Automotive & Embedded Info

Never Forget Basics Whether its Life or Anything Else ... Basics are Cores. While seeing a Tree how can we forget Seed...

UDS

Unified Diagnostic Services

UDS is application layer protocol by which ECU diagnostic can be performed.

UDS and ODB Positioning in ISO/OSI Model:

S. No.	Layer	Description	Standard for UDS	Standard for OBD
1	8	Diagnostic Application Layer	User	ISO 15031-5
2	7	Application Layer	ISO 14229-1 ISO 15765-3	ISO 15031-5
3	6	Presentation Layer	Not applicable	Not applicable
4	5	Session Layer	ISO 15765-3	Not applicable
5	4	Transport layer	ISO 15765-2	Not applicable
6	3	Data Link Layer	ISO 15765-2	ISO 15765-4
7	2	Data Link Layer	ISO 11898-1	ISO 15765-4
8	1	Physical Layer	ISO 11898-2* ISO 11898-3	ISO 15765-4

* The bus physics used is to be selected by the user. Therefore, several standards for different physical layer for e.g. High-Speed- (ISO 11898-2) or Fault-Tolerant-CAN (ISO 11898-3) can be used.

Diagnostic Messages – Structures and Types:

Diagnostic services have a common message format. Each service defines a Request Message, a Positive Response Message, and a Negative Response Message.

1. Request without Sub-Function or Positive Response:
2. Requests with Sub-Function Byte:

3. UUDT-Responses
4. Negative Responses

Diagnostic Messages	Structures and Types		
Request without Sub-Function or Positive Response	Service Identifier	Data Parameter	
Requests with Sub-Function Byte	Service Identifier	Sub-Function Byte	Data Parameter
UUDT-Responses	Data Parameter		
Negative Responses	Negative Response SID	Request Service Identifier	Response Code

Positive Messages:

The Positive Response Message is usually:

1. The Positive Response Message has an echo of the service ID with bit 6 set as first byte, plus
2. The service-defined response parameters.

Negative Messages:

The Negative Response Message is usually a three-byte message:

1. It has the Negative Response Service ID (0x7F) as first byte,

3. A Response Code as third byte.

The UDS standard partly defines the ResponseCodes, but there is room left for manufacturer-specific extensions. For some of the ResponseCodes, UDS defines an error handling procedure. Because both positive and negative responses have an echo of the requested service, you always can assign the responses to their corresponding request

The Sub-Function Byte of UDS:

SERVICES WITHOUT SUB-FUNCTION-BYTE:

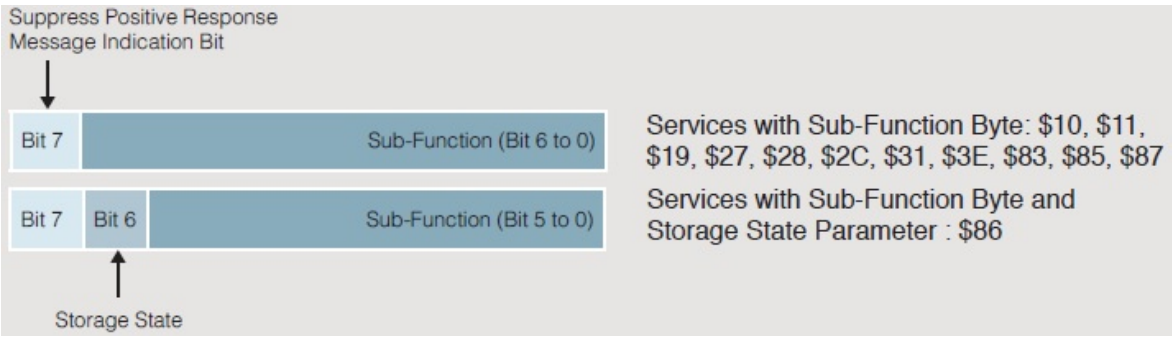
$14, $23, $24, $2A, $2E, $2F, $34, $35, $36, $37, $3D, $84.

SERVICES WITH SUB-FUNCTION BYTE:

$10, $11, $19, $27, $28, $2C, $31, $3E, $83, $85, $87

SERVICES WITH SUB-FUNCTION BYTE AND STORAGE STATE PARAMETER :

$86



BIT 7: SUPPRESSPOSRSPMSGINDICATIONBIT:

This bit defines whether a positive response of ECU is wanted

'0' = FALSE : the ECU shall send a response, that is, no suppression of positive response shall be done.

