Specification of Diagnostic Communication (Diagnostics on CAN - Application & Session Layer)

<CONTENTS>

1. Scope

This specification prescribes about Application Layer and Session Layer utilized for data communication between the electronic control unit (ECU) and a diagnostic tool on CAN bus.
Contents without this specification refer to ISO 14229-1 (UDS on CAN).

2. Application

This specification applies to ECUs and a diagnosis tester to execute data communication on CAN bus.
The specification details refer to each ECU's control specifications.

3. Definition of Terms

The key terms used in this specification are defined as follows:

(1) $
The values following to the symbol $ are the number of hexadecimal.
(Store appropriate numerical value in "$XX").

(2) System Check Code (SCC)
System check code refers to the data of 2[byte] ($XXXX) used by a tester to identify ECU. Such codes are set for the services supported by ECU (i.e. service IDs, parameters, and other kinds of data). A tester decides based on the obtained SCC, what types of services are supported by the ECU for communication to carry out the subsequent communications.

(3) DTC (Diagnostic Trouble Code)
A code that identifies one unique malfunctions detected by an individual ECU.

(4) Message Usage
The following symbols indicate the nature of each byte composing a message as follows:

- M (Mandatory)	: The data byte is supported by all means.
- C (Conditional)	: A support of the data byte depends on each Identifier.
- O (Optional)	: A support of the data byte is optional handling.

4.Diagnostic Message Format

4.1 Diagnostic Request Message Format

A Diagnostic Request Message is sent by the diagnostic tool when requesting an ECU to perform a specific action (e.g. report diagnostic data or execute diagnostic test procedures).
The format of Diagnostic Request Message is classed in two kinds of follows.

4.1.1 Diagnostic Request Message (with Sub Function)

Table4.1 Diagnostic Request Message Definition (with Sub Function)

Data byte no.	Parameter name	Message Usage	Data value
1	Request Service ID (see section 7, 8 about Detail of Service ID)	M	$XX
2	Sub Function Parameter	M	
	Option 1 :Sub Function Parameter no.1		$XX
	Option n :Sub Function Parameter no.n		$XX
3	Data Parameter no.1	O	$XX
:	:	:	:
n	Data Parameter no.n	O	$XX

(1) Sub Function Parameter Byte Structure

The structure of Sub Function Parameter Byte in Diagnostic Request Message is shown in Table4.2.
About Positive Response Message Indication Bit that shows ECU's response behavior, see section 5.

Table4.2 Sub Function Parameter Byte Structure

Sub Function Parameter Byte							
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Positive Response Message Indication Bit	Sub Function type is defined in each Service						

4.1.2 Diagnostic Request Message (without Sub Function)

Table4.3 Diagnostic Request Message Definition (without Sub Function)

Data byte no.	Parameter name	Message Usage	Data value
1	Request Service ID (see section 7, 8 about Detail of Service ID)	M	$XX
2	Data Parameter no.1	M	$XX
:	:	:	:
n	Data Parameter no.n	O	$XX

4.2 Diagnostic Response Message Format

The ECU shall send a Diagnostic Response Message (Positive or Negative) for the previously received Request Message.
The format of Diagnostic Response Message is classed in two kinds of follows.

4.2.1 Diagnostic Positive Response Message

Table4.4 Diagnostic Response Message Definition

Data byte no.	Parameter name	Message Usage	Data value
1	Response Service ID	M	Request Service ID + $40
2	Request Sub Function Type or Data parameter no.1	M	$XX
3	Data Parameter no.a	O	$XX
:	:	:	:
n	Data Parameter no.x	O	$XX

4.2.2 Diagnostic Negative Response Message

Table4.5 Diagnostic Negative Response Message Definition

Data byte no.	Parameter name	Message Usage	Data value
1	Negative Response Service ID	M	$7F
2	Request Service ID	M	$XX
3	Negative Response Code	M	$XX

5. ECU Response Implementation Requirements

5.1 Request Messages with Sub Function Parameter

5.1.1 Diagnostic Request Message (Physically Addressed)

The ECU shall respond to Diagnostic Tool requests in accordance with Table5.1, if the physically addressed request includes a Sub Function Parameter.

Table5.1 ECU Response Behavior - Physically Addressed Request Message with Sub Function Parameter

ECU Case No.	Request Message		ECU Capability		ECU Response		ECU Response Definition
	Type	Sub Function	Service ID Supported	Sub Function Parameter supported	Message	Negative Response Code	
1	Physical	FALSE (Bit=0)	Yes	Yes	Positive	-	ECU sends Positive Response
2			Yes	Yes	Negative	$XX	ECU sends Negative Response
3			No	-	Negative	$11	ECU sends Negative Response - Service not supported
4			Yes	No	Negative	$12	ECU sends Negative Response - Sub function not supported
5		TRUE (Bit=1)	Yes	Yes	No	-	ECU does not send a Response
6			Yes	Yes	Negative	$XX	ECU sends Negative Response
7			No	-	Negative	$11	ECU sends Negative Response - Service not supported
8			Yes	No	Negative	$12	ECU sends Negative Response - Sub function not supported

Note) The $XX means that the ECU responds with a particular Negative Response Code unequal to $11 or $12.

5.1.2 Diagnostic Request Message (Functionally Addressed)

The ECU shall respond to Diagnostic Tool requests in accordance with Table5.2, if the functionally addressed request includes a Sub Function Parameter.

Table5.2 ECU Response Behavior - Functionally Addressed Request Message with Sub Function Parameter

ECU Case No.	Request Message		ECU Capability		ECU Response		ECU Response Definition
	Type	Sub Function	Service ID Supported	Sub Function Parameter supported	Message	Negative Response Code	
1	Functional	FALSE (Bit=0)	Yes	Yes	Positive	-	ECU sends Positive Response
2			Yes	Yes	Negative	$XX	ECU sends Negative Response
3			No	-	No	-	ECU does not send a Response
4			Yes	No	No	-	ECU does not send a Response
5		TRUE (Bit=1)	Yes	Yes	No	-	ECU does not send a Response
6			Yes	Yes	Negative	$XX	ECU sends Negative Response
7			No	-	No	-	ECU does not send a Response
8			Yes	No	No	-	ECU does not send a Response

Note) The $XX means that the ECU responds with a particular Negative Response Code unequal to $11 or $12.

Complement) About Definition of Physical Address and Functional Address, refer to the Specification of Diagnostic Communication (Diagnostic on CAN - Network Layer)

5.2 Request Messages without Sub-Function Parameter

5.2.1 Diagnostic Request Message (Physically Addressed)

The ECU shall respond to Diagnostic Tool requests in accordance with Table5.3, if the physically addressed request does not include a Sub Function Parameter.

Table5.3 ECU Response Behavior - Physically Addressed Request Message without Sub Function Parameter

ECU Case No.	Request Message	ECU Capability		ECU Response		ECU Response Definition
		Service ID Supported	Parameter supported	Message	Negative Response Code	
1	Physical	YES	ALL	Positive	-	ECU sends Positive Response
2			At least 1	Positive	-	ECU sends Positive Response
3			At least 1, more than 1, or ALL	Negative	$XX	ECU sends negative response because error occurred reading data parameters of request message
4			NONE	Negative	$31	ECU sends Negative Response - Request out of range
5		NO	-	Negative	$11	ECU sends Negative Response - Service not supported

Note) The $XX means that the ECU responds with a particular Negative Response Code unequal to $11 or $31.

5.2.2 Diagnostic Request Message (Functionally Addressed)

The ECU shall respond to Diagnostic Tool requests in accordance with Table5.4, if the functionally addressed request does not include a Sub Function Parameter.

Table5.4 ECU Response Behavior - Functionally Addressed Request Message without Sub Function Parameter

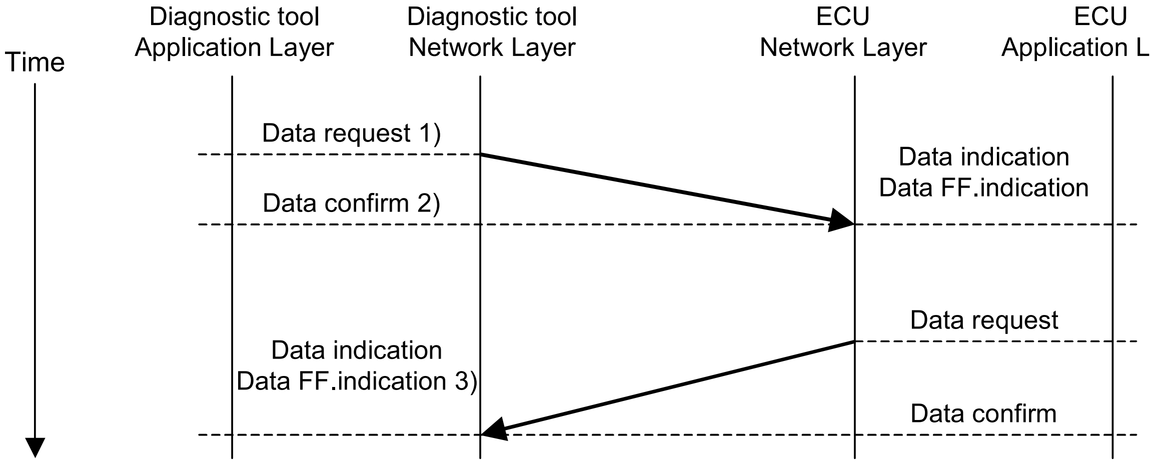
ECU Case No.	Request Message	ECU Capability		ECU Response		ECU Response Definition
		Service ID Supported	Parameter supported	Message	Negative Response Code	
1	Functional	YES	YES	Positive	-	ECU sends Positive Response
2			At least 1	Positive	-	ECU sends Positive Response
3			At least 1, more than 1, or ALL	Negative	$XX	ECU sends negative response because error occurred reading data parameters of request message
4			NONE	No	-	ECU does not send a Response
5		NO	-	No	-	ECU does not send a Response

Note) The $XX means that the ECU responds with a particular Negative Response Code unequal to $11 or $31.

Complement) About Definition of Physical Address and Functional Address, refer to the Specification of Diagnostic Communication (Diagnostic on CAN - Network Layer)

6. Application Layer and Diagnostic Session Management Timing

The interactions at the service access point between the diagnostic application layer protocol instance and the network layer protocol instance on sender or receiver side define specific dates within the communication process of the diagnostic tool and the ECU. Consequently a specific point in time is indicated when a selected service primitive is executed.
The Network Layer timing parameters refer to the Specification of Diagnostic Communication (Diagnostic on CAN - Network Layer).
The Figure6.1 show an overview how the Network Layer services primitives are used.



