Available online www.ejaet.com

European Journal of Advances in Engineering and Technology, 2021, 8(6):70-73

Research Article ISSN: 2394-658X

Unified Diagnostic Services (UDS) in Automotive: A technical Study

Vinay Nagarad Dasavandi Krishnamurthy

Software Engineer
Bosch USA
Michigan- USA
Email id - vinayndk9@gmail.com

ABSTRACT

The Engine Control Unit (ECU) stands as a cornerstone in the automotive sector, yet its design and diagnostic requirements vary widely among manufacturers. This lack of standardization poses challenges in calibration and maintenance, as only developers possess the expertise needed to navigate the intricacies of their respective ECUs. While each manufacturer implements its own diagnostic system within the ECU, adherence to common diagnostic standards is essential to ensure uniform behavior and interfaces across all ECUs. Consequently, a comprehensive diagnostic system must incorporate protocols that facilitate seamless communication between diagnostic tools used by developers, testers, and repairers to access the ECU's diagnostic information. However, implementing these protocols for individual diagnostic systems and vehicle components demands considerable effort. ISO and SAE have defined various diagnostic systems tailored to specific requirements and vehicle diagnostics. Recognizing the perpetual challenge of integrating diagnostic systems for specific needs, developers often face prolonged development timelines. To mitigate this, developers can adopt a Unified Diagnostic Services (UDS) protocol, established by either ISO or SAE standards, to support a multitude of diagnostic standards. This study covers the background, general working of UDS, architecture and UDS message format.

Keywords: UDS, ECU, OEM, OSI, CAN, DTC

INTRODUCTION

The Engine Control Unit (ECU) plays a pivotal role in the automotive industry, serving as the brain of the vehicle's electronic systems. Its primary function is to manage and regulate various aspects of the engine's operation, ensuring optimal performance, fuel efficiency, and emissions control. Specifically, the ECU monitors inputs from sensors placed throughout the vehicle, such as the throttle position sensor, oxygen sensor, and engine temperature sensor. Based on these inputs, the ECU makes real-time calculations and adjustments to control critical engine functions, including fuel injection timing, air-fuel mixture ratio, ignition timing, and idle speed. Furthermore, the ECU communicates with other vehicle systems, such as the transmission control module (TCM) and antilock braking system (ABS), to coordinate their operation and optimize overall vehicle performance and safety.

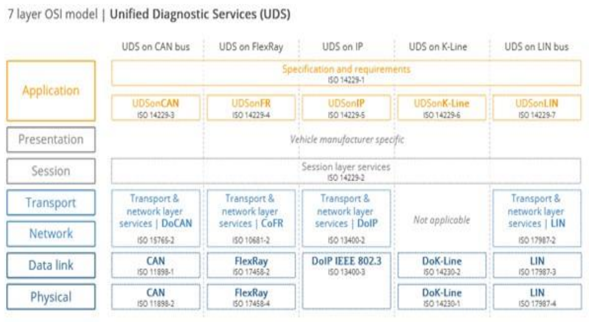

7 layer OSI model | Unified Diagnostic Services (UDS)
UDS on CAN bus
UDS on FlexRay
UDS on IP
UDS on K-Line
UDS on LIN bus
Application
Presentation
Session
Transport
Network
Data link
Physical
Specification and requirements ISO 14229-1
UDSonCAN ISO 14229-3
UDSonFR ISO 14229-4
UDSonIP ISO 14229-5
UDSonK-Line ISO 14229-6
UDSonLIN ISO 14229-7
Vehicle manufacturer specific
Session layer services ISO 14229-2
Transport & network layer services | DoCAN ISO 15765-2
Transport & network layer services | CoFR ISO 10681-2
Transport & network layer services | DoIP ISO 13400-2
Not applicable
Transport & network layer services | LIN ISO 17987-2
CAN ISO 11898-1
FlexRay ISO 17458-2
DoIP IEEE 802.3 ISO 13400-3
DoK-Line ISO 14230-2
LIN ISO 17987-3
CAN ISO 11898-2
FlexRay ISO 17458-4
DoK-Line ISO 14230-1
LIN ISO 17987-4

