** | **0x00** | **0x00** |

### 2.5.2.2 Positive Response Frame Format

**SRS-BSW-31H-08**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Server → Client** | **ID** | **DLC** | **CAN Data Field** | | | | | | | |
| **Physical Request** | **0x7F1** | **8 Byte** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| **Data Length** | **SID**  **REQ** | **SF**  **REQ** | **RID**  **(HB)** | **RID**  **(LB)** | **Data** | **Data** | **Data** |
| **0x04** | **0x71** | **0x01** | **0xFF** | **0x01** | **0x00** | **0x00** | **0x00** |

### 2.5.3 Stop Routine (0x02):

### 2.5.3.1 Request Frame Format

**SRS-BSW-31H-09**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Client → Server** | **ID** | **DLC** | **CAN Data Field** | | | | | | | |
| **Physical Request** | **0x7F0** | **8 Byte** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| **Data Length** | **SID**  **REQ** | **SF**  **REQ** | **RID**  **(HB)** | **RID**  **(LB)** | **Data** | **Data** | **Data** |
| **0x04** | **0x31** | **0x02** | **0xXX** | **0xXX** | **0x00** | **0x00** | **0x00** |

XX – RID’s as per list.

### 2.5.3.2 Positive Response Frame Format

**SRS-BSW-31H-10**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Server → Client** | **ID** | **DLC** | **CAN Data Field** | | | | | | | |
| **Physical Request** | **0x7F1** | **8 Byte** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| **Data Length** | **SID**  **REQ** | **SF**  **REQ** | **RID**  **(HB)** | **RID**  **(LB)** | **Data** | **Data** | **Data** |
| **0x04** | **0x71** | **0x02** | **0xXX** | **0xXX** | **0x00** | **0x00** | **0x00** |

### 2.5.4 Request Routine Results (0x03):

### 2.5.4.1 Request Frame Format

**SRS-BSW-31H-11**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Client → Server** | **ID** | **DLC** | **CAN Data Field** | | | | | | | |
| **Physical Request** | **0x7F0** | **8 Byte** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| **Data Length** | **SID**  **REQ** | **SF**  **REQ** | **RID**  **(HB)** | **RID**  **(LB)** | **Data** | **Data** | **Data** |
| **0x04** | **0x31** | **0x03** | **0xXX** | **0xXX** | **0x00** | **0x00** | **0x00** |

XX – RID’s as per list.

### 2.5.4.2 Positive Response Frame Format

**SRS-BSW-31H-12**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Server → Client** | **ID** | **DLC** | **CAN Data Field** | | | | | | | |
| **Physical Request** | **0x7F1** | **8 Byte** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| **Data Length** | **SID**  **REQ** | **SF**  **REQ** | **RID**  **(HB)** | **RID**  **(LB)** | **Result** | **Data** | **Data** |
| **0x05** | **0x71** | **0x03** | **0xXX** | **0xXX** | **0xFF** | **0x00** | **0x00** |

### 2.5.5 Negative Response Frame Format

**SRS-BSW-31H-13**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Server → Client** | **I** | **DLC** | **CAN Data Field** | | | | | | | |
| **Physical Request** | **0x7F1** | **8 Byte** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| **Data Length** | **Neg**  **Response** | **SID**  **REQ** | **NRC** | **Data** | **Data** | **Data** | **Data** |
| **0x03** | **0x7F** | **0x31** | **XX** | **0x00** | **0x00** | **0x00** | **0x00** |

### 2.5.6 Supported Negative Response Codes (NRC’s)

|  |  |  |
| --- | --- | --- |
| **Tags** | **NRC** | **Description** |
| **SRS-BSW-31H-14** | **0x12** | **Sub Function Not Supported** |
| **SRS-BSW-31H-15** | **0x13** | **Incorrect Message Length or Invalid Format** |
| **SRS-BSW-31H-16** | **0x22** | **Conditions Not Correct** |
| **SRS-BSW-31H-17** | **0x24** | **Request Sequence Error** |
| **SRS-BSW-31H-18** | **0x31** | **Request Out of Range** |
| **SRS-BSW-31H-19** | **0x33** | **Security Access Denied** |
| **SRS-BSW-31H-20** | **0x72** | **General Programming Failure** |

## Request Download (3416) service

**SRS-BSW-34H-01**

The Request Download service is used by the client to initiate a data transfer from the client to the server (download).