(1) This service is used to request the transfer of data. If necessary the Network Layer segments the data.
(2) This service is used to confirm to the application layer that the data request service has been carried out (successfully or not).
(3) This service either is used to provide the received data (single frame or multi frame message) to the application layer or to signalize the beginning of a segmented message.

Figure 6.1: Network Layer Service Primitives

Note) the Figure6.1 assumes that no gateway device is located between transmitting diagnostic tool and receiving ECU on the vehicle CAN.

6.1 Application Layer Timing Parameter Definitions

The application layer timing parameters for the diagnostic data exchange between a diagnostic tool and either one individual ECU or a group of ECUs are shown in Table6.1.

Table6.1 Diagnostic Application Layer Parameter

Timing Parameter	Description	Min	Max
P2 CAN_ECU	The ECU shall provide response message (positive or negative) and send its message within this time.	0 ms	20 ms
P2*CAN_ECU	After the transmission of a negative response message with response code $78, the ECU shall provide response message (positive or negative) and send its message within this time.	0 ms	2000 ms
P3 CAN_ECU	A minimum time between a ECU's response message and next diagnostic tool's request message.	0 ms	N/A

Note) Flash reprogramming session considers some timing parameter as an exception.
In that case defined separately.

6.2 Physical Communication Timing Behavior

(1) If the diagnostic tool's network layer has successfully transmitted a service request message onto the bus, the diagnostic tool application shall start its P2CAN_Tester timer.

(2) The ECU's diagnostic application layer shall start with its response message within P2CAN_ECU after the reception of Data Indication. This means that in case of a multi-frame response message the First Frame has to be sent within P2CAN_ECU and in case of a single frame response messages the Single Frame has to be sent within P2CAN_ECU.

(3) If the diagnostic tool application receives a Data FF Indication or Data Indication of its network layer, it shall stop its P2CAN_Tester (P2*CAN_Tester) timer.

(4) In case the ECU cannot provide the requested information within the P2CAN_ECU response timing, it shall request an Enhanced Response Timing window by sending a Negative Response message including response code ($78) to the diagnostic tool. The reception of this message causes the diagnostic tool to restart its P2CAN_Tester timer, but using the enhanced reload value P2*CAN_Tester.

(5) The ECU shall start with its response message within the enhanced P2*CAN_ECU following the Data confirm of the previously transmitted Negative Response message with Negative Response Code $78. In case the ECU's diagnostic application can still not provide the requested information within the enhanced P2*CAN_ECU then a further negative response message including Negative Response Code $78 shall be sent by the ECU.

(6) As long as the diagnostic tool receives Negative Response messages with Negative Response Code $78 it shall restart its P2CAN_Tester timer again using the enhanced reload value P2*CAN_Tester.

6.3 Functional Communication Timing Behavior

(1) In comparison with the physical communication scenario the functional scenario only requires changes in the diagnostic tool's timing behavior.

(2) If the diagnostic tool's network layer has successfully transmitted a service request message onto the bus, the diagnostic tool application shall start its P2CAN_Tester timer.

(3) When receiving Data FF Indication or Data Indication of an incoming response message, the diagnostic tool's application shall stop its P2CAN_Tester timer in case it knows the number of ECUs to be expected to respond and all these ECUs have responded. And, it shall restart its P2CAN_Tester timer in case not all expected ECUs responded yet or in case the diagnostic application does not know the number of ECUs to be expected to respond.

(4) When receiving a negative response message with response code $78, the diagnostic tool's application shall restart its P2CAN_Tester timer, using the enhanced reload value P2*CAN_Tester. In addition it has to grasp all ECUs which send negative response messages with Negative Response Code $78.

(5) Once an ECU stored as pending starts with its final response message (positive response message or negative response message including a response code other than Negative Response Code $78), it shall be deleted from the list of pending response messages.
In case no further response messages are pending, the diagnostic tool's application reuses the default reload value for its P2CAN_Tester timer.
As long as there is at least one ECU stored in the list of pending response messages, any start of a further response messages from any ECU will cause a restart of the P2CAN_Tester timer using the enhanced reload value P2*CAN_Tester.

6.4 Session Layer Timing Parameter Definitions

A specific diagnostic session enables a certain set of services, functionality and timing within the ECU. The diagnostic tool starts the respective diagnostic session by means of the Diagnostic Session Control service (see section 8.1) in one individual ECU or a group of ECUs. When a diagnostic session other than the Default Session is started then a session handling is required.

The session layer timing parameter values for session handling is shown in Table6.2.

Table6.2 Session Layer Timing Parameter Definitions

Timing Parameter	Description	Min	Max
S3 ECU	Time for the ECU to keep a diagnostic session other than the Default Session active while not receiving any diagnostic request message.	N/A	5000 ms

A session timer required in ECU can make only one Diagnostic session active with one ECU.

After starting a specific diagnostic session other than the Default Session, the diagnostic tool transmits the Tester Present ($3E) to keep its diagnostic session active.

In case ECUs do not receive a Diagnostic Request message within 5000ms, each ECU consider as time-out, and return in a Default Session. And this is applied to all Diagnostic requests in addition to Tester Present ($3E) and Start Diagnostic Session ($10).

6.5 Error Handling

The definition of ECU error handling is shown in table 6.3.

Table6.3 ECU Error Handling

Communication phase	ECU error type	ECU handling
Request reception	Data Indication from Network Layer with a Negative result value.	Restart S3 ECU timer, because it has been stopped based on the previously received First Frame indication. The ECU shall ignore the request.
P2 CAN_ECU	Timeout	N/A
Response transmission	Data Confirm from Network Layer with a Negative result value.	Restart S3ECU timer, because it has been stopped based on the previously received Request Message. The ECU shall not perform a re-transmission of the response message.

7. List of Diagnostic Services

The list of Diagnostic Service IDs defined by this specification is shown in Table6.3.

Table6.3 Diagnostic Services List

Diagnostic Service Name	Service ID	Details
Diagnostic and Communication Management functional unit		
Diagnostic Session Control	$10	Section 8.1
Security Access	$27	Section 8.2
Tester Present	$3E	Section 8.3
Data Transmission functional unit		
Read Data By Identifier	$22	Section 8.8
Stored Data Transmission functional unit		
Clear Diagnostic Information	$14	Section 8.11
Read DTC Information	$19	Section 8.12
Input/Output Control functional unit		
Input Output Control By Identifier	$2F	Section 8.13

8. Diagnostic Services Specification

8.1 Diagnostic Session Control (Service ID: $10)

The Diagnostic Session Control service shall be used to enable different diagnostic sessions in one ECU or a group of ECUs. A diagnostic session defines a specific set of applicable diagnostic services, amount of functionality and application layer timing parameter values (P2CAN_ECU / P2*CAN_ECU).

8.1.1 Communication Protocol Requirement

The ECU shall meet the request and response message behavior as specified in section 5.1: the Request Messages with Sub Function Parameter.

8.1.1.1 Diagnostic Request

The format of the Diagnostic Session Control Request Message is shown in Table8.1.

Table8.1 Request Message Definition - Diagnostic Session Control

Data byte no.	Parameter name	Message Usage	Data value
1	Diagnostic Session Control Request Service Id	M	$10
2	Sub Function = [Diagnostic Session Type]	M	$00 - $FF
	Default Session – Positive Response Required		$01
	Extended Diagnostic Session – Positive Response Required		$03

(1) Diagnostic Session Type Definition

The Diagnostic Session Types are shown in Table8.2.

Table8.2 Diagnostic Session Types

Diagnostic Session type	Description
Default Session	An ECU shall always start the default diagnostic session when powered up. If no other diagnostic session is requested, then the default diagnostic session shall be running as long as the ECU is powered.
Extended Diagnostic Session	This session type shall support the all diagnostic services used by an ECU. Generally vehicle diagnosis utilizes this session when inspecting vehicles. (The diagnostic tool shifts to a Extended Diagnostic Session from a default session at need.)

8.1.1.2 Diagnostic Response

The format of the Diagnostic Session Control Positive Response Message is shown in Table8.3. If an ECU responds positively to a Diagnostic Session Control Request, the Diagnostic Session Type in the response shall match the Diagnostic Session Type sent in the request.

Table8.3 Positive Response Message Definition - Diagnostic Session Control

Data byte no.	Parameter name	Message Usage	Data value
1	Diagnostic Session Control Response Service Id	M	$50
2	Sub Function = [Diagnostic Session Type]	M	$00 - $FF
 3 : : n	Session Parameter Record [] = [P2CAN_ECU_max (high byte) P2CAN_ECU_max (low byte) P2*CAN_ECU_max (high byte) P2*CAN_ECU_max (low byte)]	 M : : M	 $00 - $FF : : $00 - $FF

(1) The scaling format of the Application Layer Timing Parameter reported by Session Parameter Record is shown in Table 8.4.

Table8.4 Scaling Application Layer Timing Parameter Definition

Parameter	No. Of Bytes	Resolution	Min Value	Max Value
P2Can_ECU_max	2	1 ms/bit	0 ms	65535 ms
P2*Can_ECU_max	2	10 ms/bit	0 ms	655350 ms

The format of the Diagnostic Session Control Negative Response Message is shown in Table8.5.

Table8.5 Negative Response Message Definition - Diagnostic Session Control

Data byte no.	Parameter name	Message Usage	Data value
1	Negative Response	M	$7F
2	Diagnostic Session Control	M	$10
3	Sub-function = [Negative Response Trouble Code]	M	$00 - $FF
	Sub Function Not Supported		$12
	Incorrect Message Length Or Invalid Format		$13
	Conditions Not Correct		$22

8.1.2 General Requirements

(1) Each non-default session shall be tied to the diagnostic session timer. (For detailed requirements concerning session timing please see section 6.4 Session Layer Timing Parameter Definitions)

(2) If a diagnostic session requested by the diagnostic tool is already running, then the ECU shall send a positive response message as if the request was the first.

(3) Whenever the diagnostic tool requests a new diagnostic session, the ECU shall first send a positive response message before the new session becomes active in the ECU.

(4) If the ECU is not able to start the requested new diagnostic session, then it shall respond with a negative response message and the current session shall continue.

