International Journal of Science and Research (IJSR)
ISSN: 2319-7064
Impact Factor 2024: 7.101

Streamlining End-of-Line ECU Testing with UDS over CAN

Rutika Pawar

Advanced Engineering Department TATA Motors, Pimpri, India
Email: *rp924625.ttl[at]tatamotors.com, rutikapawar2615[at]gmail.com*

Abstract: ***Modern vehicles have electronic control devices called Electronic Control Units (ECUs). These ECUs use a communication system called the Controller Area Network (CAN) to share information. CAN is reliable, strong, and helps prevent errors. However, as the number of ECUs in a vehicle increases, identifying issues in a specific ECU becomes more challenging. To diagnose these electronic devices and circuits, a standard communication method called Unified Diagnostic Services (UDS) is used. This enables to diagnose the ECU and troubleshoot it without taking it out from the vehicle assembly. This paper explains how UDS is used to test inputs and outputs in a vehicle.***

Keywords: ECU, CAN, ISO 14229, UDS, I/O Testing, DID

1. Introduction

In modern automotive electronic networks, Testing ECUs grows critical as their numbers rise. A standardized solution for this is provided by ISO 14229. The CAN-based UDS is the preferred method for ECU testing because it only requires an ISO 14229-compatible tester tool. The only other requirement is that the ECUs in the network must have a UDS protocol stack.

The communication system used by ECUs, called Controller Area Network (CAN), is defined in ISO 11898. The diagnostic protocol, Unified Diagnostic Services (UDS), is defined in ISO 14229 [1]. This protocol allows a tester tool (acting as a client) to communicate with an ECU (acting as a server).

This paper focuses on UDS over CAN, as defined in ISO 14229, and explains how it is used for testing ECU inputs and outputs. A brief overview of these protocols is provided below.

2. Overview

a) Electronic Control Unit (ECU)

In the automotive field, an Electronic Control Unit (ECU) is a small computer inside a vehicle that controls one or more systems. The ECU collects signals from sensors to check the status of different parts of the vehicle. Based on this data, a control algorithm decides what actions to take and sends commands to other control devices to ensure the vehicle runs efficiently.

ECUs can also store sensor data to improve performance through adaptive control systems. Depending on their function, ECUs have different names, such as Body Control Module, Transmission Control Unit, and Engine Management System.

These ECUs use the CAN communication protocol to exchange information within the vehicle.

b) Controller Area Network (CAN)

In the early 1980s, Electronic Control Units (ECUs) began to be used in vehicles. With the introduction of ECUs, real-time communication within the vehicle became necessary. Initially, adding more wires to the cable harness was used for controller communication, but this increased cost, complexity, and reduced reliability. To solve this issue, Robert Bosch GmbH developed the CAN network, which requires only two wires—CAN High and CAN Low—for communication throughout the vehicle [2].

Many semiconductor companies manufacture CAN controller chips, which are used in various automotive applications, such as test and measurement systems for different vehicle parts and ECU development.

The CAN communication protocol has two types of frame formats, as defined in the Bosch Version 2.0 specifications:

Standard 2.0A and Extended 2.0B. The main difference between them is the length of the arbitration ID. The Standard 2.0A frame has an 11-bit ID, while the Extended 2.0B frame has a 29-bit ID. The complete frame structures are shown in Fig. 1[3].

Standard Frame Format

SOF	11- Bit Arbitration ID	RTR	IDE	DLC	0-8 Data Bytes	16- Bit CRC	ACK	End of Frame

Extended Frame Format

SOF	High 11- Bit of Arbitration ID	RTR	IDE	DLC	0-8 Data Bytes	16- Bit CRC	ACK	End of Frame

Figure 1: 2.0A standard and 2.0B extended CAN frame

The CAN protocol ensures accurate data transfer and includes features for detecting errors [4]. It is resistant to noise because it uses differential signaling between the CAN High and CAN Low lines. CAN communication is also reliable since each frame is assigned a priority, which helps prevent data collisions. This priority is determined by an arbitration ID included in every packet [5].