'1' = TRUE: suppression of positive response, that is , the ECU must not send a positive response.

Negative responses shall be send by ECU independent of this bit.

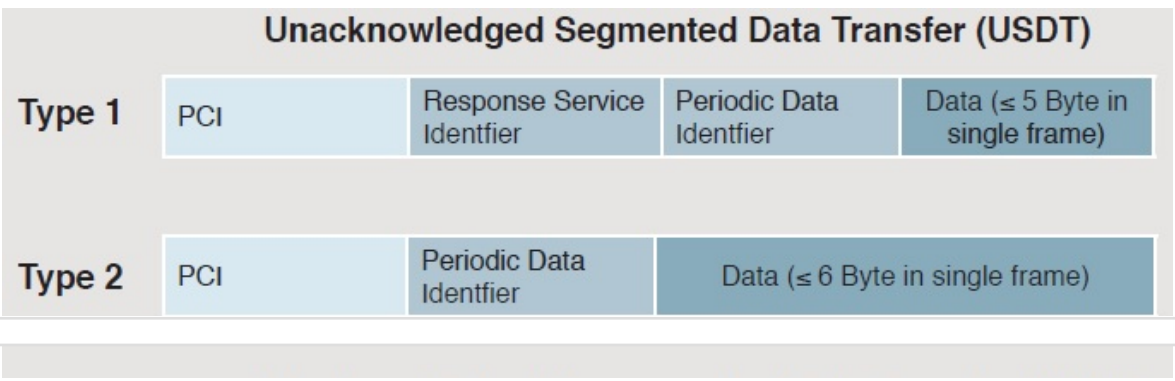
BIT 6 TO 0: SUB-FUNCTION PARAMETER VALUE

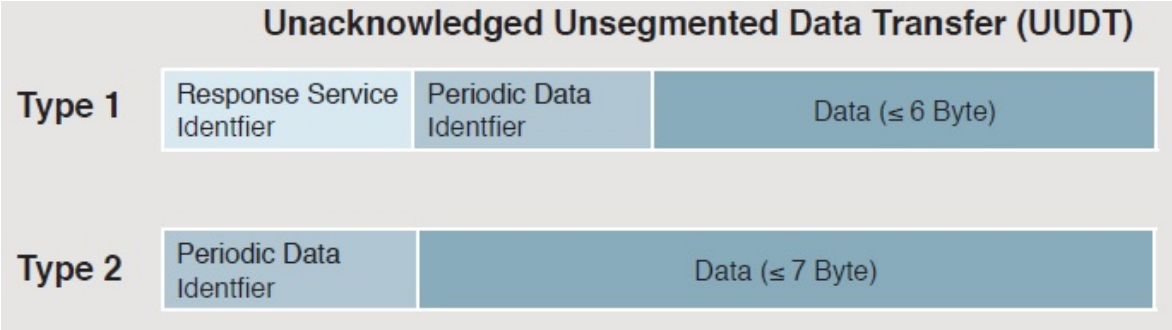
The Bits 0 to 6 contain the value for a Sub-Function parameter for diagnostic services

Note: A service, which uses the suppressPosRspMsgIndicationBit only, shall set the value of the Sub-Function parameter to 0 to support the sending of the bit 7. The Bits 0 to 6 are meaningless then.

Periodic Message Types of UDS:

1. Unacknowledged Segmented Data Transfer (USDT)
2. Unacknowledged Un-segmented Data Transfer (UUDT)





Responses can be of two different formats/types:

1. Type 1: With response service identifier.
2. Type 2: Without response service identifier.

The type is respected by service $2A read data by periodic identifier.

UDS Diagnostic Services:

Diagnostic services are used to diagnose within ECU like Diagnostic and Communications Management, Data Transmission, Stored Data Transmission, Input / Output Control, Remote Activation of Routine, Upload / Download etc. There are many diagnostic services used/defines by ISO standards. Below are the diagnostic services used in Automotive:

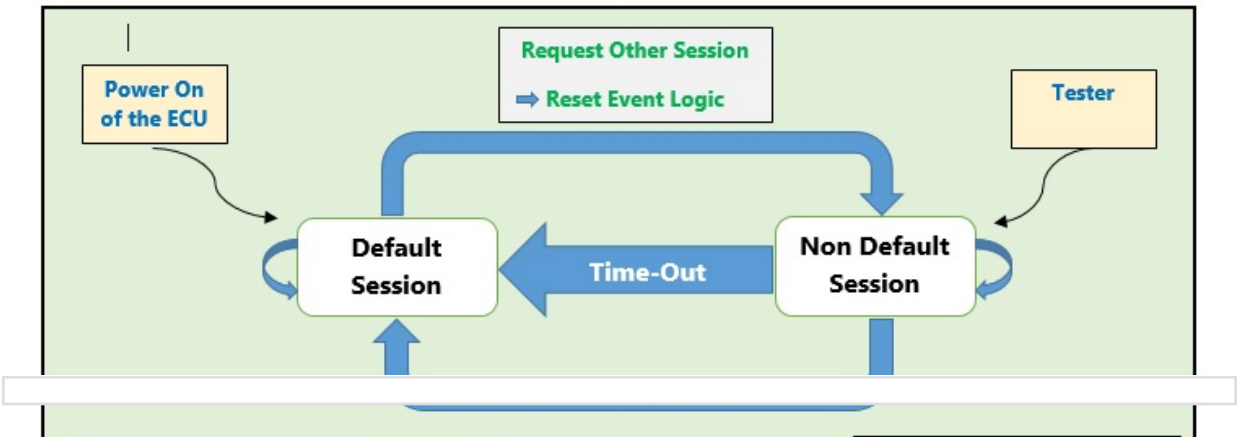
Function group	Request SID	Response SID	Service	Available in Default Session	Available for RoE	Has Sub-Function
Diagnostic and Communications Management	$10	$50	Diagnostic Session Control	X	-	X
	$11	$51	ECU Reset	X	-	X
	$27	$67	Security Access	X	-	X
	$28	$68	Communication Control	X	-	X
	$3E	$7E	Tester Present	X	-	X
	$83	$C3	Access Timing Parameters	X	-	X
	$84	$C4	Secured Data Transmission	X	-	X
	$85	$C5	Control DTC Settings	X	-	X
	$86	$C6	Response On Event	X	-	X
	$87	$C7	Link Control	X	-	X

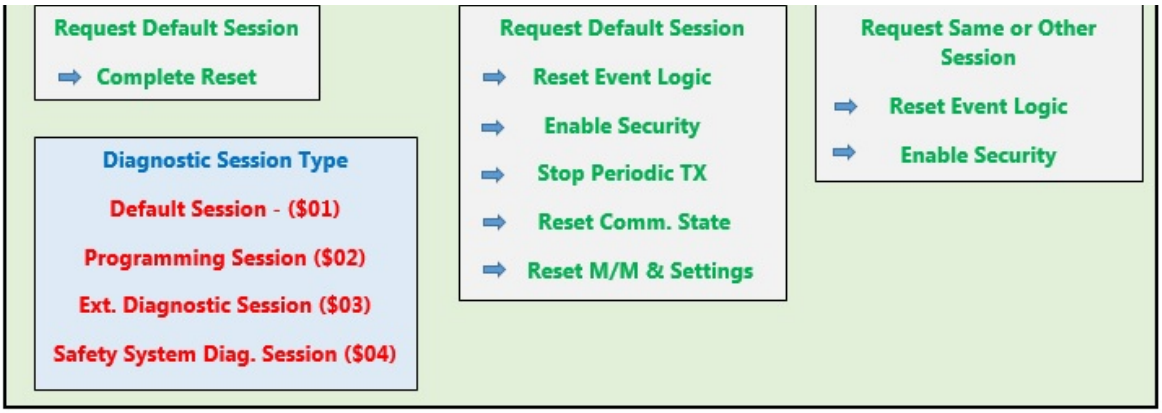
Data Transmission	$22	$62	Read Data By Identifier	X	X	-
	$23	$63	Read Memory By Address	X	X	-
	$24	$64	Read Scaling Data By Identifier	X	X	-
	$2A	$6A	Read Data By Identifier Periodic	X	X	-
	$2C	$6C	Dynamically Define Data Identifier	X	X	X
	$2E	$6E	Write Data By Identifier	X	X	X
	$3D	$7D	Write Memory By Address	X	X	X
Stored Data Transmission	$14	$54	Clear Diagnostic Information	X	-	-
	$19	$59	Read DTC Information	X	X	X
Input / Output Control	$2F	$6F	Input Output Control By Identifier	-	X	-
Remote Activation of Routine	$31	$71	Routine Control	X	X	X
Upload / Download	$34	$74	Request Download	-		
	$35	$75	Request Upload	-		
	$36	$76	Transfer Data	-		
	$37	$77	Request Transfer Exit	-		
	$38	$78	Request File Transfer	-		

ROE- Response of Event

Session Handling:

Diagnostic ensures data flow concerning diagnostic requests and responses, supervises and guarantees diagnostic protocol timing and manages diagnostic states.





