

Survey on UDS

Bhagyawant S Jumanal, Rudrappa B Gunjatti, Basavaraj kotyal

Department of ECE,

KLE Dr. MSSCET,

Belgaum, India,

Assistant Professor

Department of ECE,

KLE Dr. MSSCET,

Belgaum, India

Department of ECE,

KLE Dr. MSSCET,

Belgaum, India

Abstract — UDS (Unified Diagnostic Services) implementation on connected road vehicles and application on remote diagnostic systems are discussed in this paper. The diagnostic equipment can remotely access vehicles information through wireless connections and acquire vehicle information, such as DTCs (diagnostic trouble codes), speed of vehicle, battery voltage, engine speed, temperature of coolant, etc. Verifications of the remote diagnostic systems at ARTC proving grounds are presented in this paper. Finally we can conclude that the driver safety can be secured and the malfunction vehicles can be diagnosed quickly by using wireless remote diagnostic system.

Keywords- API; CAN; DTC; ECU; Gateway; Internet of things; OSI; Unified Diagnostic Service; etc.

I. INTRODUCTION

The Internet of Things refers to ubiquitous forms of computing in which every object is connected to the Internet in some or the other way. It can be applied to different applications such as smart cities and smart metering. Because of improvement in wireless communications, connected vehicle is one of the emerging applications of IOT. Because of this, vehicle details can be sent to a server online and further analyzed for providing different types of services. The following is one of most popular application for automotive industry. Nowadays, governments and private sectors have invested in these developments. For example the United States Department of Transportation built a test bed in Michigan for checking the improvement in road safety by applying connected vehicle technologies. Depending upon emergency services, automotive manufactures cooperated with telecommunication carriers on e Call services, by which a call can be connected manually or automatically when any accident occurred and also relevant information about that incident can be sent to nearby emergency centre. Hence, the severity of accident can be assessed in advance and emergency services personnel with proper equipment can be served quickly. For different information services, real-time traffic data can be collected via connected vehicle technologies. Then data can be used to support services such as dynamic navigation and other cloud services in order to improve fuel efficiency. The following paper discusses remote diagnostic system [1] for internet-connected Vehicles. First generation of on-board diagnostic requirements was developed by the California Air Resources Board in 1985 and became law in 1988. The rules and regulations require emission related system on vehicles to be operated. A system failure should be indicated through Malfunction Indicator Lamp on dashboard. Hence, the problem can be fixed as early as possible. In contrast to the early OBD system, today's enhance-OBD provides versatile services, such as communication management and transmission of data. Among these services, remote diagnostics via wireless communication is the focus of this paper. UDS protocol is a concept proposed by ISO-14229 [2] standard to cover a big number of diagnostic service types.

According to ISO 14229, an international Unified Diagnostic Service (UDS) standard, following paper reveals software architecture for the implantation of UDS services. Depending upon the UDS software, further, it presents a prototype of remote diagnostic system for electric vehicles. Diagnostic system consists of UDS software, a remote supervisory station, and implemented UDS software. Vehicle in real-time information is sent to the RSS automatically via telecommunication links. If any abnormal situation is detected technicians at RSS will launch the UDS software to check and diagnose condition.

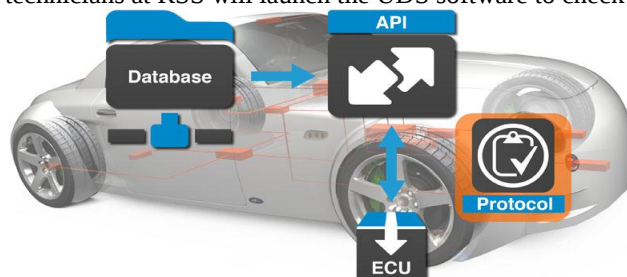


Figure 1 Model of car with UDS protocol

II. COMMUNICATION LAYER

A. Application Layer

Protocol at application layer specifies data link independent requirements of diagnostic services; this allows the external diagnostic tester to control diagnostic functions in on-vehicle Engine Control Unit (ECU). Specifically, tester makes a request of diagnostic service to the ECU through gateway. Then ECU responds with a positive or negative diagnostic message upon a request from the tester.

