SDS Protocol

Simple Diagnostic Service (SDS) is a communication protocol for the certain vehicle ECUs used to perform diagnostic operations to make sure ECUs run properly.

1 Services

The following services are implemented in SDS:

- Initiate Diagnostic Session ($20)
- Return To Normal ($21)
- Security Access ($22)
- Read Memory By Address ($23)
- Read DID By ID ($24)
- Programming Mode ($25)
- Request Download ($26)
- Transfer Data ($27)

1.1 Initiate Diagnostic Session ($20)

An ECU can be in one of three modes:

- Default
- Diagnostic
- Device Control

Default Mode is when the ECU is running normally. During this mode, all services are disabled except for InitiateDiagnosticSession($20) to allow the switching of modes.

Diagnostic Mode is used to perform the READ-ONLY actions in order to passively diagnose problems that may occur in the ECU.

Device Control Mode is used for ECU programming. This mode allows ECU firmware to be updated or rewritten in case of security updates or calibration modifications.

1.1.1 Request Message Definition

Data Byte	Parameter Name	Hex Value
#1	IniitiateDiagnosticSession request ServiceID	20

Data Byte	Parameter Name	Hex Value
#2	sub-function = [DEFAULT DIAGNOSTIC DEVICE_CONTROL]	 01 02 03

1.1.2 Positive Response Message Definition

Data Byte	Parameter Name	Hex Value
#1	IniitiateDiagnosticSession positive response ServiceID	60
#2	current_mode = [DEFAULT DIAGNOSTIC DEVICE_CONTROL]	 00 01 02

1.1.3 Supported Negative Response Codes

Description	Hex Value
ConditionsNotCorrect: Security Access has not been granted	13
RequestOutOfRange: The request mode value is not 1-3	14

1.2 Return To Normal ($21)

Return To Normal Service will reset all diagnostic settings and set the session to DEFAULT.

1.2.1 Request Message Definition

Data Byte	Parameter Name	Hex Value
#1	ReturnToNormal request ServiceID	21

1.2.2 Positive Response Message Definition

Data Byte	Parameter Name	Hex Value
#1	ReturnToNormal positive response ServiceID	61

1.3 Security Access ($22)

Security Access Service is used to request security challenge token and solve security challenge. Unlocking security access allows for Device_Control mode to be enabled.

1.3.1 Security Key Algorithm

The security key shall be derived from the security seed using an internal cryptographic algorithm. The 5-Byte seed/key algorithm is only known to the Original Vehicle Manufacturer, ensuring that only properly authorized parties can unlock an ECU.

1.3.2 Steps for unlocking an ECU

- Tester requests the Seed
- ECU sends the Seed
- Tester sends the correct Key
- ECU response that the Key is valid
- ECU unlocks itself

1.3.3 Request Message Definition

Data Byte	Parameter Name	Hex Value
#1	SecurityAccess request ServiceID	22
#2	sub-function = [REQUEST_SEED VALIDATE_KEY]	 01 02
 #3 #4 #5 #6 #7	SecurityKey = [Byte 1 Byte 2 Byte 3 Byte 4 Byte 5]	 XX XX XX XX XX

Security Key is only required for subfunction VALIDATE_KEY.

1.3.4 Positive Response Message Definition

Positive Response for Sub-Function REQUEST_SEED

Data Byte	Parameter Name	Hex Value
#1	SecurityAccess positive response ServiceID	62
#2	sub-function REQUEST_SEED	01

Data Byte	Parameter Name	Hex Value
	SecuritySeed = [	
#3	Byte 1	XX
#4	Byte 2	XX
#5	Byte 3	XX
#6	Byte 4	XX
#7	Byte 5	XX
	]	

Positive Response for Sub-Function VALIDATE_KEY

Data Byte	Parameter Name	Hex Value
#1	SecurityAccess positive response ServiceID	62
#2	sub-function VALIDATE_KEY	02

1.3.5 Supported Negative Response Codes

Description	Hex Value
SubFunctionNotSupported-InvalidFormat: Invalid SubFunction or message length incorrect	12
ConditionsNotCorrect: Current Session is not Diagnostic Mode	13
InvalidKey: Key supplied does not match calculated key	15
ExceedNumAttempts: Incorrectly solved the security challenge too many times	16

1.4 Read Memory By Address ($23)

Read Memory By Address Service allows for tester to request ECU memory. Requested memory address may not fall within ECU protected region.