Diagnostic session makes sure the communication between the ECU and the diagnostic tool. Different types of diagnostics sessions:

1. Default Diagnostic Session,
2. Programming Session
3. Extended Diagnostic Session.
4. Safety System Diagnostic Session.

After power on, ECU will be switched to a Default Diagnostic Session. After receiving different request from Diagnostic Tool, the ECU will be switched to different diagnostic session.

UDS Response Handling:

Specifies:

- All services, which have a parameter “Sub-Function”, support the “Response-Suppression-Handling“.
- All services to read data do not support this feature.
- A service, which uses the suppressPosRspMsgIndicationBit of the Sub-Function Byte only, must set the other bits of the Sub-Function Byte to 0, to support the transmission of the Bit 7.
- SuppressPositiveResponseMsgIndicationBit = TRUE
- Suppression of the positive response
- All negative responses are send nevertheless

- For Functional requests, some specific negative responses are always to be suppressed, independent of the value of the Suppress Positive Response Message Indication Bit:
- Service not supported (NRC $11)
- Subfunction not supported (NRC $12)

“Respond-to”-Services

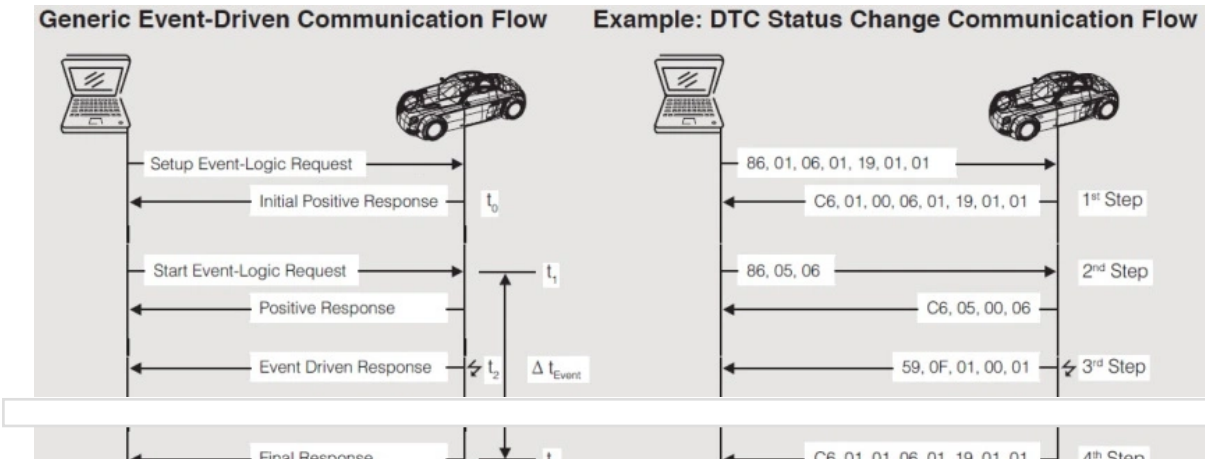
- ReadDataByIdentifier ($22)
- ReadDTCInformation ($19)
- RoutineControl ($31)
- InputOutputControlByIndentifier ($2F

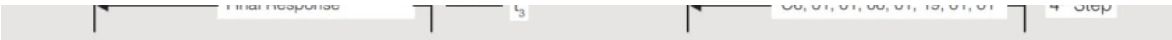
Response on Event:

For one or two setup and start Requests one or two initial Responses are given, followed by 0 to n event-driven Responses depending on the number of occurrences of tracked events. The distance between several events is non-deterministic.