B. Session Layer

Protocol at session layer specifies services to provide independence between the application layer and transport as well as network layers. Specifically, each time the diagnostic service is called; session layer can signal the completion of the message transmission to the upper layer. At the same time it shall verify that arrival of the service response is within certain waiting duration. A typical time limit between the sending of a service and receiving of the response is 2000 ms. An error code will be produced if any exceeding of the time limit.

C. Transport Layer and Network Layer

ISO 13400 provides the requirements for diagnostic communication between external tester and on-vehicle gateway using Internet Protocol (IP) as well as transmission user datagram protocol (UDP) and control protocol (TCP). On the other hand, ISO 15765-2 A network protocol tailored to meet the requirements of CAN-based vehicle network systems specifies the requirements for diagnostic communication between the on-vehicle ECU and gateway. The important task here is to ensure data is compliance with certain formats.

D. Data Link Layer and Physical Layer

The protocols here consist of ISO 11898-1 and IEEE 802.11. IEEE 802.11 gives the data link layer and physical signaling of the IP network outside vehicle, while ISO 11898-1 gives data link layer and physical signaling of the controller area network inside the vehicle.

III. IMPLEMENTATION OF UDS

UDS protocol is developing to use with Controlled Area Network buses. CAN frame contains a data field and a CAN Identification field. The values of these fields are determined by higher layer protocol such as UDS, Calibration Protocol, Universal Calibration Protocol, CAN and On-Board Diagnostics II, which have different types of frame formats. The OSI layer must be classified first for an efficient implementation. The classification of each protocol layer depends upon the frame format. Hence, the OSI model is defined in order to implant UDS protocol in an ECU. OSI model which supports every protocol on the same CAN bus is as shown in Figure 1.

Application Layer	XCP	CCP	UDS	OBD2
Presentation Layer	N/A (Not Available)		N/A	
Session Layer	N/A		N/A	
Transport Layer	N/A		Diagnostic on CAN – Transport Layer (ISOTP)	
Network Layer	User defined		Diagnostic on CAN – Network Layer (ISOTP)	
Data Link Layer	CAN			
Physical Layer	CAN bus			

Figure 2 OSI layer model

A. Network Layer Development: Network layer needs to have the ability to differentiate the type of protocols. XCP and CCP protocols are calibration protocols. A calibration protocol does not have a specific requirement for CAN Identification definition as in diagnostic protocol. Therefore, the developers should define address of vehicle components by using eleven bit CAN identifiers for all calibration protocols. Developers must define address of diagnostic protocols following ISO standard, which uses 29-bit CAN identifiers, to give full information of packets. This provides less concurrency of addressing the calibration protocol and diagnostic protocol. Frames of CAN should be stored in CAN buffers and remapped to a structure type variable for reasons of efficiency.

B. Transport Layer Development: Transport layer of UDS protocol on the CAN bus is defined by ISO-15765. The transport layer uses the Protocol Control Information byte to control message flow. The PCI byte is control byte that has information on the

frame type and frame type parameters. There are 4 types of frames, Single Frame, First Frame, Flow Control Frame and Consecutive Frame. Frame formats of UDS are shown in fig 1. Response data from an ECU smaller than 8 bytes, including the PCI byte, must use the Single Frame type to transmit messages. If it exceeds eight bytes then it has to be encapsulated in multiple CAN frames. There might be a huge load of response data sent from the ECU in one transmission, which cause the bus to overload. Flow control method is a countermeasure to this circumstance.