OSI Model for UDS Application

The UDS application is built according to the ISO OSI model. UDS is a separate application in the ECU, so it runs above the typical lower layers in the ECU as shown in Fig. 2.

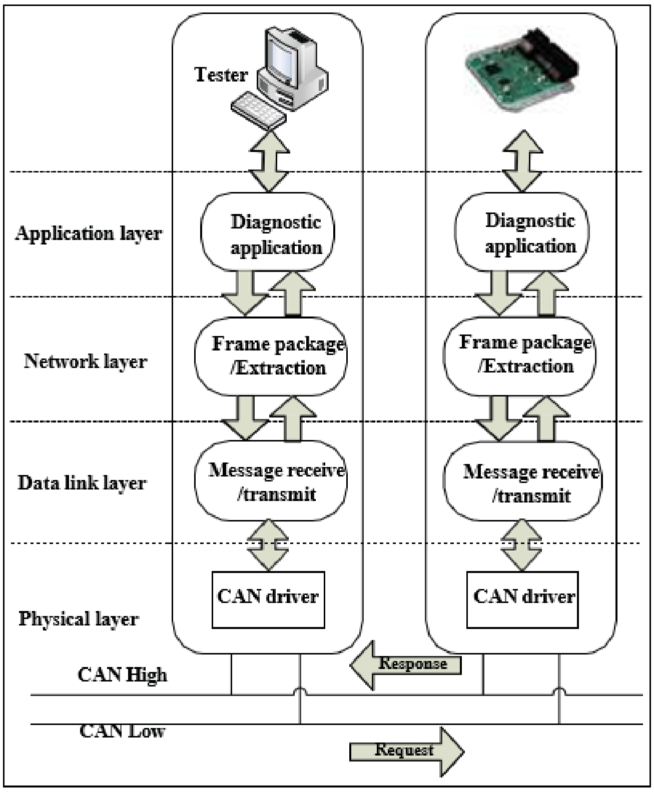

Figure 2: The OSI model for UDS application

Diagnostic services fit in the application layer of the OSI model. Services defined in the UDS are application routines which receives and transmits their messages over a CAN network. CAN driver detects UDS requests from tester by an arbitration ID set by OEM and supplies the data packets to the UDS application which operates on the requested messages and sends appropriate response over the same CAN network on which tester is connected.

Unified Diagnostic Services

UDS is a unified diagnostic services protocol standardized by the International standards Organization (ISO) in the ISO 14229.UDS has no restrictions to the communication medium as it can be implemented on any Medium including CAN. UDS application software can be used for various applications such as diagnostics, calibration and reprogramming. If implemented over a CAN network a single off-board tester can be configured to test multiple ECUs at once.

UDS specifies different diagnostic services to achieve diagnostic functions. UDS diagnostic services are divided into six functional units, including diagnostic management unit, data transmission unit, data storage unit, input/output control unit, routine control unit, and upload/download unit. Diagnostic and communication management unit defines the main session control service and the communication timing parameter settings, which allows a service to run into supported sessions. Data transmission service monitors the parameters of vehicle by a unique Data Identifier to each parameter. Write memory by address service writes data into ECU internal non-volatile memory which is used as faulty memory or calibration memory. Input/output control functional unit controls the I/O pins of vehicle ECU. I/O control unit can be extensively used for the ECU testing as it gives direct control over the I/O of a ECU. We can directly check the performance of the control devices such as fans, actuator relays, or the cascaded control algorithm dependent on an input signal to an ECU [6].

1) Importance of UDS Application

ECUs are considered black box devices in the car. After installation of an ECU in the car user access to its internal ports is restricted. Therefore, if a technician wants to test the hardware of an ECU after installation in the car, the only way is through a communication medium and a application running on chip. If such UDS application is present on every ECU in the car, the I/O testing can be done via an external tester easily. The UDS tester should also be compliant with ISO 14229 [8].