· The RoE mechanism can be activated in any Session, including the Default-Session

· It does not need Tester Present messages to stay active



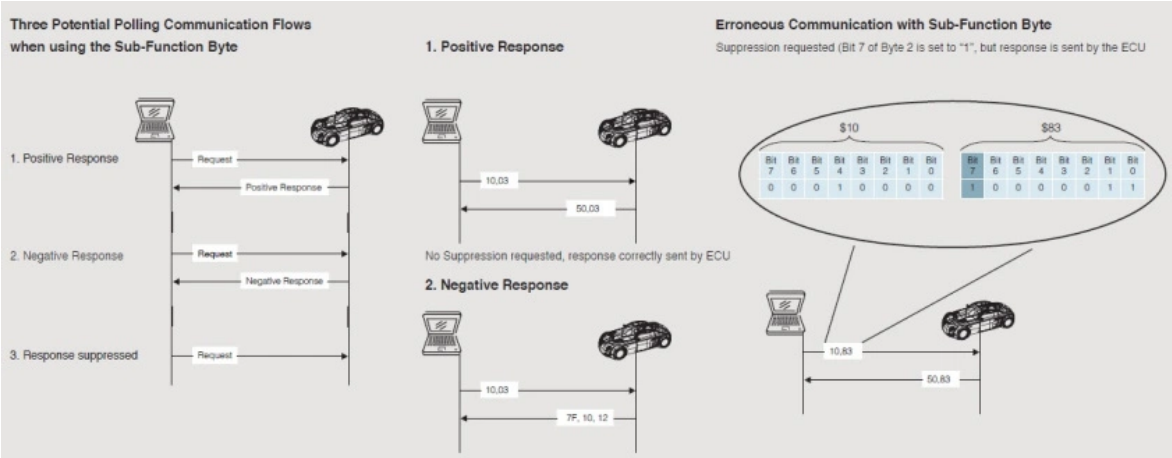


Pre-defined values for parameter Event Type (6 Bits)

1. $0 Stop Event-Logic
2. $1 Event on change of error memory, e. g. number of errors matching to mask has increased
3. $3 Event on change of measurement's value, which is described by data identifier
4. $5 Start Event-Logic
5. $6 Clear Event-Logic

Simple/Polling Diagnostic Service:

These services consist of one request and one response (max.) for physical addressing, or a group of responses for functional addressing.



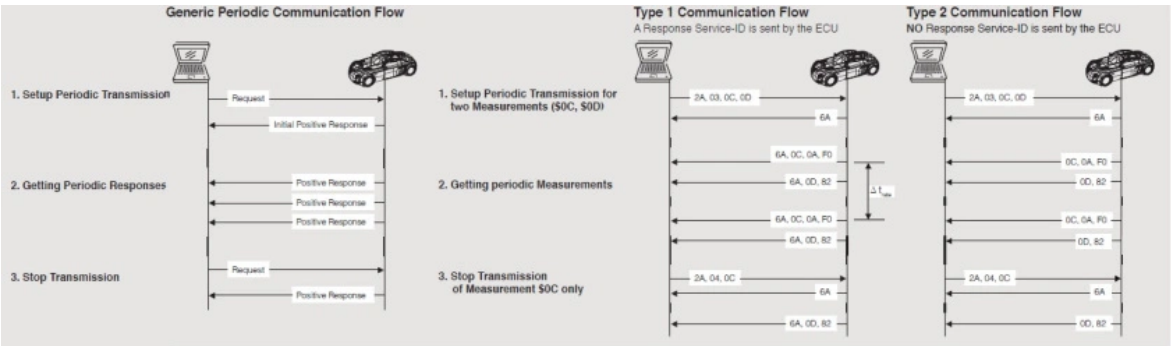
Periodic Service Execution:

For one Request follows one initial Response. After that, periodically more responses will follow. The sending can be stopped by using a simple diagnostic service Service parameter "Transmission Mode"

The frequency of data transmission can be configured using the transmission mode. The UDS standard define abstract values as:

1. Slow = $01
2. Medium = $02
3. Fast = $03
4. Stop Sending = $04

The real value in Hertz for each frequency need to be defined between OEM and ECU supplier.



Error Memory Functions:

1. Erase Error Memory ($14 Clear Diagnostic Information)

   Parameter 1: Service ID = $14

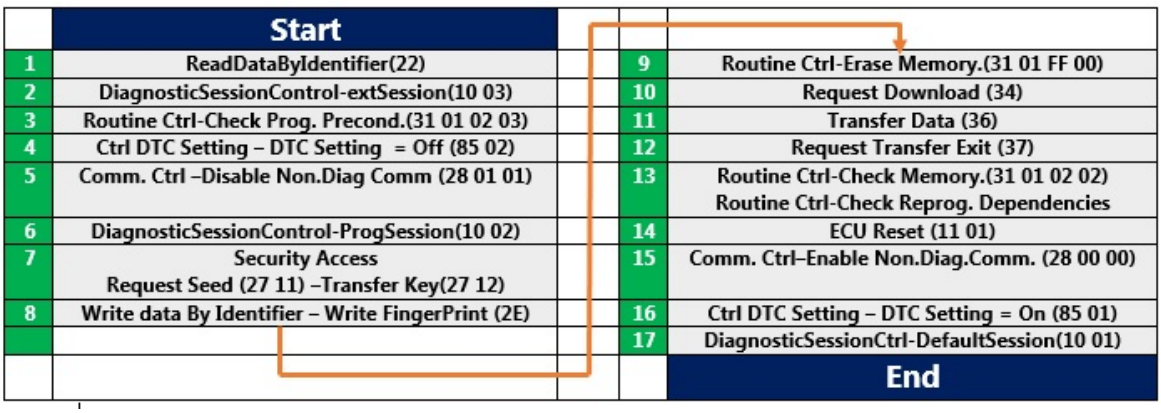
   Parameter 2: Diagnostic Trouble Code (DTC) with three byte length

2. Sub functions For services $19 Read DTC Information

S. No.	HEX	Description
1	0x01	Report number of DTC by Status Mask
2	0x02	Report DTC by Status Mask
3	0x03	Report DTC snapshot Identification
5	0x05	Report DTC Snapshot Record by Record number

6	0x06	Report DTC Extended Data Record by DTC number
7	0x07	Report number of DTC by Severity Mask Record
8	0x08	Report DTC by Severity Mask Record
9	0x09	Report Severity Information of DTC
10	0x0A	Report Supported DTC
11	0x0B	Report First Test Failed DTC
12	0x0C	Report First Test Confirmed DTC
13	0x0D	Report Most Recent Test Failed DTC
14	0x0E	Report Most Recent Test Confirmed DTC
15	0x0F	Report Mirror Memory DTC by Status Mask
16	0x10	Report Mirror Memory DTC Extended Data Record by DTC number
17	0x11	Report number of Mirror Memory DTC by Status Mask
18	0x12	Report number of Emissions Related OBD DTC by Status Mask
19	0x13	Report Emissions Related OBD DTC by Status Mask
20	0x14	Report DTC Fault Detection Counter
21	0x15	Report DTC with Permanent Status

Example: UDS Flash Programming:



1. The diagnostic tester sends a Growl *ReadDataByIdentifier*. With this request, it reads the hardware ID and software ID from the controller to see which device it exactly right.

2. Then, the diagnostic tester, the control unit switches to a special diagnostic session. Not the actual diagnosis session for the program, but a session in which there are a number of advanced services available. This is done with the diagnostic service *diagnosticSession control*. This advanced diagnostic session asks the diagnostic tester, the control unit whether the preconditions are met for flash programming.

3. Then, the diagnostic tester usually with the service *Communication Control* the fault memory and off the bus communication in other controllers. This advanced diagnostic session has served its purpose.

4. Now the diagnostic tester on the diagnostic service *diagnosticSession control* to the programming session. At least now is a *SecurityAccess* necessary.

5. Thereafter, the diagnostic tester usually sends the so-called fingerprint of the control unit. This is information that is stored in the ECU memory permanently, to indicate that programming. It typically has a workshop identifier in the memory of the controller is written, can be enacted so that afterwards who has reprogrammed the ECU.

6. Before the flash memory can be reprogrammed, it must be deleted. This is achieved by calling a routine in the ECU memory of the diagnostic service *routine control* done. Thereafter, the service *request download* the actual programming operation is initiated. This service allows the controller will also be notified of which are loaded into memory the data and how much data can be expected.

7. Now starts the actual download of data in a loop with the service *transfer data*. The storage area is transmitted here in blocks. At the end of the diagnostic tester says the control device *transfer exit* Now that all data has been transferred. After examining the data transmitted in the control unit now takes place, the actual flash process. Typically, the programming operation will take some time. During this time the controller is not able to process requests from the tester. Therefore, the control unit the service *transfer exit* usually with a negative response and the error code *ResponsePending* . Reply Only when the programming is completed, the controller sends a positive confirmation *transfer exit.*

8. Then examine the diagnostic tester, whether programming was

successful, he *routine control* a routine in the control unit is activated, which checks the memory. Thereafter, a further call to *routine control* different n dependence of the flash programming examined, such as whether the software or the corresponding record must be programmed. The download process is completed the controller, the controller is normally *ECUReset* reset. The controller will reboot and goes to normal operation, so back to the default diagnostic session.

Like 

One blogger likes this.