After the server has received the Request Download request message, the server shall take all necessary actions to receive data before it sends a positive response message.

### 2.6.1 Request Frame Format

**SRS-BSW-34H-02**

Sending a First Frame like as mentioned below,

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Client → Server** | **ID** | **DLC** | **CAN Data Field** | | | | | | | |
| **Physical Request** | **0x7F0** | **8 Byte** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| **Data Length** | **Data Length** | **SID**  **REQ** | **Data Format**  **ID** | **Address**  **Mem**  **ID** | **Address**  **data**  **(HB)** | **Address**  **data** | **Address**  **data** |
| **0x10** | **0x0B** | **0x34** | **0x00** | **XY** | **XX** | **XX** | **XX** |

\*\*\*XY – Address and Length Format Identifier (0x44),

XX – Memory Address,

YY – Memory Size.

ZZ – Total length of data,

Receiving a Flow Control Frame like as mentioned below,

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Server → Client** | **ID** | **DLC** | **CAN Data Field** | | | | | | | |
| **Physical Request** | **0x7F1** | **8 Byte** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| **Data** | **Data** | **Data** | **Data** | **Data** | **Data** | **Data** | **Data** |
| **0x30** | **0x00** | **0x00** | **0x00** | **0x00** | **0x00** | **0x00** | **0x00** |

Sending the Consecutive Frame like as mentioned below,

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Client → Server** | **ID** | **DLC** | **CAN Data Field** | | | | | | | |
| **Physical Request** | **0x7F0** | **8 Byte** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| **Data Length** | **Address**  **data**  **(HB)** | **Mem**  **data** | **Mem**  **data** | **Mem**  **data** | **Mem**  **data**  **(LB)** | **Data** | **Data** |
| **0x21** | **XX** | **YY** | **YY** | **YY** | **YY** | **0x00** | **0x00** |

### 2.6.2 Positive Response Frame Format

**SRS-BSW-34H-03**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Server → Client** | **ID** | **DLC** | **CAN Data Field** | | | | | | | |
| **Physical Request** | **0x7F1** | **8 Byte** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| **Data Length** | **SID**  **REQ** | **Length Format**  **ID** | **No of Write**  **(HB)** | **No of Block**  **(LB)** | **Data** | **Data** | **Data** |
| **0x04** | **0x74** | **0x20** | **0x04** | **0x02** | **0x00** | **0x00** | **0x00** |

### 2.6.3 Negative Response Frame Format

**SRS-BSW-34H-04**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Server → Client** | **I** | **DLC** | **CAN Data Field** | | | | | | | |
| **Physical Request** | **0x7F1** | **8 Byte** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| **Data Length** | **Neg**  **Response** | **SID**  **REQ** | **NRC** | **Data** | **Data** | **Data** | **Data** |
| **0x03** | **0x7F** | **0x34** | **XX** | **0x00** | **0x00** | **0x00** | **0x00** |

### 2.6.4 Supported Negative Response Codes (NRC’s)

|  |  |  |
| --- | --- | --- |
| **Tags** | **NRC** | **Description** |
| **SRS-BSW-34H-05** | **0x13** | **Incorrect Message Length or Invalid Format** |
| **SRS-BSW-34H-06** | **0x22** | **Conditions Not Correct** |
| **SRS-BSW-34H-07** | **0x31** | **Request Out of Range** |
| **SRS-BSW-34H-08** | **0x33** | **Security Access Denied** |
| **SRS-BSW-34H-09** | **0x34** | **Authentication Required** |
| **SRS-BSW-34H-10** | **0x70** | **Upload Download Not Accepted** |

## Request Upload (3516) service

**SRS-BSW-35H-01**

The Request Upload service is used by the client to initiate a data transfer from the server to the client (upload).

After the server has received the Request Upload request message the server shall take all necessary actions to send data before it sends a positive response message.