8.1.3 Support of Diagnostic Services

In each Diagnostic session, Diagnostic Service IDs that an ECU can support are shown in Table8.6. Within one specific diagnostic session only a predefined set of diagnostic services and/or functionality and application layer timing parameters shall be activated.

Table8.6 Diagnostic Sessions Supporting Specific Sets Of Diagnostic Services

Service Name	Service ID	Default Session	Programming Session	Extended Diagnostic Session	Stand By Session
ISO 15031-5 service identifier	$01 - $09	X (C1)		X (C1)	
Diagnostic Session Control	$10	X	X	X	X
Security Access	$27		X	X	X
Tester Present	$3E	X	X	X	X
Read Data By Identifier	$22	X (C2)	X	X	X
Clear Diagnostic Information	$14	X		X	
Read DTC Information	$19	X		X	
Input Output Control By Identifier	$2F			X	X

X: The specific diagnostic service is allowed while ECU is running in the respective diagnostic session.

C1: An ECU supports according to legislative regulations all ISO 15031-5 related service IDs if the ECU is OBD II / EOBD compliant.

C2: The data that is necessary the ECU's security feature in Default Session shall be disabled.

As the security access service is not allowed during default session, only diagnostic services that do not need deactivating of the ECU's security feature before are allowed.
If each session supports specific timing parameters (P2CAN_ECU, P2*CAN_ECU), the ECU shall be reported within the positive response message to the diagnostic tool.

8.1.4 Session transition requirements

A diagnostic session transition diagram is shown in Figure8.1.

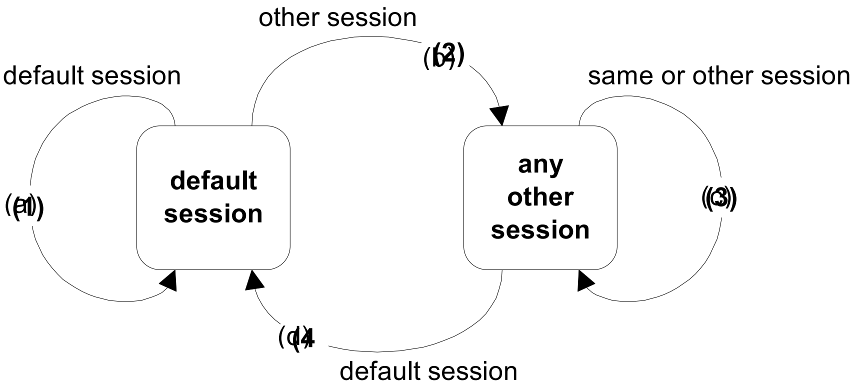


Figure8.1 ECU Diagnostic Session State Diagram

Only the depicted session state transitions are allowed.
- The ECU transitions from the Default session to another session.
- The ECU transitions from non-default session to another session or same session.

(1) If Default session is actually active and the diagnostic tool requests the ECU to start the Default session again the ECU shall re-initialize the Default session completely. This means the ECU shall reset all activated, initiated or changed settings and controls during the activated session. This does not include changes programmed into non-volatile memory. (See Figure8.1 (a))

(2) When the ECU transitions from any diagnostic session other than the Default session to another session other than the Default session, then the ECU shall (re-) initialize the diagnostic session. This means the security feature shall be (re-) enabled. (See Figure8.1 (c))

(3) When the ECU transitions from any diagnostic session other than the Default session to the Default session, then the ECU shall initialize the new diagnostic session. The ECU shall reset all activated, initiated, and changed settings and controls during the activated non-default session. This does not include changes programmed into non-volatile memory. (See Figure8.1 (d))

8.2 Security Access (Service ID: $27)

The Security Access service is used to access diagnostic services, which have restricted access for security, emissions, or safety reasons.

8.2.1 Communication Protocol Requirement

The ECU shall meet the request and response message behavior as specified in section 5.1: the Request Messages with Sub Function Parameter.

8.2.1.1 Diagnostic Request

The format of the Security Access Request Message is shown in Table8.7.

Table8.7 Request Message Definition - Security Access

Data byte no.	Parameter name	Message Usage	Data value
1	Security Access Request Service Id	M	$27
2	Sub Function = [Security Access Type]	M	$00 - $FF
	Request Seed – Positive Response Required		$01
	Send Key – Positive Response Required		$02
3 : n	Security Key [] = [Key byte no. 1 (high byte) : Key byte no. m (low byte)]	C : C	$00 - $FF : $00 - $FF

C: This parameter is only present if the diagnostic tool sends Key to an ECU.

Use Security Access Type by the following range.

(a) $01 - $7D (range of an odd number) are reserved for Request Seed.
(b) $02 - $7E (range of an even number) are reserved for Send Key.
(c) The values except the above are reserved for future definition.

(1) Security Access Type Definition

The Security Access Types are shown in Table8.8.

Table8.8 Security Access Types

Security Access type	Description
Request Seed	This type specifies that the ECU shall transmit to the diagnostic tool the seed value with a defined security level for calculating an appropriate key value to unlock the ECU. Different request type values represent different security levels and formats.
Send Key	This type signalizes to the ECU that the diagnostic tool transmits now the appropriate key value associated with the seed requested before. Different request type values represent different security levels and formats.

(2) Security Key Definition

The Security Key is shown in Table8.9.

Table8.9 Security Key

Parameter type	Description
Security Key	The “Key” parameter in the request message is the value generated by the security algorithm corresponding to a specific “Seed” value.

8.2.1.2 Diagnostic Response

The format of the Security Access Positive Response Message is shown in Table8.10.
If an ECU responds positively to a Security Access request, the Security Access Type in the response shall match the Security Access Type sent in the request.

Table8.10 Positive Response Message Definition - Security Access

Data byte no.	Parameter name	Message Usage	Data value
1	Read Memory By Address Response Service Id	M	$67
2	Security Access Type	M	$00 - $FF
	Security Seed [] = [		
3	Seed byte no. 1 (high byte)	C	$00 - $FF
:	:	:	:
n	Seed byte no. m (low byte)	C / O	$00 - $FF
	]		

C: This parameter is only present if an ECU sends Seed to the diagnostic tool.

(1) Security Seed Definition
The Security Seed is shown in Table8.11.

Table8.11 Security Seed

Parameter type	Description
Security Seed	The "Seed" parameter is a data value sent by the ECU and is used by the diagnostic tool when calculating the "Key" needed to access security.

The format of the Security Access Negative Response Message is shown in Table8.12.

Table8.12 Negative Response Message Definition - Security Access

Data byte no.	Parameter name	Message Usage	Data value
1	Negative Response	M	$7F
2	Security Access	M	$27
3	Sub-function = [Negative Response Trouble Code]	M	$00 - $FF
	Sub-Function Not Supported		$12
	Incorrect Message Length – Invalid Format		$13
	Conditions Not Correct		$22
	Request Sequence Error		$24
	Invalid Key		$35
	Exceeded Number Of Attempts		$36
	Required Time Delay Not Expired		$37

8.2.2 General Requirements

(1) Deactivating of ECU's security feature is only allowed in non-default sessions.

(2) Every diagnostic session transition forces the ECU to enable the ECU's security feature if it was disabled before.

(3) The Seed and Key values shall each be a minimum of 2 bytes in length. However, selection of the minimum number of bytes will result in a minimum-security level. Use of 4 or more bytes is recommended.

(4) The following procedure shall be used for “unlocking” an ECU:

1. A non-default session is active in the ECU by the diagnostic tool request.
2. The diagnostic tool shall send the Security Access Request ($27) service requesting the Seed parameter (sub-function: Request Seed).
3. The ECU shall respond by sending the Seed parameter value using the Security Access Positive Response ($67) service.
4. The diagnostic tool shall respond by returning the calculated Key value back to the ECU using the Security Access Request ($27) service (sub-function: Send Key).
5. The ECU shall compare this Key value to one internally calculated.
   If the two numbers match, then the ECU shall enable ("unlock") the diagnostic tool's access to specific services. This shall be indicated to the diagnostic tool via the Security Access Positive Response ($67) service.

(5) If an ECU supports security, but is already unlocked when a Security Access (sub-function: Request Seed) message is received, that ECU shall respond with a Security Access Positive Response message service with a seed value equal to zero (0).

(6) The ECU shall implement a security delay timer as follows:

1. After 2 unsuccessful attempts of Security Access procedures by the diagnostic tool, the ECU shall insert a 10 second time delay before allowing further attempts.
2. A 10 second time delay shall be required before the ECU responds positively to a Security Access Request ($27) service (sub-function: Request Seed) from the diagnostic tool after ECU power-up / reset if there have been 2 unsuccessful attempts of Security Access before.
3. After a successful Security Access service execution, a delay timer on a power-up / reset shall be cleared by the ECU.
4. If the last 2 Security Access services prior to the power-up / reset have failed, then the delay timer shall always be active after power up/reset.

(7) Even if the delay timer is active, the Diagnostic requests without Security Access and normal communications shall be possible.

(8) A Security Seed parameter value equal to $FF - $FF (all bytes are equal to $FF) shall not be used because this may occur if the ECU's memory has been erased.

8.2 Tester Present (Service ID: $3E)

The Tester Present service is used to indicate to an ECU or a group of ECUs that a diagnostic tool is still connected to the vehicle and that certain diagnostic services and/or communication that have been previously activated are to remain active.

8.2.1 Communication Protocol Requirement

The ECU shall meet the request and response message behavior as specified in section 5.1: the Request Messages with Sub Function Parameter.

8.2.1.1 Diagnostic Request

The format of the Tester Present Request Message is shown in Table8.13.

Table8.13 Request Message Definition - Tester Present

Data byte no.	Parameter name	Message Usage	Data value
1	Tester Present Request Service Id	M	$3E
2	Sub Function = [Zero Sub-Function]	M	$XX
	Positive Response Required		$00
	No Positive Response Required		$80

8.2.1.2 Diagnostic Response

The format of the Tester Present Positive Response Message is shown in Table8.14.

Table8.14 Positive Response Message Definition - Tester Present

Data byte no.	Parameter name	Message Usage	Data value
1	Tester Present Response Service Id	M	$7E
2	Sub Function = [Zero Sub-Function]	M	$00

The format of the Tester Present Negative Response Message is shown in Table8.15.