Figure 1: UDS architecture in 7 layer OSI

In essence, the ECU acts as the central nervous system of the vehicle, continuously monitoring, analyzing, and adjusting various parameters to ensure smooth and efficient operation under diverse driving conditions Unified Diagnostic Services (UDS) is a standardized diagnostic protocol used in the automotive industry for communication between electronic control units (ECUs) in vehicles and diagnostic tools. 'Unified' in this context means that it is an international and not a company-specific standard. The UDS protocol (ISO 14229) is standardized across both manufacturers and standards (such as CAN, KWP 2000, Ethernet, LIN). Further, UDS is today used in ECUs across all tier 1 Original Equipment Manufacturers (OEMs).

LITERATURE REVIEW

A. UDS ARCHITECTURE

The Architecture of the UDS protocol is designed based on the Open System Interconnection (OSI) Reference Model. Hence, the UDS software stack has a layered architecture. Diagnostic tools can contact all ECUs installed in a vehicle which has UDS services enabled. In contrast to the CAN bus protocol, which only uses the first and second layers of the OSI model, UDS utilizes the fifth and seventh layers of the OSI model.

WORKING OF UDS

In practice, UDS communication is performed in a clientserver relationship - with the client being a tester-tool and the server being a vehicle ECU. For example, you can connect a CAN bus interface to the OBD2 connector of a car and send UDS requests into the vehicle. Assuming the targeted ECU supports UDS services, it will respond accordingly. Modern vehicles have a diagnostic interface for off-board diagnostics, which makes it possible to connect a computer (client) or diagnostics tool, which is referred to as tester, to the communication system of the vehicle. Thus, UDS requests can be sent to the controllers which must provide a response (this may be positive or negative). This makes it possible to interrogate the fault memory of the individual control units, to update them with new firmware, have low-level interaction with their hardware (e.g. to turn a specific output on or off), or to make use of special functions (referred to as routines) to attempt to understand the environment and operating conditions of an ECU to be able to diagnose faulty or otherwise undesirable behavior. The UDS protocol provides a number of necessary functionalities for repairers, developers and testers so that they can, for example, read or write data in ECU memory, program the flash memory and create specific behavior for an ECU such as provide a response. An application software inside ECU can be viewed as two system as shown in figure 1 such as diagnostic and control system. Control system is used for controlling the engine if we take internal combustion engine. It possesses the ability to generate Diagnostic Trouble Codes (DTCs) upon detecting a malfunction. These DTCs convey the status information of an engine, which is crucial for repairers or technicians in diagnosing and maintaining the engine. However, it is imperative that repairers and technicians do not have access to the control system itself. Hence, a diagnostic system is developed to restrict access rights. This diagnostic system should furnish comprehensive information about a DTC and certain aspects of the control system for calibration purposes. As mentioned, the diagnostic system within an ECU varies among manufacturers, and each diagnostic system employs a unique set of DTCs. Therefore, the Unified Diagnostic Services (UDS) protocol is designed to accommodate various types of DTCs specified by ISO standards, SAE standards, and vehicle manufacturers (Assawinjaipetch, Panuwat, et a).

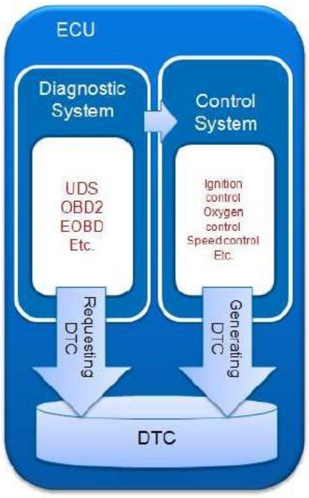
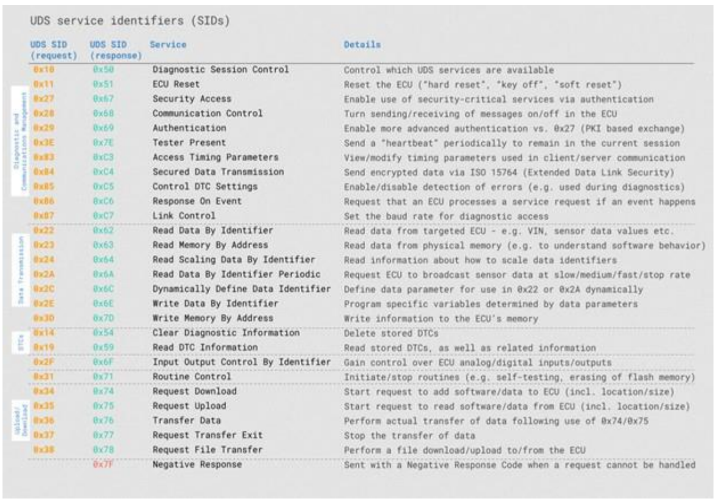