### 2.7.1 Request Frame Format

**SRS-BSW-35H-02**

Sending a First Frame like as mentioned below,

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Client → Server** | **ID** | **DLC** | **CAN Data Field** | | | | | | | |
| **Physical Request** | **0x7F0** | **8 Byte** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| **Data Length** | **Data Length** | **SID**  **REQ** | **Data Format**  **ID** | **Address**  **Mem**  **ID** | **Address**  **data**  **(HB)** | **Address**  **data** | **Address**  **data** |
| **0x10** | **0x0B** | **0x35** | **0x00** | **XY** | **XX** | **XX** | **XX** |

\*\*\*XY – Address and Length Format Identifier (0x44),

XX – Memory Address,

YY – Memory Size,

ZZ – Total length of data.

Receiving a Flow Control Frame like as mentioned below,

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Server → Client** | **ID** | **DLC** | **CAN Data Field** | | | | | | | |
| **Physical Request** | **0x7F1** | **8 Byte** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| **Data** | **Data** | **Data** | **Data** | **Data** | **Data** | **Data** | **Data** |
| **0x30** | **0x00** | **0x00** | **0x00** | **0x00** | **0x00** | **0x00** | **0x00** |

Sending the Consecutive Frame like as mentioned below,

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Client → Server** | **ID** | **DLC** | **CAN Data Field** | | | | | | | |
| **Physical Request** | **0x7F0** | **8 Byte** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| **Data Length** | **Address**  **data**  **(LB)** | **Mem**  **data**  **(LB)** | **Mem**  **data** | **Mem**  **data** | **Mem**  **data**  **(LB)** | **Data** | **Data** |
| **0x21** | **XX** | **YY** | **YY** | **YY** | **YY** | **0x00** | **0x00** |

### 2.7.2 Positive Response Frame Format

**SRS-BSW-35H-03**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Server → Client** | **ID** | **DLC** | **CAN Data Field** | | | | | | | |
| **Physical Request** | **0x7F1** | **8 Byte** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| **Data Length** | **SID**  **REQ** | **Length Format**  **ID** | **No of Block**  **(HB)** | **No of Block**  **(LB)** | **Data** | **Data** | **Data** |
| **0x04** | **0x75** | **0x20** | **0x04** | **0x02** | **0x00** | **0x00** | **0x00** |

### 2.7.3 Negative Response Frame Format

**SRS-BSW-35H-04**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Server → Client** | **I** | **DLC** | **CAN Data Field** | | | | | | | |
| **Physical Request** | **0x7F1** | **8 Byte** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| **Data Length** | **Neg**  **Response** | **SID**  **REQ** | **NRC** | **Data** | **Data** | **Data** | **Data** |
| **0x03** | **0x7F** | **0x35** | **XX** | **0x00** | **0x00** | **0x00** | **0x00** |

### 2.7.4 Supported Negative Response Codes (NRC’s)

|  |  |  |
| --- | --- | --- |
| **Tags** | **NRC** | **Description** |
| **SRS-BSW-35H-05** | **0x13** | **Incorrect Message Length or Invalid Format** |
| **SRS-BSW-35H-06** | **0x22** | **Conditions Not Correct** |
| **SRS-BSW-35H-07** | **0x31** | **Request Out of Range** |
| **SRS-BSW-35H-08** | **0x33** | **Security Access Denied** |
| **SRS-BSW-35H-09** | **0x34** | **Authentication Required** |
| **SRS-BSW-35H-10** | **0x70** | **Upload Download Not Accepted** |

## Transfer Data (3616) service

**SRS-BSW-36H-01**

The Transfer Data service is used by the client to transfer data either from the client to the server (download) or from the server to the client (upload).

The data transfer direction is defined by the preceding Request Download or Request Upload service. If the client initiated a Request Download the data to be downloaded is included in the parameter(s) transfer Request Parameter in the Transfer Data request message(s). If the client initiated a Request Upload the data to be uploaded is included in the parameter(s) transfer Response Parameter in the Transfer Data response message(s).

The Transfer Data service request includes a block Sequence Counter to allow for an improved error handling in case a Transfer Data service fails during a sequence of multiple Transfer Data requests. The block Sequence Counter of the server shall be initialized to one when receiving a Request Download (3416), Request Upload (3516). This means that the first Transfer Data (3616) request message following the Request Download (3416), Request Upload (3516) request message starts with a block Sequence Counter of one.