Table8.15 Negative Response Message Definition - Tester Present

Data byte no.	Parameter name	Message Usage	Data value
1	Negative Response	M	$7F
2	Tester Present	M	$3E
3	Sub-function = [Negative Response Trouble Code]	M	$00 - $FF
	Sub Function Not Supported		$12
	Incorrect Message Length – Invalid Format		$13

8.4 Read Data By Identifier (Service ID: $22)

The Read Data By Identifier service is used to read the data (e.g. input/output signals, internal parameters, system information) supported by the ECU.

8.4.1 Communication Protocol Requirement

The ECU shall meet the request and response message behavior as specified in section 5.2: the Request Messages without Sub Function Parameter.

8.4.1.1 Diagnostic Request

The format of the Read Data By Identifier Request Message is shown in Table8.16.

Table8.16 Request Message Definition - Read Data By Identifier

Data byte no.	Parameter name	Message Usage	Data value
1	Read Data By Identifier Request Service Id	M	$22
 2 3	Data Identifier []= [Byte 1 (MSB) Byte 2]	 M M	 $00 - $FF $00 - $FF

(1) Data Identifier Definition

The Data Identifier is shown in Table8.17.

Table8.17 Data Identifier

Parameter type	Description
Data Identifier	This parameter identifies the ECU's data record that is being requested by the diagnostic tool. The Data Identifiers used to this service are defined in the Part Specification: ES-W45263 (Diagnostic on CAN - Standardized).

8.4.1.2 Diagnostic Response

The format of the Read Data By Identifier Positive Response Message is shown in Table8.18.

If an ECU responds positively to a Read Data By Identifier request, the response message Data Identifier parameter value shall be an echo of the value provided in the request message by the diagnostic tool.

The parameter of the Data Record shall be used to provide the data associated with the specified Data Identifier to the diagnostic tool.

Table8.18 Positive Response Message Definition - Read Data By Identifier

Data byte no.	Parameter name	Message Usage	Data value
1	Read Data By Identifier Response Service Id	M	$62
 2 3	Data Identifier []= [Byte 1 (MSB) Byte 2]	 M M	 $00 - $FF $00 - $FF
 4 : (k-1)+4	Data Record [] = [Data no. 1 : Data no. k]	 M : O	 $00 - $FF : $00 - $FF

The format of the Read Data By Identifier Negative Response Message is shown in Table8.19.

Table8.19 Negative Response Message Definition - Read Data By Identifier

Data byte no.	Parameter name	Message Usage	Data value
1	Negative Response	M	$7F
2	Read Data By Identifier	M	$22
3	Sub-function = [Negative Response Trouble Code]	M	$00 - $FF
	Incorrect Message Length – Invalid Format		$13
	Conditions Not Correct		$22
	Request Out Of Range		$31
	Security Access Denied		$33

8.3 Clear Diagnostic Information (Service ID: $14)

The Clear Diagnostic Information service is used to clear one or multiple diagnostic information in ECUs' memory.

The ECU shall send a positive response when the Clear Diagnostic Information service is completely processed. Additionally, the ECU shall send a positive response even if no DTCs are stored.

8.3.1 Communication Protocol Requirement

The ECU shall meet the request and response message behavior as specified in section 5.2: the Request Messages without Sub Function Parameter.

8.3.1.1 Diagnostic Request

The format of the Clear DTC Information Request Message is shown in Table8.20.

Table8.20 Request Message Definition - Clear DTC Information

Data byte no.	Parameter name	Message Usage	Data value
1	Clear Diagnostic Information Request Service Id	M	$14
2 3 4	Group Of DTC [] = [Group Of DTC High Byte Group Of DTC Middle Byte Group Of DTC Low Byte]	M	 $00 - $FF $00 - $FF $00 - $FF
	Emission Related Systems	C	$000000
	All Groups / All DTCs ($000001 - $FFFFFF)	M	$FFFFFF
	Individual DTC	M	$XXXXXX

C: If the ECU is required to meet the emission related diagnostic requirements according to ISO 15031, the ECU shall support additionally a Group of DTC parameter value equal to $000000 that means Clear all Emission Related DTCs within a emission related systems.

8.3.1.2 Diagnostic Response

The format of the Clear DTC Information Positive Response Message is shown in Table8.21.

Table8.21 Positive Response Message Definition - Clear DTC Information

Data byte no.	Parameter name	Message Usage	Data value
1	Clear Diagnostic Information Positive Response Service Id	M	$54

The format of the Clear DTC Information Negative Response Message is shown in Table8.22.

Table8.22 Negative Response Message Definition - Clear DTC Information

Data byte no.	Parameter name	Message Usage	Data value
1	Negative Response	M	$7F
2	Clear Diagnostic Information Request Service Id	M	$14
3	Sub-function = [Negative Response Trouble Code]	M	$00 - $FF
	Incorrect Message Length – Invalid Format		$13
	Conditions Not Correct		$22
	Request Out Of Range		$31

8.6 Read DTC Information (Service ID: $19)

The Read DTC Information service is used to read Diagnostic Trouble Code (DTC) information in ECUs' memory.

8.6.1 Communication Protocol Requirement

The ECU shall meet the request and response message behavior as specified in section 5.1: the Request Messages with Sub Function Parameter.

8.6.1.1 Diagnostic Request

The format of the Read DTC Information Request Message is shown in Table8.23.

Table8.23 Request Message Definition - Read DTC Information

Data byte no.	Parameter name	Message Usage	Data value
1	Read DTC Information Request Service Id	M	$19
2	Sub-function = [Request type]	M	$00 - $FF
	Report Number Of DTC By Status Mask - Positive Response Required		$01
	Report DTC By Status Mask - Positive Response Required		$02
3	DTC Status Mask	M	$00 - $FF

(1) Request Data Parameter Type Description

The Request Data Parameter Types are shown in Table8.24.

Table8.24 Request Data Parameter types

Request data parameter type	Description
DTC Status Mask	This parameter contains eight (8) DTC status bits. The definitions for each of the eight (8) bits can be found in Table8.25: DTC Status Byte Description. This byte is used in the request message to allow a diagnostic tool to request DTC information for the DTCs whose status matches the DTC Status Mask. A DTC's status matches the DTC Status Mask if any one of the DTC's actual status bits is set to ‘1’ and the corresponding status bit in the DTC Status Mask is also set to ‘1’. If the diagnostic tool specifies a status mask that contains bits that the ECU does not support, then the ECU shall process the DTC information using only the bits that it does support.
DTC Mask Record	This parameter contains DTC High Byte, DTC Middle Byte and DTC Low Byte. DTC High Byte, DTC Middle Byte and DTC Low Byte together represent a unique identification number for a specific diagnostic trouble code supported by an ECU. (See section 8.12.3)

(2) DTC Status Byte Description

The DTC Status Mask / Status Of DTC parameters are shown in Table8.25.

Table8.25 Status Bit Definition

Bit	Description		ECU Support Mandatory / Optional	
			Emission Related	Non-Emission Related
0	**Test Failed** This bit shall indicate the result of the most recently performed test. Reset to logical '0' if the result of the most recently performed test returns a “pass” result meaning that all de-mature criteria have been fulfilled.		M	M
	1	Most recent return from DTC test indicated a failed result.		
	0	Most recent result from DTC test indicated no failure detected.		
1	**Test Failed This Operation Cycle** This bit shall indicate whether or not a diagnostic test has reported a test Failed result at any time during the current operation cycle. Reset to logical '0' when a new operation cycle is initiated or after executing Clear Diagnostic Information service. If this bit is set to logical '1', it shall remain a '1' until a new operation cycle is started.		M	O / C1
	1	Test Failed: result was reported at least once during the current operation cycle.		
	0	Test Failed: result has not been reported during the current operation cycle or after a call was made to Clear Diagnostic Information during the current operation cycle.		
2	**Pending DTC** This bit shall indicate whether or not a diagnostic test has reported a test Failed result at any time during the current or last completed operation cycle. The status shall only be updated if the test runs and completes. The criteria to set the pending DTC (Bit=2) and the Test Failed This Operation Cycle (Bit=1) are the same. The difference is that the test Failed This Operation Cycle is cleared at the end of the current operation cycle and the pending DTC bit is not cleared until an operation cycle has completed where the test has passed at least once and never failed. If the test did not complete during the current operation cycle, the status bit shall not be changed.		M	O
	1	This bit shall be set to 1 and latched if a malfunction is detected during the current operation cycle.		
	0	This bit shall be set to 0 after completing an operation cycle during which the test completed and a malfunction was not detected or upon a call to the Clear Diagnostic Information service.		
3	**Confirmed DTC** This bit shall indicate whether a malfunction was detected enough times to warrant that the DTC is stored in long-term memory (In emission related ECUs, the Pending DTC (Bit=2) shall be set to 1 once.). In case additional diagnostic information (e.g. Snapshot data) are requested, the information of this bit is used. A confirmed DTC does not always indicate that the malfunction is present at the time of the request. (Test Failed can be used to determine if a malfunction is present at the time of the request). Reset to logical '0' after a call to Clear Diagnostic Information service or after aging criteria has been satisfied (e.g., 40 engine warm-ups without another detected malfunction). Furthermore this bit is reset when the fault record associated with this DTC is overwritten by a newer DTC based upon vehicle manufacturer specific fault memory overflow requirements.		M	M
	1	DTC confirmed at least once since the last call to Clear Diagnostic Information service and aging criteria have not yet been satisfied.		
	0	DTC has never been confirmed since the last call to Clear Diagnostic Information service or after the aging criteria have been satisfied for the DTC		

Table8.25 Status Bit Definition

Bit	Description		ECU Support Mandatory / Optional	
			Emission Related	Non-Emission Related
4	**Test Not Completed Since Last Clear** This bit shall indicate whether a DTC test has ever run and completed since the last time a call was made to Clear Diagnostic Information service. If the test runs and passes or if the test runs and fails (e.g. test Failed This Operation Cycle = '1') then the bit shall be set to a '0'. Reset to a logical ‘1’ after a call to Clear Diagnostic Information service.		M	M
	1	DTC test has not run to completion since the last time diagnostic information was cleared.		
	0	DTC test has returned either a passed or failed test result at least one time since the last time diagnostic information was cleared.		
5	**Test Failed Since Last Clear** This bit shall indicate whether a DTC test has ever returned a Test Failed This Operation Cycle (Bit=1) = 1 result since the last time a call was made to Clear Diagnostic Information service. Reset to a logical ‘0’ after a call to Clear Diagnostic Information service. Unlike confirmed DTC (Bit=3), this bit is not reset by aging-criteria or due to an overflow of the fault memory.		O	O
	1	DTC test returned a Test Failed This Monitoring Cycle = 1 results at least once since the last time diagnostic information was cleared.		
	0	DTC test has not indicated a Test Failed This Monitoring Cycle = 1 results since the last time diagnostic information was cleared.		
6	**Test Not Completed This Operation Cycle** This bit shall indicate whether a DTC test has ever run and completed during the current operation cycle. If the test runs and passes or fails then the bit shall be set to '0' until a new operation cycle is started. Reset to a logical ‘1’ after a call to Clear Diagnostic Information service.		M	O
	1	DTC test has not run to completion this Operation cycle.		
	0	DTC test has returned either a passed or Test Failed This Monitoring Cycle = 1 result during the current monitoring cycle or a call was made to Clear Diagnostic Information service.		
7	**Warning Indicator Requested** This bit shall report the status of any warning indicators associated with a particular DTC. If no warning indicators exist for a given system or particular DTC, this status shall default to a logic '0' state. Reset to a logical ‘1’ after a call to Clear Diagnostic Information service. Conditions for activating and deactivating the warning indicator are defined by the ECU's control specifications.		M	O
	1	Warning indicator requested to be ON. (Confirmed DTC (Bit=3) is set to 1)		
	0	Warning indicator requested to be OFF.		