1.4.1 Request Message Definition

Data Byte	Parameter Name	Hex Value
#1	ReadMemoryByAddress request ServiceID	23
	MemoryAddress = [	
#2	Byte 1	XX
#3	Byte 2	XX
#4	Byte 3	XX
#5		XX

Data Byte	Parameter Name	Hex Value
	Byte 4	
	]	
	ReadSize = [	
#6	Byte 1	XX
#7	Byte 2	XX
	]	

1.4.2 Positive Response Message Definition

If requested memory is less than 7 bytes

Data Byte	Parameter Name	Hex Value
#1	ReadMemoryByAddress positive response ServiceID	63
	MemoryBytes = [	
#2	Byte 1	XX
#3	Byte 2	XX
#4	Byte 3	XX
#5	Byte 4	XX
#6	Byte 5	XX
#7	Byte 6	XX
	]	

If requested memory is greater than 6 Bytes

A Flow Control Frame is sent with the rest of the data returned.

1.4.3 Supported Negative Response Codes

Description	Hex Value
SubFunctionNotSupported-InvalidFormat: Invalid SubFunction or message length incorrect	12
ConditionsNotCorrect: Current Session is not Diagnostic Mode	13
RequestOutOfRange: The memory address requested is not within readable range	14

1.5 Read DID By ID ($24)

Read DID By ID Service will return a Data IDentifier associated to the ID requested.

1.5.1 DID Table

DID ID	DID Value
0	Challenge Author
1	Vehicle Manufactur
2	Vehicle Year
3	Vehicle Identifier Number

1.5.2 Request Message Definition

Data Byte	Parameter Name	Hex Value
#1	ReadDIDByID request ServiceID	24
#2	DID_ID	XX

1.5.3 Positive Response Message Definition

If DID response is less than 7 bytes

Data Byte	Parameter Name	Hex Value
#1	ReadDIDByID positive response ServiceID	64
	MemoryBytes = [	
#2	Byte 1	XX
#3	Byte 2	XX
#4	Byte 3	XX
#5	Byte 4	XX
#6	Byte 5	XX
#7	Byte 6	XX
	]	

If DID response is greater than 6 Bytes

A Flow Control Frame is sent with the rest of the data returned.

1.5.4 Supported Negative Response Codes

Description	Hex Value
ConditionsNotCorrect: Current Session is not Diagnostic Mode	13
RequestOutOfRange: The DID ID requested is not valid	14

1.6 Programming Mode ($25)

The Programming Mode Service is used to initiate a programmings session. This puts the ECU in a state to be ready for future programming. DeviceControl Session is required.

1.6.1 Request Message Definition

Data Byte	Parameter Name	Hex Value
#1	ProgrammingMode request ServiceID	25

1.6.2 Positive Response Message Definition

Data Byte	Parameter Name	Hex Value
#1	ProgrammingMode positive response ServiceID	65
#2	ProgrammingMode Sucessful	01

1.6.3 Supported Negative Response Codes

Description	Hex Value
ConditionsNotCorrect: Current Session is not DeviceControl	13

1.7 Request Download ($26)

The RequestDownload Service allows the tester to initiate a Programming Sequence by supplying the size of a future TransferData($27) Service request. Programming Mode is required. The maximum size of a transfer value is 0xffff bytes.

1.7.1 Request Message Definition

Data Byte	Parameter Name	Hex Value
#1	RequestDownload request ServiceID	26
	DownloadSize = [	
#2	Byte 1	XX
#3	Byte 2	XX
	]	

1.7.2 Positive Response Message Definition

Data Byte	Parameter Name	Hex Value
#1	RequestDownload positive response ServiceID	66
	DownloadSize = [	
#2	Byte 1	XX

Data Byte	Parameter Name	Hex Value
#3	Byte 2	XX
	]	

1.7.3 Supported Negative Response Codes

Description	Hex Value
SubFunctionNotSupported-InvalidFormat: Invalid SubFunction or message length incorrect	12
ConditionsNotCorrect: Current Session is not DeviceControl Mode	13

1.8 Transfer Data ($27)

The TransferData Service is used to send bytes to an ECU at a specified address. The subfunction DownloadAndExecute ($80) can be used to execute the bytes sent. DeviceControl Session is required.

