UDS Request and Response codes

UDS uses a frame structure to send data to controllers. Single Frames (SF) are for short messages, where all the data can fit into six bytes. If the data is longer, a FirstFrame (FF) is sent to start the transaction, then Consecutive Frames (CF) are sent with data. Here’s a layout of how the frames are structured.

Single Frame:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Byte0  7-4 bits:0  3-0 bits: number of data bytes in the message | Byte 1  SID | Byte2  DATA | Byte3  DATA | Byte4  DATA | Byte5  DATA | Byte6  DATA | Byte7  DATA |

First Frame:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Byte0  7-4 bits:1  3-0 bits: number of data bytes in the message | Byte 1  7-4 bits:0x01  3-0 bits: Upper bits of message length | Byte2  Lower Byte of message length | Byte3  SID | Byte4  DATA | Byte5  DATA | Byte6  DATA | Byte7  DATA |

Consecutive Frames:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Byte0  7-4 bits:2  3-0 bits: number of data bytes in the message | Byte 1  DATA | Byte2  DATA | Byte3  DATA | Byte4  DATA | Byte5  DATA | Byte6  DATA | Byte7  DATA |

The service ID mentioned in the UDS message response could be positive or negative. The DATA following the request SID is usually the sub functions used by the services. One can also suppress the response to be not positive. This is indicated when the highest bit of byte representing the sub function is set to 1. All negative response codes(NRC) will have the following format:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0x7F | Request Service ID | Response Code |  |  |  |  |  |

There are various codes defined by UDS protocol under ISO. Some of the codes can be shown below

***Diagnostic Session Control***

The UDS-service Diagnostic Session Control is used to control which diagnostic

session the ECU should be in. There are a few different sessions, which are used

for different purposes. The sessions, which are specified in the UDS-standard, are

default-, programming-, extended- and safety system diagnostic session.

Request Code:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte# | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| Value | 0x10 | 0x02 |  |  |  |  |  |  |

0x10 denotes the service ID for Diagnostics session, while 0x02 sub function requests programming session in the ECU to upload software.

Positive Response code:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte# | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| Value | 0x50 | 0x02 | 0x00 | 0x0A | 0x01 | 0xF4 |  |  |

A positive diagnostic session control service response will contain P2 and P2 extended timing values. These values represent the maximum time the ECU should take to return a response to a UDS-request in the current session. Byte 1 is the response code, Byte 2 is the sub function in the request. Byte 3 and 4 denotes the time the ECU does not send a NRC, which is extended to Byte 5 and 6.

Negative Response:

|  |  |
| --- | --- |
| NRC | Description |
| 0x12 | Sub-function Not Supported |
| 0x13 | Incorrect Message Length Or Invalid Format |
| 0x22 | Conditions Not Correct |
|  |  |

***ECU Reset service***

The UDS-service ECU Reset is used to perform a reset of the ECU. This service is

usually used in post-programming. There are several sub-functions defined in the UDS-standard but only the

sub-function “Hard Reset” is used in the software loading sequences

Request:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte# | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| Value | 0x11 | 0x01 |  |  |  |  |  |  |

Positive Response

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte# | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| Value | 0x51 | 0x01 |  |  |  |  |  |  |

Negative Response:

|  |  |
| --- | --- |
| NRC | Description |
| 0x12 | Sub-function Not Supported |
| 0x13 | Incorrect Message Length Or Invalid Format |
| 0x22 | Conditions Not Correct |
| 0x33 | Security Access Denied |
|  |  |

***Security Access service***

To prevent unauthorized access to the ECU the vehicle manufacturers implement the Security Access service, which is specified in the UDS-standard. Generally, security access is required before any transfer of new software to the ECU can be performed. The Security Access service utilizes a seed and key structure; the tester (client) will request security access with a UDS-request

Request:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte# | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| Value | 0x27 | 0x01 |  |  |  |  |  |  |

Response with Security seed:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte# | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| Value | 0x67 | 0x01 | 0xC6 | 0xF8 | 0x98 | 0x69 |  |  |

Request with Security key:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte# | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| Value | 0x27 | 0x02 | 0xBF | 0xFC | 0xE7 | 0xC3 |  |  |

Response with access to tester(Client):

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte# | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| Value | 0x67 | 0x02 |  |  |  |  |  |  |

Negative Response:

|  |  |
| --- | --- |
| NRC | Description |
| 0x12 | Sub-function Not Supported |
| 0x13 | Incorrect Message Length Or Invalid Format |
| 0x22 | Conditions Not Correct |
| 0x24 | Request sequence Error |
| 0x31 | Request out of Range |
| 0x35 | Invalid key |
| 0x36 | Exceeded number of attempts |
| 0x37 | Required time delay not required |

***Read by Identifier service***

Service to read data at a memory location specified, used in a flashing sequence to read programming-, fingerprint-data and prepare the ECU for reprogramming. It is a vehicle manufacturer specific step that is sometimes included in the pre-programming part of the software loading sequence.

Request:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte# | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| Value | 0x22 | 0xF1 | 0x58 |  |  |  |  |  |

The byte 2 and 3 specifies the memory location where the required data is stored.

Positive Response:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte# | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| Value | 0x62 | 0x02 | 0x1A | 0x39 |  |  |  |  |

The value for bytes 3 and 4 is data stored in the specific memory location that is shown in the specific UDS request for read data by identifier.

Negative Response:

|  |  |
| --- | --- |
| NRC | Description |
| 0x12 | Sub-function Not Supported |
| 0x13 | Incorrect Message Length Or Invalid Format |
| 0x22 | Conditions Not Correct |
| 0x31 | Request out of Range |
| 0x14 | Response too long |

***Write by Identifier service***

The Write by Identifier service is used to write data to a specific memory location, for example writing programming date and fingerprint data. Typically, this is done right before transferring data to the ECU and/or after a successful software loading sequence.

Request:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte# | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| Value | 0x2E | 0xF1 | 0x58 | 0x15 | 0x11 | 0x19 | 0x02 | 0x03 |

Byte 2 and 3 refer to memory address of the location where the data, represented by remaining bytes, is to be written over

Positive Response:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte# | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| Value | 0x2E |  |  |  |  |  |  |  |

Negative Response:

|  |  |
| --- | --- |
| NRC | Description |
| 0x13 | Incorrect Message Length Or Invalid Format |
| 0x22 | Conditions Not Correct |
| 0x31 | Request out of Range |
| 0x33 | Security Access denied |
| 0x72 | General Programming failure |

***Write by Identifier service***