C1: Bit 1 (Test Failed This Monitoring Cycle) is Mandatory if Bit 2 (Pending DTC) is supported. Bit 1 (Test Failed This Monitoring Cycle) is Optional if Bit 2 (Pending DTC) is not supported.

O: Optional may be supported by the ECU if applicable.

< Reference information >

The Status Mask for reading DTC information shall be used a status in Table8.26.

Table8.26 DTC Status Mask

Data value	Parameter name	Use	Remarks
$01	Active DTCs	Read Present DTCs	
$04	Pending DTCs	-	Same as Mode $07 (Request emission-related DTCs detected during current or last completed driving cycle)
$08	Confirmed / Stored DTCs	Read Past DTCs	Same as Mode $03 (Request emission-related DTCs)
$FF	All record DTCs	-	

The transitions of individual Bit Status are shown in Figure8.2.

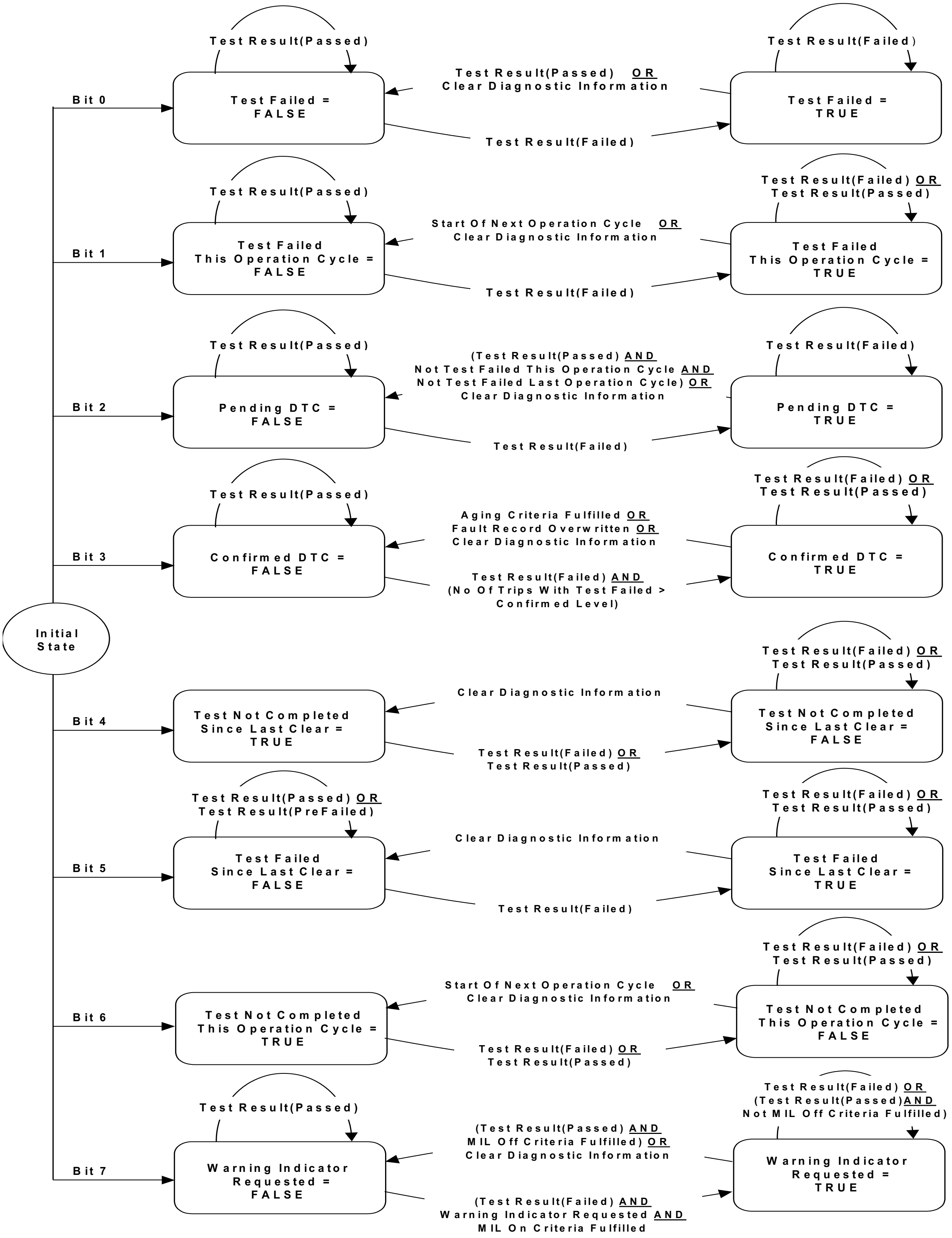

Initial State
Bit 0
Test Result(Passed)
Test Failed = FALSE
Test Result(Passed) OR Clear Diagnostic Information
Test Result(Failed)
Test Result(Failed)
Test Failed = TRUE
Bit 1
Test Result(Passed)
Test Failed This Operation Cycle = FALSE
Start Of Next Operation Cycle OR Clear Diagnostic Information
Test Result(Failed)
Test Result(Failed) OR Test Result(Passed)
Test Failed This Operation Cycle = TRUE
Bit 2
Test Result(Passed)
Pending DTC = FALSE
(Test Result(Passed) AND Not Test Failed This Operation Cycle AND Not Test Failed Last Operation Cycle) OR Clear Diagnostic Information
Test Result(Failed)
Test Result(Failed)
Pending DTC = TRUE
Bit 3
Test Result(Passed)
Confirmed DTC = FALSE
Aging Criteria Fulfilled OR Fault Record Overwritten OR Clear Diagnostic Information
Test Result(Failed) AND (No Of Trips With Test Failed > Confirmed Level)
Test Result(Failed) OR Test Result(Passed)
Confirmed DTC = TRUE
Bit 4
Test Not Completed Since Last Clear = TRUE
Clear Diagnostic Information
Test Result(Failed) OR Test Result(Passed)
Test Result(Failed) OR Test Result(Passed)
Test Not Completed Since Last Clear = FALSE
Bit 5
Test Result(Passed) OR Test Result(PreFailed)
Test Failed Since Last Clear = FALSE
Clear Diagnostic Information
Test Result(Failed)
Test Result(Failed) OR Test Result(Passed)
Test Failed Since Last Clear = TRUE
Bit 6
Test Not Completed This Operation Cycle = TRUE
Start Of Next Operation Cycle OR Clear Diagnostic Information
Test Result(Failed) OR Test Result(Passed)
Test Result(Failed) OR Test Result(Passed)
Test Not Completed This Operation Cycle = FALSE
Bit 7
Test Result(Passed)
Warning Indicator Requested = FALSE
(Test Result(Passed) AND MIL Off Criteria Fulfilled) OR Clear Diagnostic Information
(Test Result(Failed) AND Warning Indicator Requested AND MIL On Criteria Fulfilled
Test Result(Failed) OR (Test Result(Passed)AND Not MIL Off Criteria Fulfilled)
Warning Indicator Requested = TRUE

Figure8.2 Transitions of DTC Status bit

8.6.1.2 Diagnostic Response

(1) Report Number Of DTC By Status Mask

The format of the Read DTC Information Positive Response Message is shown in Table8.27.

Table8.27 Positive Response Message Definition - Read DTC Information (Report Number Of DTC)

Data byte no.	Parameter name	Message Usage	Data value
1	Read DTC Information Response Service Id	M	$59
2	Report Type	M	$00 - $FF
	Report Number Of DTC By Status Mask		$01
3	DTC Status Availability Mask	M	$00 - $FF
4	DTC Format Identifier	M	$00 - $FF
	ISO14229-1DTCFormat		$01
	DTC Count [] = [		
5	DTC Count High Byte	M	$00 - $FF
6	DTC Count Low Byte	M	$00 - $FF
	]		

(2) Report DTC By Status Mask

The format of the Read DTC Information Positive Response Message is shown in Table8.28.

Table8.28 Positive Response Message Definition - Read DTC Information (Report DTC)

Data byte no.	Parameter name	Message Usage	Data value
1	Read DTC Information Response Service Id	M	$59
2	Report Type	M	$00 - $FF
	Report DTC By Status Mask		$02
3	DTC Status Availability Mask	M	$00 - $FF
	DTC And Status Record [] = [		
4	DTC High Byte no. 1	C1	$00 - $FF
5	DTC Middle Byte no. 1	C1	$00 - $FF
6	DTC Low Byte no. 1	C1	$00 - $FF
7	Status Of DTC no. 1	C1	$00 - $FF
8	DTC High Byte no. 2	C2	$00 - $FF
9	DTC Middle Byte no. 2	C2	$00 - $FF
10	DTC Low Byte no. 2	C2	$00 - $FF
11	Status Of DTC no. 2	C2	$00 - $FF
:	:	:	:
n-3	DTC High Byte no. m	C2	$00 - $FF
n-2	DTC Middle Byte no. m	C2	$00 - $FF
n-1	DTC Low Byte no. m	C2	$00 - $FF
n	Status Of DTC no. m	C2	$00 - $FF
	]		

C1: This parameter is only present if one DTC information is available to be reported.

C2: This parameter is only present if more than one DTC information is available to be reported.