### 2.8.1 If it is Request Download (Service 34):

### 2.8.1.1 Request Frame Format

**SRS-BSW-36H-02**

Sending a First Frame like as mentioned below,

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Client → Server** | **ID** | **DLC** | **CAN Data Field** | | | | | | | |
| **Physical Request** | **0x7F0** | **8 Byte** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| **Data Length** | **Data Length** | **SID**  **REQ** | **Block**  **Count** | **Data**  **Record** | **Data**  **Record** | **Data**  **Record** | **Data**  **Record** |
| **0x14** | **0x02** | **0x36** | **XX** | **YY** | **YY** | **YY** | **YY** |

\*\*\*XX – Block Sequence counter (Starts from 1), YY – Write data,

**Block count will be increased as per the number of blocks which is calculated as per the memory size**.

Receiving a Flow Control Frame like as mentioned below,

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Server → Client** | **ID** | **DLC** | **CAN Data Field** | | | | | | | |
| **Physical Request** | **0x7F1** | **8 Byte** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| **Data** | **Data** | **Data** | **Data** | **Data** | **Data** | **Data** | **Data** |
| **0x30** | **0x00** | **0x00** | **0x00** | **0x00** | **0x00** | **0x00** | **0x00** |

Sending the Consecutive Frame like as mentioned below,

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Client → Server** | **ID** | **DLC** | **CAN Data Field** | | | | | | | |
| **Physical Request** | **0x7F0** | **8 Byte** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| **Data Length** | **Data**  **Record** | **Data**  **Record** | **Data**  **Record** | **Data**  **Record** | **Data**  **Record** | **Data**  **Record** | **Data**  **Record** |
| **0x2X** | **YY** | **YY** | **YY** | **YY** | **YY** | **YY** | **YY** |

### 2.8.1.2 Positive Response Frame Format

**SRS-BSW-36H-03**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Server → Client** | **ID** | **DLC** | **CAN Data Field** | | | | | | | |
| **Physical Request** | **0x7F1** | **8 Byte** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| **Data Length** | **SID**  **REQ** | **Block**  **Count** | **Data** | **Data** | **Data** | **Data** | **Data** |
| **0x02** | **0x76** | **XX** | **0x00** | **0x00** | **0x00** | **0x00** | **0x00** |

### 2.8.2 If it is Request Upload (Service 35):

### 2.8.2.1 Request Frame Format

**SRS-BSW-36H-04**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Client → Server** | **ID** | **DLC** | **CAN Data Field** | | | | | | | |
| **Physical Request** | **0x7F0** | **8 Byte** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| **Data Length** | **SID**  **REQ** | **Block**  **Count** | **Data** | **Data** | **Data** | **Data** | **Data** |
| **0x02** | **0x36** | **XX** | **0x00** | **0x00** | **0x00** | **0x00** | **0x00** |

\*\*\*XX – Block Sequence counter (Starts from 1).

**Block count will be increased as per the number of blocks which is calculated as per the memory size**.

### 2.8.2.2 Positive Response Frame Format

**SRS-BSW-36H-05**

Receiving a First Frame like as mentioned below,

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Server → Client** | **ID** | **DLC** | **CAN Data Field** | | | | | | | |
| **Physical Request** | **0x7F1** | **8 Byte** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| **Data Length** | **Data Length** | **SID**  **REQ** | **Block**  **Count** | **Data**  **Record** | **Data**  **Record** | **Data**  **Record** | **Data**  **Record** |
| **0x14** | **0x02** | **0x76** | **XX** | **0xZZ** | **0xZZ** | **0xZZ** | **0xZZ** |

\*\*\*XX – Block Sequence counter (Starts from 1),

ZZ – Read data,

Sending a Flow Control Frame like as mentioned below,

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Server → Client** | **ID** | **DLC** | **CAN Data Field** | | | | | | | |
| **Physical Request** | **0x7F1** | **8 Byte** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| **Data** | **Data** | **Data** | **Data** | **Data** | **Data** | **Data** | **Data** |
| **0x30** | **0x00** | **0x00** | **0x00** | **0x00** | **0x00** | **0x00** | **0x00** |