Figure 2: ECU Application software DTC view

UDS MESSAGE FORMAT

UDS is a Request and Response-based protocol based on client-server architecture, and it has having unique service ID(SID). SID is the size of one byte, and it ranges from 0x00 to 0x3E.

A. Service ID It is a 1-byte identifier, it indicates the services which are defined in ISO-14229. The server (ECU) sees this identifier and does the operation based on this identifier. Examples are shown in Figure 3



UDS service identifiers (SIDs)

	UDS SID (request)	UDS SID (response)	Service	Details
Diagnostic and Communications Management	0x10	0x50	Diagnostic Session Control	Control which UDS services are available
	0x11	0x51	ECU Reset	Reset the ECU ("hard reset", "key off", "soft reset")
	0x27	0x67	Security Access	Enable use of security-critical services via authentication
	0x28	0x68	Communication Control	Turn sending/receiving of messages on/off in the ECU
	0x29	0x69	Authentication	Enable more advanced authentication vs. 0x27 (PKI based exchange)
	0x3E	0x7E	Tester Present	Send a "heartbeat" periodically to remain in the current session
	0x83	0xC3	Access Timing Parameters	View/modify timing parameters used in client/server communication
	0x84	0xC4	Secured Data Transmission	Send encrypted data via ISO 15764 (Extended Data Link Security)
	0x85	0xC5	Control DTC Settings	Enable/disable detection of errors (e.g. used during diagnostics)
	0x86	0xC6	Response On Event	Request that an ECU processes a service request if an event happens
	0x87	0xC7	Link Control	Set the baud rate for diagnostic access
Data Transmission	0x22	0x62	Read Data By Identifier	Read data from targeted ECU - e.g. VIN, sensor data values etc.
	0x23	0x63	Read Memory By Address	Read data from physical memory (e.g. to understand software behavior)
	0x24	0x64	Read Scaling Data By Identifier	Read information about how to scale data identifiers
	0x2A	0x6A	Read Data By Identifier Periodic	Request ECU to broadcast sensor data at slow/medium/fast/stop rate
	0x2C	0x6C	Dynamically Define Data Identifier	Define data parameter for use in 0x22 or 0x2A dynamically
	0x2E	0x6E	Write Data By Identifier	Program specific variables determined by data parameters
	0x3D	0x7D	Write Memory By Address	Write information to the ECU's memory
DTCs	0x14	0x54	Clear Diagnostic Information	Delete stored DTCs
	0x19	0x59	Read DTC Information	Read stored DTCs, as well as related information
	0x2F	0x6F	Input Output Control By Identifier	Gain control over ECU analog/digital inputs/outputs
	0x31	0x71	Routine Control	Initiate/stop routines (e.g. self-testing, erasing of flash memory)
Upload/ Download	0x34	0x74	Request Download	Start request to add software/data to ECU (incl. location/size)
	0x35	0x75	Request Upload	Start request to read software/data from ECU (incl. location/size)
	0x36	0x76	Transfer Data	Perform actual transfer of data following use of 0x74/0x75
	0x37	0x77	Request Transfer Exit	Stop the transfer of data
	0x38	0x78	Request File Transfer	Perform a file download/upload to/from the ECU
		0x7F	Negative Response	Sent with a Negative Response Code when a request cannot be handled