(4) Response Data Parameter Type Description

The Response Data Parameter Types are shown in Table8.29.

Table8.29 Response Data Parameter types

Response data parameter type	Description
Report Type	This parameter is an echo of the sub-function parameter provided in the request message from the diagnostic tool.
DTC Record	This parameter record contains one or more groupings of DTC High Byte, DTC Middle Byte and DTC Low Byte. About DTC format, see section 8.12.3.
Status Of DTC	The status of a particular DTC. The definition of the bits contained in the Status Of DTC byte can be found in Table8.25 (Status Bit Definition).
DTC And Status Record	This parameter record contains one or more groupings of DTC Record and Status Of DTC.
DTC Status Availability Mask	A byte whose bits are defined the same as Status Of DTC and represents the status bits that are supported by the ECU. Bits that are not supported by the ECU shall be set to 0.
DTC Format Identifier	This 1-byte parameter value defines the format (ISO 14229-1) of a DTC reported by the ECU.
DTC Count	This 2-byte parameter refers collectively to the DTC Count High Byte and DTC Count Low Byte parameters that are sent in response to a Report Number Of DTC request.

The format of the Read DTC Information Negative Response Message is shown in Table8.30.

Table8.30 Negative Response Message Definition - Read DTC Information

Data byte no.	Parameter name	Message Usage	Data value
1	Negative Response	M	$7F
2	Read DTC Information	M	$19
3	Sub-function = [Negative Response Trouble Code]	M	$00 - $FF
	Sub Function Not Supported		$12
	Incorrect Message Length – Invalid Format		$13
	Request Out Of Range		$31

8.6.2 General Requirements

(1) DTC Status Information Byte

1. If an ECU provides emissions-related functionality, all DTCs defined by legislated emissions-related regulations shall support all eight (8) bits of the DTC status.
2. All non-emissions-related DTCs shall at least support the following status bits:
   - Test Failed (Bit 0)
   - Confirmed DTC (Bit 3)
   - Test Not Completed Since Last Clear (Bit 4)
3. If an ECU is required to support a Warning Indicator either by legislation or by application specific requirements it shall support the Warning Indicator Requested bit (bit 7).

(2) Report Number Of DTCs By Status Mask (Request type = $01)

1. If the ECU is requested to Report Number Of DTC By Status Mask it shall at first scan through all supported DTCs, performing a bit-wise logical AND-ing operation between the mask specified by the diagnostic tool and the actual status of each supported DTC. For each AND-ing operation yielding a non-zero result, the ECU shall increment a counter by one (1).
2. As soon as all supported DTCs have been checked once, the ECU shall respond with the appropriate response message that includes the resulted DTC count value.
3. If no DTCs within the ECU match the masking criteria specified in the diagnostic tool's request, the count returned by the ECU to the diagnostic tool shall be 0.
4. If the diagnostic tool specifies a status mask that contains bits that the ECU does not support, then the ECU shall process the DTC information using only the bits that it does support.

(3) Report DTC By Status Mask (Request type = $02)

1. If the ECU is requested to Report DTC By Status Mask it shall at first scan through all supported DTCs, performing a bit-wise logical AND-ing operation between the mask specified by the diagnostic tool and the present status associated with each DTC supported by the ECU.
2. As soon as all supported DTCs have been checked once, the ECU shall respond with the appropriate response message that includes all DTCs for which the AND-ing operation has yielded a non-zero result. (Status Of DTC & DTC Status Mask! = 0).
3. If the diagnostic tool specifies a status mask that contains bits that the ECU does not support, then the ECU shall process the DTC information using only the bits that it does support.
4. If no DTCs within the ECU match the masking criteria specified in the diagnostic tool's request, no DTC or status information shall be provided within the positive response message.

8.6.3 Structure of DTC

DTC format (ISO14229-1DTC Format) consists of Numerical value code with Identification symbol (2 bytes) and Additional information code (1 byte).

Table8.31 DTC format

High Byte			Middle Byte		Low Byte	
Bit 7 - 6	Bit 5 – 4	Bit 3 - 0	Bit 7 - 4	Bit 3 – 0	Bit 7 - 4	Bit 3 - 0
No. (1)	No. (2)	No. (3)	No. (4)	No. (5)	No. (6)	No. (7)
Identification symbol		Numerical value code			Additional information code (Failure type)	

Table8.32 Detailed structure of each byte

Byte	No.	Display	Description	Binary
High Byte	(1)	P	Powertrain	00
		C	Chassis	01
		B	Body	10
		U	Network	11
	(2)	0	ISO/SAE controlled	00
		1	Manufacturer controlled	01
		2	ISO/SAE controlled	10
		3	ISO/SAE controlled	11
	(3)	0 - F	Numerical value code (the 1st digit)	0000 - 1111
Middle Byte	(4)		Numerical value code (the 2nd digit)	0000 - 1111
	(5)		Numerical value code (the 3rd digit)	0000 - 1111
Low Byte	(6)		Additional information code[*1] (the 1st digit)	0000 - 1111
	(7)		Additional information code[*1] (the 2nd digit)	0000 - 1111

*1) This byte shall be basically stored "$ 00" in. Refer to ISO15031-5 when using addition information.

8.7 Input Output Control By Identifier (Service ID: $2F)

Input Output Control By Identifier service is used to drive (or stop) actuators controlled by an ECU with input/output signal set by the diagnostic tool. After receiving this service, an ECU shall drive (or stop) actuators, and shall complete with a time set beforehand.
If the diagnostic tool requests another service while the ECU is executing the actuator control, the ECU shall respond its service. If an ECU can execute the required control, an ECU sends a Positive response message.

8.7.1 Communication Protocol Requirement

The ECU shall meet the request and response message behavior as specified in section 5.2: the Request Messages without Sub Function Parameter.

8.7.1.1 Diagnostic Request

The format of the Input Output Control By Identifier Request Message is shown in Table8.33.

Table8.33 Request Message Definitions - Input Output Control By Identifier

Data byte no.	Parameter name	Message Usage	Data value
1	Input Output Control By Identifier Request Service Id	M	$2F
	Data Identifier []= [		
2	Byte 1 (MSB)	M	$00-$FF
3	Byte 2	M	$00-$FF
	]		
4	Input Output Control Type	M	$00-$FF
	Return Control To ECU		$00
	Freeze Current State		$02
	Short Term Adjustment		$03
	Control Option Record [] = [		
5	Control State no. 1	C	$00-$FF
:	:	:	:
4+(m-1)	Control State no. m	C	$00-$FF
	]		
	Control Enable Mask Record no. 1 [] = [		
4+m	Control Mask no. 1	O	$00-$FF
:	:	:	:
4+m+(r-1)	Control Mask no. r	O	$00-$FF
	]		

C: The presence of those parameters depends on the respective Input Output Control Type and the number of Input / Output parameters identified by one individual Data Identifier.

(1) Data Identifier Definition

The Data Identifier is shown in Table8.34.

Table8.34 Data Identifier

Parameter type	Description
Data Identifier	This parameter identifies either a group of an ECU's local input signal, internal parameter and/or output signal or one individual input signal, internal parameter or output signal. The Data Identifiers used to this service are defined in the Part Specification: ES-W45263 (Diagnostic on CAN - Standardized).

(2) Input Output Control Type Definition

The Input Output Control Types are shown in Table8.35.

Table8.35 Input Output Control Types

Input Output Control Type	Description
Return Control To ECU	This type shall indicate to the ECU that it is requested to reset the I/O Control referenced by the Data Identifier to ECU's normal control (A control state before requesting the Short Term Adjustment.). (Number of Control State bytes in request: 0)
Freeze Current State	This type shall indicate to the ECU that it is requested to freeze the current state of the I/O Control referenced by the Data Identifier. (Number of Control State bytes in request: 0)
Short Term Adjustment	This type shall indicate to the ECU that it is requested to adjust the I/O Control referenced by the Data Identifier to the value included in the Control Option parameter. (Number of Control State bytes in request: depends on a particular Data Identifier)

(3) Control Option Record Definition

The Control Option Record of each Data Identifier consists of one or multiple bytes with each byte or bit representing the respective adjustment value associated with one individual input signal, internal parameter or output signal out of the group of parameters. In the case of bit value adjustments, the assignment of adjustment values starts always with the MSB of Control State no. 1.

(4) Control Enable Mask Record Definition

Do not use it at present.

8.7.1.2 Diagnostic Response

The format of the Input Output Control By Identifier Positive Response Message is shown in Table8.36.

Table8.36 Positive Response Message Definition - Input Output Control By Identifier

Data byte no.	Parameter name	Message Usage	Data value
1	Input Output Control By Identifier Response Service Id	M	$6F
 2 3	Data Identifier no 1 []= [Byte 1 (MSB) Byte 2]	 M M	 $00-$FF $00-$FF
4	Input Output Control Type	M	$00-$03
 5 : 4+(m-1)	Control Status Record [] = [Control State no. 1 : Control State no. m]	 C : C	 $00-$FF : $00-$FF

C: The presence of those parameters depends on the number of Input / Output parameters identified by one individual Data Identifier.

The format of the Input Output Control By Identifier Negative Response Message is shown in Table8.37.

Table8.37 Negative Response Message Definition - Input Output Control By Identifier

Data byte no.	Parameter name	Message Usage	Data value
1	Negative Response	M	$7F
2	Input Output Control By Identifier	M	$2F
3	Sub-function = [Negative Response Trouble Code]	M	$00-$FF
	Incorrect Message Length – Invalid Format		$13
	Conditions Not Correct		$22
	Request Out Of Range		$31
	Security Access Denied		$33

8.7.2 Message flow example of the Input Output Control By Identifier service

(1) In case that an ECU memorizes a control pattern
In case that the Data Identifier does not include Control State Bytes (an ECU memorizes a control pattern), the ECU shall send a Control State Byte including a content of Table8.38 in the Positive Response Message.