2) Services Under UDS

ISO-14229 provides extensive support for UDS diagnostic services. Overall, 26 number of services are supported including "DiagnosticSessionControl", "TesterPresent", "ReadDataByIdentifier", "Writedataby address", "Input/Output control" etc. Using these services ECU can be programmed for device testing, diagnostics. But following are the services which are essential for diagnostics of ECU inputs and outputs.

a) Diagnostic Session Control (0x10)

An ECU contains a state machine that determines which diagnostic functions are available in the active state. In the context of diagnostic communication, these states are referred to as diagnostic sessions. Fig. 3 provides an overview about the diagnostic session transitions and what the server (ECU) shall do when it is set to another session.

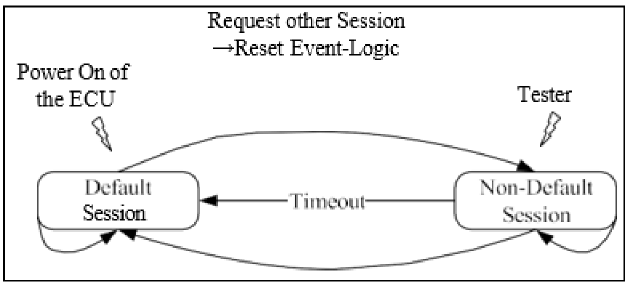

Figure 3: State diagram and session transition logic

In the default session, a subset of the UDS services has to be supported. Any session other than the default session also supports these services. For each session other than the default session, the vehicle manufacturer determines which set of services has to be supported.

Paper ID: SR25317171223 DOI: https://dx.doi.org/10.21275/SR25317171223

b) *Tester present (0x3E)*

This service is used to indicate to a server (ECU) that a client (Tester) is still connected to the vehicle and that certain diagnostic services and/or communication that have been previously activated are to remain active.

This service is used to keep one or multiple servers in a diagnostic session other than the defaultSession. This can either be done by transmitting the TesterPresent request message periodically or in case of the absence of other diagnostic services to prevent the servers from automatically returning to the defaultSession. The detailed session requirements that apply to the use of this service when keeping a single server or multiple servers in a diagnostic session other than the defaultSession can be found in the implementation specifications of ISO 14229.

c) *Read data by identifier (0x22)*

This service requests to read data record values identified by a specific parameter, also referred to as data identifier or DID. The DID is used to identify a specific local data record. e.g. DIDs can be 0xFF24, 0xFF25, 0xFF26, etc. The DIDs may contain data records such as battery voltage, stator current, rotor speed, and calculated load value.

The DID identify data record maintained by the server. The format and definition of the data record shall be vehicle manufacturer or system supplier specific, and may include analog input and output signals, digital input and output signals, internal data, and system status information if supported by the server.

d) *Input and output control by identifier (0x2F)*

This service is used to substitute a value for an input signal and/or control an output of an ECU. Typically, this service bypasses the ECUs appliction software and directly triggers the output circuit, and then directly reads the sensors connected to input circuits. This service can be used to perform various tests on input and output devices as well as ECU ports itself.

I/O control service also uses DIDs to store a parameter control signal. But unlike write data by identifier, the I/O control service sends the control signals to the input-output ports on the ECU. And this is only short-term adjustment of the port signals, the ECU takes back control of the port signals when ECU goes back into the default session. Hence this service can be used for I/O testing purpose.

3. Testing on A ECU

The UDS application stack for an ECU is developed and flashed on it. An 8-bit I/O port is bitmapped by 0x9B29 DID with 8-bit data value in it. An output port is selected

for test. The control status of each pin will be in form of a boolean. The User can give the command for short term adjustment of a pin using 0x03 as byte#4 in the request message as shown in Fig. 4 I/O control transmits request. The definition of request message is taken as per the ISO 14229.