Figure 3: UDS Service Identifiers

B. Sub-function

This is an optional field. Example: Under ECU Reset service ID (0x11), there are 3 sub-function IDs

[1]. Hard Reset(0x01)

[2]. Key-Off On Reset(0x02)

[3]. Soft Reset(0x03)

FRAME FORMAT TYPES

Basically, there are 4 types of frame formats,

[1]. Request frame with sub-function ID

[2]. Request frame without sub-function ID

[3]. Positive Response Frame

[4]. Negative Response Frame

A. REQUEST WITH SUB-FUNCTION ID

The request frame is used to send the request to the server (ECU) from the client (Tester tool).

Table 1: Request with Sub-function ID

Service ID (SID)	Sub-Function ID	Data Parameters

B. REQUEST WITHOUT SUB-FUNCTION ID

Table 2: Request without sub-function ID

Service ID (SID)	Data Parameters

C. POSITIVE RESPONSE FRAME

When the server (ECU) receives the service request from the tester, the server checks the message. If everything is fine, then it executes the requested service and responds to the client with a positive response. If the response is positive, then the 6th bit of the SID should be 1. For example, Service ID – 0x31 = 0 0 1 1 0 0 0 1 For a positive response, 0 1 1 1 0 0 0 1 is equal to 0x71 (0x31 + 0x40). In another way, we can say the positive response means SID+ 0x40, there is no logical reason for this. Simply, it is defined in International standard IS0-14229-1.

Table 3: Positive Response with Sub-function ID

Service ID (SID) + 0x40	Sub-Function ID	Data Parameters

Table 4: Positive Response without Sub-function ID

Service ID (SID) + 0x40	Data Parameters

D. NEGATIVE RESPONSE FRAME

When the server (ECU) receives the service request from the tester, ECU checks the message. If the server finds something wrong, then it executes the negative response and sends the Negative response code (NRC). Below are some Negative Response Codes given.

Table 5: Negative response examples with description

NRC	**Description**
0x10	General reject
0x11	Service not supported
0x12	Subfunction not supported
0x13	Incorrect message length or invalid format
0x14	Response too long

CONCLUSION

In conclusion the Unified Diagnostic Services (UDS) protocol serves as a standardized diagnostic protocol, facilitating seamless communication between ECUs and diagnostic tools. The UDS protocol, based on the Open System Interconnection (OSI) Reference Model, offers a layered architecture and supports various diagnostic functionalities necessary for repairers, developers, and testers. With unique service IDs and sub-functions, the UDS message format enables effective request and response-based communication between clients and servers.
Overall, the UDS protocol plays a crucial role in streamlining diagnostics and maintenance processes in the automotive industry, promoting interoperability and efficiency across different vehicle platforms and diagnostic tools. As automotive technology continues to evolve, the adoption of standardized diagnostic protocols like UDS will be instrumental in ensuring the reliability and performance of modern vehicles.

REFERENCES

[1]. Assawinjaipetch, Panuwat, et al. "Unified Diagnostic Services Protocol Implementation in an Engine Control Unit." (2013).
[2]. XIE, Yue-yin, Chao ZHOU, and Feng LUO. "Implementation of Automotive Unified Diagnostic Services Based on AUTOSAR."
[3]. Khan, Jihas. Generic Model Based Architecture for Implementing Client Side Algorithms Used in Unified Diagnostic Service and On Board Diagnostics for Different Hardware Targets. No. 2016-01-0072. SAE Technical Paper, 2016.
[4]. Rings, Matt, and Paul Phillips. Adding unified diagnostic services over CAN to an HIL test system. No. 2011-01-0454. SAE Technical Paper, 2011
[5]. https://embetronicx.com/tutorials/automotive/uds-protocol/uds-protocol-introduction-unified-diagnostic-services-uds-protocol-tutorial-part-1/
[6]. https://www.csselectronics.com/pages/uds-protocol-tutorial-unified-diagnostic-services
[7]. https://en.wikipedia.org/wiki/Unified_Diagnostic_Services