Table8.38 Control Status

Control State Byte	Description
$00	Controlled by ECU as a Normal Operation (Not execute I/O Control)
$FF	Controlled by ECU a Tester (Execute I/O Control)

Example 1) Case of executing a control of a VGT M/V (Data Identifier: $D001)

<Step1: Freeze Current State>

Table8.39 Report Current State for I/O Local Identifier example ($D001)

(a) Request Message

Data byte no.	Parameter name	Data value
1	Input Output Control By Identifier Request Service Id	$2F
2	Data Identifier (MSB)	$D0
3	Data Identifier	$01
4	Input Output Control Type (Freeze Current State)	$02

(b) Response Message - Positive Response

Data byte no.	Parameter name	Data value
1	Input Output Control By Identifier Request Service Id	$6F
2	Data Identifier (MSB)	$D0
3	Data Identifier	$01
4	Input Output Control Type (Freeze Current State)	$02
5	Control State no. 1 (Current VGT M/V Control Status =OFF)	$00

(c) Response Message - Negative Response

Data byte no.	Parameter name	Data value
1	Negative Response	$7F
2	Input Output Control By Identifier	$2F
3	Sub-function = [Negative Response Trouble Code]	$XX

<Step2: Short Term Adjustment>

Table8.40 Short Term Adjustment for I/O Local Identifier example ($D001)

(a) Request Message

Data byte no.	Parameter name	Data value
1	Input Output Control By Identifier Request Service Id	$2F
2	Data Identifier (MSB)	$D0
3	Data Identifier	$01
4	Input Output Control Type (Short Term Adjustment)	$03

(b) Response Message - Positive Response

Data byte no.	Parameter name	Data value
1	Input Output Control By Identifier Request Service Id	$6F
2	Data Identifier (MSB)	$D0
3	Data Identifier	$01
4	Input Output Control Type (Short Term Adjustment)	$03
5	Control State no. 1 (Current VGT M/V Control Status =ON)	$FF

(c) Response Message - Negative Response

Data byte no.	Parameter name	Data value
1	Negative Response	$7F
2	Input Output Control By Identifier	$2F
3	Sub-function = [Negative Response Trouble Code]	$XX

<Step3: Return Control To ECU>

Table8.41 Return Control To ECU for I/O Local Identifier example ($D001)

(a) Request Message

Data byte no.	Parameter name	Data value
1	Input Output Control By Identifier Request Service Id	$2F
2	Data Identifier (MSB)	$D0
3	Data Identifier	$01
4	Input Output Control Type (Return Control To ECU)	$00

(b) Response Message - Positive Response

Data byte no.	Parameter name	Data value
1	Input Output Control By Identifier Request Service Id	$6F
2	Data Identifier (MSB)	$D0
3	Data Identifier	$01
4	Input Output Control Type (Return Control To ECU)	$00

(c) Response Message - Negative Response

Data byte no.	Parameter name	Data value
1	Negative Response	$7F
2	Input Output Control By Identifier	$2F
3	Sub-function = [Negative Response Trouble Code]	$XX

(2) In case of executing a variable control

In case that the Data Identifier includes Control State Bytes (a control pattern provided by the diagnostic tool), the ECU shall send a Control State Byte including a content of a current value (a target value in RAM) in the Positive Response Message.

Example 2) Case of controlling an EGR (Data Identifier: $D002)

Table8.42 Input Control Condition for EGR

Data Identifier	Control State no. 1	Control State no. 2
EGR = $D002	$XX: EGR Control State - $00: Controlled by ECU (Normal Operation) - $01: ON	$YY: Control Percentage - $00: Controlled by ECU (Normal Operation) - $XX: XX %

<Step1: Freeze Current State>

Table8.43 Report Current State for I/O Local Identifier example ($D002)

(a) Request Message

Data byte no.	Parameter name	Data value
1	Input Output Control By Identifier Request Service Id	$2F
2	Data Identifier (MSB)	$D0
3	Data Identifier	$02
4	Input Output Control Type (Freeze Current State)	$02

(b) Response Message - Positive Response

Data byte no.	Parameter name	Data value
1	Input Output Control By Identifier Request Service Id	$6F
2	Data Identifier (MSB)	$D0
3	Data Identifier	$02
4	Input Output Control Type (Freeze Current State)	$02
5	Control State no. 1 (Current EGR Control State =OFF)	$00
6	Control State no. 2 (Current EGR Valve Position =50%)	$32

(c) Response Message - Negative Response

Data byte no.	Parameter name	Data value
1	Negative Response	$7F
2	Input Output Control By Identifier	$2F
3	Sub-function = [Negative Response Trouble Code]	$XX

<Step2: Short Term Adjustment>

Table8.44 Short Term Adjustment for I/O Local Identifier example ($D002)

(a) Request Message

Data byte no.	Parameter name	Data value
1	Input Output Control By Identifier Request Service Id	$2F
2	Data Identifier (MSB)	$D0
3	Data Identifier	$02
4	Input Output Control Type (Short Term Adjustment)	$03
5	Control State no. 1 (Desired EGR Control State =ON)	$01
6	Control State no. 2 (Desired EGR Valve Position =100%)	$64

(b) Response Message - Positive Response

Data byte no.	Parameter name	Data value
1	Input Output Control By Identifier Request Service Id	$6F
2	Data Identifier (MSB)	$D0
3	Data Identifier	$02
4	Input Output Control Type (Short Term Adjustment)	$03
5	Control State no. 1 (Desired EGR Control State =ON)	$01
6	Control State no. 2 (Desired EGR Valve Position =100%)	$64

(c) Response Message - Negative Response

Data byte no.	Parameter name	Data value
1	Negative Response	$7F
2	Input Output Control By Identifier	$2F
3	Sub-function = [Negative Response Trouble Code]	$XX

<Step3: Return Control To ECU>

Table8.45 Return Control To ECU for I/O Local Identifier example ($D002)

(a) Request Message

Data byte no.	Parameter name	Data value
1	Input Output Control By Identifier Request Service Id	$2F
2	Data Identifier (MSB)	$D0
3	Data Identifier	$02
4	Input Output Control Type (Return Control To ECU)	$00

(b) Response Message - Positive Response

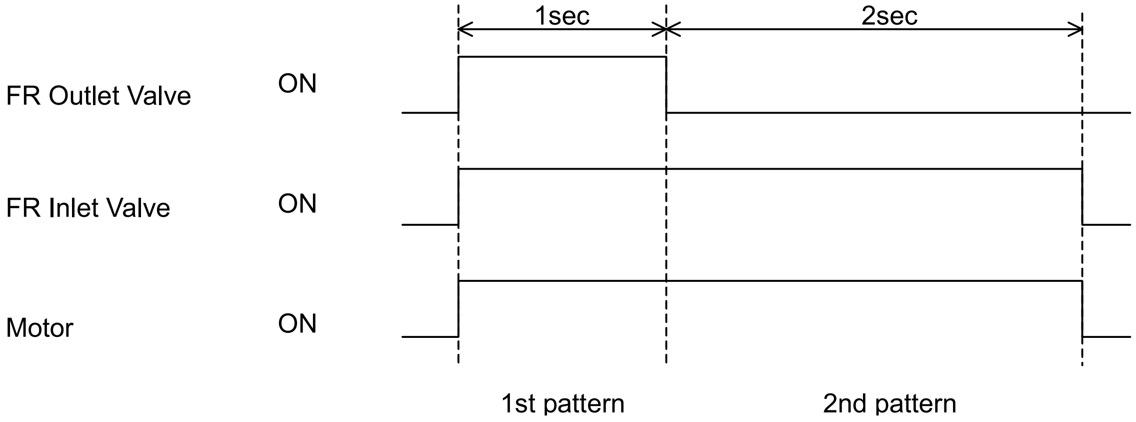
Data byte no.	Parameter name	Data value
1	Input Output Control By Identifier Request Service Id	$6F
2	Data Identifier (MSB)	$D0
3	Data Identifier	$02
4	Input Output Control Type (Return Control To ECU)	$00

(c) Response Message - Negative Response

Data byte no.	Parameter name	Data value
1	Negative Response	$7F
2	Input Output Control By Identifier	$2F
3	Sub-function = [Negative Response Trouble Code]	$XX

(3) In case of executing a multiple control at the same time
In case that the Data Identifier includes Control State Bytes (a control pattern set by the diagnostic tool optionally), the ECU shall send a Control State Byte including a content of a current value (a target value in RAM) in the Positive Response Message.

Example 3) Case of controlling ABS Solenoid Valves and Motor (Data Identifier: $D003) at the same time during defined time.



Parameter	Control State	Description	Control State No.
Number of Pattern	$NN	2 pattern (= $02)	1
Valve Selection	$XX	FR Outlet Valve =ON, FR Inlet Valve =ON (=$11)	2
Motor Control	$YY	Motor ON (= $01)	3
Duration	$ZZ	1sec (= $64) (*1)	4
Valve Selection	$XX	FR Inlet Valve =ON (=$01)	5
Motor Control	$YY	Motor ON (= $01)	6
Duration	$ZZ	2sec (= $C8) (*1)	7

*1) 1bit = 10ms

Figure 8.3: I/O Control Conditions for ABS Solenoid Valve & Motor

Table8.46 Bit Encoding Control State of I/O Local Identifier example ($D003)

Control State Name	Bit Encode	State (bit = 1:ON, bit = 0:OFF)
Valve Selection	B7	RL Outlet Valve
	B6	RR Outlet Valve
	B5	FL Outlet Valve
	B4	FR Outlet Valve
	B3	RL Inlet Valve
	B2	RR Inlet Valve
	B1	FL Inlet Valve
	B0	FR Inlet Valve
Motor	B0	Motor

<Step1: Freeze Current State>

Table8.47 Report Current State for I/O Local Identifier example ($D003)

(a) Request Message

Data byte no.	Parameter name	Data value
1	Input Output Control By Identifier Request Service Id	$2F
2	Data Identifier (MSB)	$D0
3	Data Identifier	$03
4	Input Output Control Type (Freeze Current State)	$02

(b) Response Message - Positive Response

Data byte no.	Description	Data value
1	Input Output Control By Identifier Request Service Id	$6F
2	Data Identifier (MSB)	$D0
3	Data Identifier	$03
4	Input Output Control Type (Freeze Current State)	$02
5	Control State no. 1 (Controlled by the ECU as a Normal State)	$00

(c) Response Message - Negative Response

Data byte no.	Parameter name	Data value
1	Negative Response	$7F
2	Input Output Control By Identifier	$2F
3	Sub-function = [Negative Response Trouble Code]	$XX

<Step2: Short Term Adjustment>

Table8.48 Short Term Adjustment for I/O Local Identifier example ($D003)