Figure 4: Tester Request and Response

The 0x9B29 is set as I/O DID in the ECU, so it can be controlled via I/O control service (0x2F). When a request message with 0x01 data is sent, the ECU comprehends the commands by checking the DID and will turn ON and remaining pin will turn OFF as only bit 0 is set to 1 in the data. We can also cross check the status of the port by using Read Data by ID service(0x22). The ECU sends the data of requested DID when it receives Read by ID request. The trace of the communication is shown in Fig. 5. Here the 0x01 and 0x00 data is used for the test case. The mask byte can also be used to mask one or more pins on the port for diagnostic control. The mask is specified in byte#6 of the request message.

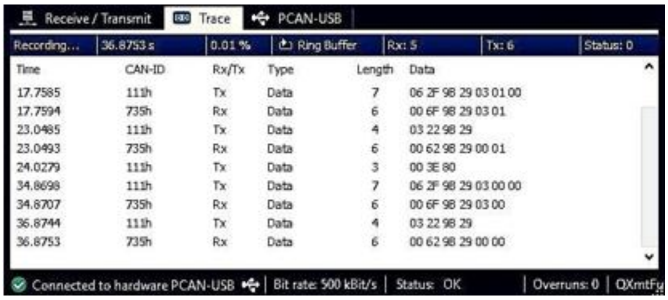

Figure 5: Trace of the communication between Tester and ECU

To check the results on a physical system, a small cooling fan having BLDC motor is interfaced with the ECU as shown in Fig. 6.

Figure 6: Experimental Setup for I/O testing

This setup demonstrated that a diagnostic system can reliably verify the operational status of a control system. And multiple trials confirmed consistent fan activation

4. Conclusion

This paper demonstrates that UDS enables efficient I/O testing of multiple ECUs without its removal, offering a powerful tool for diagnosing vehicle electronics. Its flexibility and control make it invaluable for modern automotive systems, paving the way for faster, less invasive maintenance.

Acknowledgment

The author would like to acknowledge Mr. Sankar Gopalan, Mrs. Geetanjali Khot of TATA Motors, Pimpri for giving the opportunity to work in the field of automotive electronics.

References

[1] P. Kharche, M. Murali and G. Khot, "UDS implementation for ECU I/O testing," 2018 3rd IEEE International Conference on Intelligent Transportation Engineering (ICITE), Singapore, 2018, pp. 137-140, doi: 10.1109/ICITE.2018.8492642.

[2] ISO 14229. Road vehicles - Unified Diagnostic Services (UDS). ISO, Geneva,Switzerland,2013.

[3] Muneeswaran. A., "Automotive diagnostics communication protocols analysis kwp2000, can, and uds," in IOSR Journal of Electronics and Communication Engineering (IOSR-JECE). IOSR, Feb 2015, pp. 20- 31.

[4] Y. Song, T. Wang, A. Xu, K. Wang, and Z. Yang, ``Can based unified customizable diagnostic measure research and realization," in Control (CONTROL)," 2012 UKACC International Conference on. IEEE, 2012, pp. 825-829.

[5] Controller area network (can) overview. National Instruments. [Online]. Available: http://www.ni.com/white-paper/2732/en/. [Accessed: 06-Mar-2018].

[6] Barrett, Steven Frank, and Daniel J. Pack. “Embedded systems design and applications with the 68HC12 and HCS12”. Upper Saddle River, NJ: Pearson/Prentice Hall, 2005.pp 617-656.

[7] “UDS: Protocol Stack | Unified Diagnostic Services | ISO 14229 |” Embitel. [Online]. Available: https://www.embitel.com/uds-protocol- stack-solution-for-vehicle-diagnostics. [Accessed: 06-Mar-2018].

[8] ISO 15765. Road vehicles - Diagnostics on Controller Area Networks (CAN). ISO, Geneva, Switzerland 2006.

[9] Marscholik, C. and Subke, P. "Road Vehicles Diagnostic Communication Technology and Applications". Laxmi Publications Ltd. ,2009, pp 105-145

[10] Rolf Isermann. 2011. "Fault-Diagnosis Applications: Model-Based Condition Monitoring Actuators, Drives, Machinery, Plants, Sensors, and Fault-Tolerant Systems (1st ed.)". Springer Publishing Company, Incorporated.