1.8.1 Request Message Definition

Data Byte	Parameter Name	Hex Value
#1	TransferData request ServiceID	27
#2	sub-function = [	
	Download	00
	DownloadAndExecute	80
	]	
	DownloadAddress* = [	
#3	Byte 1	XX
#4	Byte 2	XX
#5	Byte 3	XX
#6	Byte 4	XX
	]	
#7	TransferDataByte*	XX

* DownloadAddress and TransferDataByte are not required if subfunction DownloadAndExecute is requested and data has been transferred. DownloadAndExecute will execute the bytes at the last TransferData Service request DownloadAddress

1.8.2 Positive Response Message Definition

Data Byte	Parameter Name	Hex Value
#1	RequestDownload positive response ServiceID	66

1.8.3 Supported Negative Response Codes

Description	Hex Value
SubFunctionNotSupported-InvalidFormat: Invalid SubFunction or message length incorrect	12
ConditionsNotCorrect: Current Session is not DeviceControl Mode	13
RequestOutOfRange: The Download Address is out of range	14

1.9 Flow Control ($30)

A Flow control frame is used when the response of an SDS message is larger than what one frame can hold (7 bytes).

1.9.1 Request Message Definition

A Flow control message uses the 0x10 as the first byte

Data Byte	Parameter Name	Hex Value
#0	Flow Control Indicator	10
#1	Response Length	XX
	MemoryBytes = [	
#2	Byte 1	XX
#3	Byte 2	XX
#4	Byte 3	XX
#5	Byte 4	XX
#6	Byte 5	XX
#7	Byte 6	XX
	]	

To recieve the rest of the message, an 0x30 message should be sent.

Data Byte	Parameter Name	Hex Value
#0	Flow Control Indicator	30

The response of the rest of the message will be in the following format:

Data Byte	Parameter Name	Hex Value
#0	Frame ID	2X
 #1 #2 #3 #4 #5 #6 #7	MemoryBytes = [Byte 1 Byte 2 Byte 3 Byte 4 Byte 5 Byte 6 Byte 7]	 XX XX XX XX XX XX XX

The Frame ID will set with 0x21 and increment wrapping back to 0x21 after reaching 0x2F.

2 Negative Response Codes ($7F)

The negative response service shall be used by an ECU to indicate a diagnostic service message request has failed.

Negative Response Message Format

Data Byte	Parameter Name	Hex Value
#1	Negative Response Service Idenfifier	7F
#2	requestServiceId	XX
#3	returnCode	XX

2.1 Service Not Supported ($11)

2.2 SubFunction Not Supported-Invalid Format ($12)

2.3 Conditions Not Correct ($13)

2.4 Request Out Of Range ($14)

2.5 Invalid Key ($15)

2.6 Exceed Number of Attempts ($16)

Engine Control Module

The ECU ID for the ECM is `7E0`

1 Memory Mapping

The ECM contains 3 memory sections:

- ROM
- PROTECTED
- RAM

Section	Memory Start	Memory End
ROM	0x60010000	0x61000000
PROTECTED	0x61000000	0x62000000
RAM	0x70000000	0x71000000

1.1 ROM

The Read Only Memory consists of the read only code section which hold the functionality for handling SDS messages.

1.2 PROTECTED

The Protected memory region holds ECM secrets that should not be accessed through any SDS messages.

1.3 RAM

The Random Access Memory region is where hold the .data and .bss section. This is also the only region of memory the TransferData($27) service can write to.

Body Control Module

The ECU ID for the BCM is `7C0`. The BCM only supports the ReadDIDByID($24) message.

SDS Software

The SDS Software is a tool to interact with the ECUs.

1 Commands

The following commands are available in the SDS Software:

- help
- cansend

- candump
- start_engine
- reboot
- exit

1.1 help

The help command prints the command list.

1.2 cansend

The cansend utility allows for the tester to send CAN messages to ECUs. The utility is fixed to the `can0` interface.

Usage:

`cansend [ECU ID]#[CAN MESSAGE]`

Example:

`cansend 7e0#022001`
This will send an InitiateDiagnosticSession($20) message to the ECM

1.3 candump

The cansend utility will retrieve all the can messages sent on the CAN network. To clear the buffer, there is a subfunction `clear`

Usage:

```
candump
candump clear
```

1.4 start_engine

This utility is used to test if the engine can start.

Usage:

`start_engine`

1.5 reboot

This utility is used to reboot all the ECUs if case they are in a broken state.

Usage:

```
reboot
```

1.6 exit

This utility is used to exit the SDS Software

Usage:

```
exit
```