(a) Request Message

Data byte no.	Parameter name	Data value
1	Input Output Control By Identifier Request Service Id	$2F
2	Data Identifier (MSB)	$D0
3	Data Identifier	$03
4	Input Output Control Type (Short Term Adjustment)	$03
5	Control State no. 1 (Number of Pattern = 2)	$02
6	Control State no. 2 (Valve Selection = FR Outlet & Inlet Valve =ON)	$11
7	Control State no. 3 (Motor Control = ON)	$01
8	Control State no. 4 (Duration = 1sec)	$64
9	Control State no. 5 (Valve Selection = FR Inlet Valve =ON)	$01
10	Control State no. 6 (Motor Control = ON)	$01
11	Control State no. 7 (Duration = 2sec)	$C8

(b) Response Message - Positive Response

Data byte no.	Parameter name	Data value
1	Input Output Control By Identifier Request Service Id	$6F
2	Data Identifier (MSB)	$D0
3	Data Identifier	$03
4	Input Output Control Type (Short Term Adjustment)	$03
5	Control State no. 1 (Number of Pattern = 2)	$02
6	Control State no. 2 (Valve Selection = FR Outlet & Inlet Valve =ON)	$11
7	Control State no. 3 (Motor Control = ON)	$01
8	Control State no. 4 (Duration = 1sec)	$64
9	Control State no. 5 (Valve Selection = FR Inlet Valve =ON)	$01
10	Control State no. 6 (Motor Control = ON)	$01
11	Control State no. 7 (Duration = 2sec)	$C8

(c) Response Message - Negative Response

Data byte no.	Parameter name	Data value
1	Negative Response	$7F
2	Input Output Control By Identifier	$2F
3	Sub-function = [Negative Response Trouble Code]	$XX

<Step3: Return Control To ECU>

Table8.49 Return Control To ECU for I/O Local Identifier example ($D003)

(a) Request Message

Data byte no.	Parameter name	Data value
1	Input Output Control By Identifier Request Service Id	$2F
2	Data Identifier (MSB)	$D0
3	Data Identifier	$03
4	Input Output Control Type (Return Control To ECU)	$00

(b) Response Message - Positive Response

Data byte no.	Parameter name	Data value
1	Input Output Control By Identifier Request Service Id	$6F
2	Data Identifier (MSB)	$D0
3	Data Identifier	$03
4	Input Output Control Type (Return Control To ECU)	$00

(c) Response Message - Negative Response

Data byte no.	Parameter name	Data value
1	Negative Response	$7F
2	Input Output Control By Identifier	$2F
3	Sub-function = [Negative Response Trouble Code]	$XX

9. Negative Response Code

The Negative Response Code ($00 - $FF) used by this specification is divided into 3 ranges as Table9.1.

Table9.1 Negative Response Code grouping

Hex value	Description
$00	Positive response parameter value for ECU internal implementation.
$01 - $7F	Communication related negative response codes.
$80 - $FF	Negative response codes for specific conditions that are not correct at the point in time the request are received by the ECU.

All Negative Response Codes used by each Diagnostic Service are shown in Table9.2.

Table9.2 Definition of Negative Response Code Values

Hex value	Description
$01 - $10	Reserved
$11	**Service Not Supported** This response code indicates that the requested action will not be taken because the ECU does not support the requested service. This negative response code is not shown in the list of negative response codes to be supported for a diagnostic service, because this negative response code is not applicable for supported services.
$12	**Sub Function Not Supported** This response code indicates that the requested action will not be taken because the ECU does not support the service specific parameters of the request message.
$13	**Incorrect Message Length or Invalid Format** This response code indicates that the requested action will not be taken because the length of the received request message does not match the prescribed length for the specified service or the format of the parameters do not match the prescribed format for the specified service.
$14 - $20	Reserved
$21	**Busy Repeat Request** This response code indicates that the ECU is temporarily too busy to perform the requested operation. In this circumstance the diagnostic tool shall perform repetition of the "identical request message" or "another request message". (Repetition maximum: 3 times)。 If the ECU is able to perform the diagnostic task but needs additional time to finish the task and prepare the response, the Negative Response Code $78 shall be used instead of Negative Response Code $21. This response code is in general supported by each diagnostic service, therefore it is not listed in the list of applicable response codes of the diagnostic services.
$22	**Conditions Not Correct** This response code indicates that the requested action will not be taken because the ECU prerequisite conditions are not met.
$23	Reserved
$24	**Request Sequence Error** This response code indicates that the requested action will not be taken because the ECU expects a different sequence of request messages or message as send by the diagnostic tool.
$25 - $30	Reserved
$31	**Request Out Of Range** This response code indicates that the requested action will not be taken because the ECU has detected that the request message contains a parameter that attempts to substitute a value beyond its range of authority.
$32	Reserved
$33	**Security Access Denied** This response code indicates that the requested action will not be taken because the diagnostic tool has not satisfied the ECU's security strategy.
$34	Reserved
$35	**Invalid Key** This response code indicates that the ECU has not given security access because the key sent by the diagnostic tool did not match with the key in the ECU's memory.
$36	**Exceed Number Of Attempts** This response code indicates that the requested action will not be taken because the diagnostic tool has unsuccessfully attempted to gain security access more times than the ECU's security strategy will allow.
$37	**Required Time Delay Not Expired** This response code indicates that the requested action will not be taken because the diagnostic tool's latest attempt to gain security access was initiated before the ECU's required timeout period had elapsed.
$38 – $6F	Reserved
$70	**Upload Download Not Accepted** This response code indicates that an attempt to upload/download to a ECU's memory cannot be accomplished due to some fault conditions.
$71	**Transfer Data Suspended** This response code indicates that a data transfer operation was halted due to some fault.
$72	**General Programming Failure** This response code indicates that the ECU detected an error when erasing or programming a memory location in the permanent memory device (e.g. Flash Memory).
$73	**Wrong Block Sequence Counter** This response code indicates that the ECU detected an error in the sequence of Block Sequence Counter values.
$74 – $77	Reserved

Table9.2 Definition of Negative Response Code Values

Hex value	Description
$78	**Request Correctly Received-Response Pending** This response code indicates that the request message was received correctly, and that all parameters in the request message were valid, but the action to be performed is not yet completed and the ECU is not yet ready to receive another request. As soon as the requested service has been completed, the ECU shall send a positive response message or negative response message with a response code different from this. The negative response message with this response code may be repeated by the ECU until the requested service is completed and the final response message is sent. After reception of this code, the diagnostic tool shall wait without sending a request message untill receiving a positive response message or negative response message with a response code different from this. This response code might impact the application layer timing parameter values. This response code shall only be used in a negative response message if the ECU will not be able to receive further request messages from the diagnostic tool while completing the requested diagnostic service. This response code is in general supported by each diagnostic service, therefore it is not listed in the list of applicable response codes of the diagnostic services.
$79 – $7D	Reserved
$7E	**Sub Function Not Supported In Active Session** This response code indicates that the requested action will not be taken because the ECU does not support the requested sub-function in the session currently active. This response code is in general supported by each diagnostic service, therefore it is not listed in the list of applicable response codes of the diagnostic services.
$7F	**Service Not Supported In Active Session** This response code indicates that the requested action will not be taken because the ECU does not support the requested service in the session currently active. This response code is in general supported by each diagnostic service, therefore it is not listed in the list of applicable response codes of the diagnostic services.
$81	**RPM Too High** This response code indicates that the requested action will not be taken because the ECU prerequisite condition for RPM is not met (current RPM is above a pre-programmed maximum threshold).
$82	**RPM Too Low** This response code indicates that the requested action will not be taken because the ECU prerequisite condition for RPM is not met (current RPM is below a pre-programmed minimum threshold).
$83	**Engine Is Running** This response code is required for those actuator tests, which cannot be actuated while the Engine is running.
$84	**Engine Is Not Running** This response code is required for those actuator tests, which cannot be actuated unless the Engine is running. This is different from RPM too low negative response, and needs to be allowed.
$85	**Engine Run Time Too Low** This response code indicates that the requested action will not be taken because the ECU prerequisite condition for engine run time is not met (current engine run time is below a pre-programmed limit).
$86	**Temperature Too High** This response code indicates that the requested action will not be taken because the ECU prerequisite condition for temperature is not met (current temperature is above a pre-programmed maximum threshold).
$87	**Temperature Too Low** This response code indicates that the requested action will not be taken because the ECU prerequisite condition for temperature is not met (current temperature is below a pre-programmed minimum threshold).
$88	**Vehicle Speed Too High** This response code indicates that the requested action will not be taken because the ECU prerequisite condition for vehicle speed is not met (current VS is above a pre-programmed maximum threshold).
$89	**Vehicle Speed Too Low** This response code indicates that the requested action will not be taken because the ECU prerequisite condition for vehicle speed is not met (current VS is below a pre-programmed minimum threshold).
$8A	**Throttle/Pedal Too High** This response code indicates that the requested action will not be taken because the ECU prerequisite condition for throttle/pedal position is not met (current TP/APP is above a pre-programmed maximum threshold).
$8B	**Throttle/Pedal Too Low** This response code indicates that the requested action will not be taken because the ECU prerequisite condition for throttle/pedal position is not met (current TP/APP is below a pre-programmed minimum threshold).
$8C	**Transmission Range In Neutral** This response code indicates that the requested action will not be taken because the ECU prerequisite condition for being in neutral is not met.
$8D	**Transmission Range In Gear** This response code indicates that the requested action will not be taken because the ECU prerequisite condition for being in gear is not met.
$8F	**Brake Switch (es) Not Closed (Brake Pedal not pressed or not applied)** For safety reasons, this is required for certain tests before it begins, and must be maintained for the entire duration of the test.

Table9.2 Definition of Negative Response Code Values

Hex value	Description
$90	**Shifter Lever Not In Park** For safety reasons, this is required for certain tests before it begins, and must be maintained for the entire duration of the test.
$91	**Torque Converter Clutch Locked** This response code indicates that the requested action will not be taken because the ECU prerequisite condition for torque converter clutch is not met.
$92	**Voltage Too High** This response code indicates that the requested action will not be taken because the ECU prerequisite condition for voltage at the primary pin of the ECU (ECU) is not met (current voltage is above a preprogrammed maximum threshold).
$93	**Voltage Too Low** This response code indicates that the requested action will not be taken because the ECU prerequisite condition for voltage at the primary pin of the ECU (ECU) is not met (current voltage is below a preprogrammed maximum threshold).
$94 – $FF	Reserved

Note)

If an ECU will send a Negative Response message with a Response Code $11, $12, or $31, the ECU does not send a Negative Response message with a Negative Response Code $78 before. In other words, an ECU shall send a Negative Response message with a Response Code $11, $12, or $31 within P2Can_ECU.