Receiving the Consecutive Frame like as mentioned below,

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Server → Client** | **ID** | **DLC** | **CAN Data Field** | | | | | | | |
| **Physical Request** | **0x7F1** | **8 Byte** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| **Data Length** | **Data**  **Record** | **Data**  **Record** | **Data**  **Record** | **Data**  **Record** | **Data**  **Record** | **Data**  **Record** | **Data**  **Record** |
| **0x2X** | **0xZZ** | **0xZZ** | **0xZZ** | **0xZZ** | **0xZZ** | **0xZZ** | **0xZZ** |

### 2.8.3 Negative Response Frame Format

**SRS-BSW-36H-06**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Server → Client** | **ID** | **DLC** | **CAN Data Field** | | | | | | | |
| **Physical Request** | **0x7F1** | **8 Byte** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| **Data Length** | **Neg**  **Response** | **SID**  **REQ** | **NRC** | **Data** | **Data** | **Data** | **Data** |
| **0x03** | **0x7F** | **0x36** | **XX** | **0x00** | **0x00** | **0x00** | **0x00** |

### 2.8.4 Supported Negative Response Codes (NRC’s)

|  |  |  |
| --- | --- | --- |
| **Tags** | **NRC** | **Description** |
| **SRS-BSW-31H-07** | **0x13** | **Incorrect Message Length or Invalid Format** |
| **SRS-BSW-31H-08** | **0x24** | **Request Sequence Error** |
| **SRS-BSW-31H-09** | **0x31** | **Request Out of Range** |
| **SRS-BSW-31H-10** | **0x33** | **Security Access Denied** |
| **SRS-BSW-31H-11** | **0x71** | **Transfer Data Suspended** |
| **SRS-BSW-31H-12** | **0x72** | **General Programming Failure** |
| **SRS-BSW-31H-13** | **0x73** | **Wrong Block Sequence Counter** |

## Request Transfer Exit (3716) service

**SRS-BSW-37H-01**

This service is used by the client to terminate a data transfer between client and server (upload or download).

### 2.9.1 Request Frame Format

**SRS-BSW-37H-02**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Client → Server** | **ID** | **DLC** | **CAN Data Field** | | | | | | | |
| **Physical Request** | **0x7F0** | **8 Byte** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| **Data Length** | **SID**  **REQ** | **Data** | **Data** | **Data** | **Data** | **Data** | **Data** |
| **0x01** | **0x37** | **0x00** | **0x00** | **0x00** | **0x00** | **0x00** | **0x00** |

### 2.9.2 Positive Response Frame Format

**SRS-BSW-37H-03**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Server → Client** | **ID** | **DLC** | **CAN Data Field** | | | | | | | |
| **Physical Request** | **0x7F1** | **8 Byte** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| **Data Length** | **SID**  **REQ** | **Data** | **Data** | **Data** | **Data** | **Data** | **Data** |
| **0x01** | **0x77** | **0x00** | **0x00** | **0x00** | **0x00** | **0x00** | **0x00** |

### 2.9.3 Negative Response Frame Format

**SRS-BSW-37H-04**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Server → Client** | **I** | **DLC** | **CAN Data Field** | | | | | | | |
| **Physical Request** | **0x7F1** | **8 Byte** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| **Data Length** | **Neg**  **Response** | **SID**  **REQ** | **NRC** | **Data** | **Data** | **Data** | **Data** |
| **0x03** | **0x7F** | **0x37** | **XX** | **0x00** | **0x00** | **0x00** | **0x00** |

### 2.9.4 Supported Negative Response Codes (NRC’s)

|  |  |  |
| --- | --- | --- |
| **Tags** | **NRC** | **Description** |
| **SRS-BSW-37H-05** | **0x13** | **Incorrect Message Length or Invalid Format** |
| **SRS-BSW-37H-06** | **0x24** | **Request Sequence Error** |
| **SRS-BSW-37H-07** | **0x31** | **Request Out of Range** |
| **SRS-BSW-37H-08** | **0x72** | **General Programming Failure** |