C. Application Layer Development: UDS Protocol resides in the application layer. It processes data which is stored in message buffers. The protocol will process completely received message in a buffer to ensure the correctness of corresponding request. Hence, a UDS can only be called by the use of a Single Frame and last Consecutive Frame. Server must then react to request and send a response signal following the defined standard of ISO standard. Development of application layer is done by completing each service of UDS function following specification of ISO standard and SAE standard. The standard provides a specification of services, request messages and response messages. Each service of UDS can be developed differently as long as format of response messages and request messages are the same as standard specified.

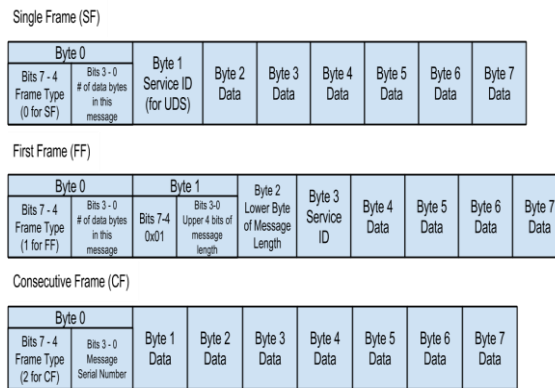


Figure 3 Frame formats of UDS

IV. SOFTWARE ARCHITECTURE OF THE IMPLEMENTED UDS

Through the human machine interface (HMI), users specify address of the diagnostic service and the on-vehicle ECU along with some options. When any service request is triggered, the application layer of the diagnostic tester initially makes sure the request is illegal then passes the request message to session layer. This layer marks down departure time of the request then waits for the arrival of response from the on-vehicle ECU. Network and Transport layer packs the request message to a specific format and passes it down to data link and physical layers for the transmission. The transmission result will be reported to application layer so as to wait for make a retransmission or service response. Upon receiving of the service response from on-vehicle ECU, the message is decoded at network and transport layers and delivered to session layer, which checks if arrival time of the service response within time limit. Finally, application layer checks whether the diagnostic services have been rendered.

The paper Microchip's Explorer 16 Demonstration Board is used as platform of the gateway implementation. The software is ported to the PIC32MX795F512L micro-processor on this board. This demo board accesses the Controlled Area Network bus through Microchip's CAN transceiver [3] and Wi-Fi channel through ROVINC Networks' RN-171 Wi-Fi module. Main task of the gateway is to transform package format to CAN from TCP/IP protocol, and vice versa.

On-vehicle ECU upon receiving any diagnostic request, the message is decoded at network and transport layers delivered to the session layer, which marks down the arrival time of the service request and waits for departure of diagnostic response from the upper layer. Once the service request is arrived, application layer makes specific response and sends the responding message down to the layer below it. Session layer checks whether the response is made within time limit. The request message is then packed into a specific format at network and transport layers passed down to data link and physical layers for the transmission. The transmission result will be reported to application layer so as to check if a retransmission is required.

V. FUNCTIONAL VERIFICATIONS OF IMPLEMENTED DIAGNOSTIC SERVICES

To verify functionalities of the UDS protocol software, we provide functional test before combining our UDS software into the diagnostic. The external diagnostic tester can access the on-vehicle Engine Control Units through gateway. The main intention is to create simulate a situation that a consumer's vehicle breaks down and the manufacturer can offers roadside assistance remotely. In the following section, we demonstrate 3 UDS functionalities, including time-out function, clear and read DTC. The installation of the developed UDS services on a self build electric vehicle Time-out is one of the important functions that UDS software should implement, because the communication link maybe lost due to varying quality of radio signal. If any response message does not arrive within time limit, then a time-out warning will be raised. In this time-out verification, the vehicle will be kept out of the reach of the external diagnostic tester.

Clear DTC and Read DTC are 2 essential functions the UDS should support. DTC codes along with their

status are how On Board Diagnostic identifies and communicates with technicians where and what on-board problems present. This paper adopts three DTC codes, 0x110114, 0xd54233 and 0x124251 which represent overheat of motor, over current and over speed of motor respectively. The track was too much steep that the motor of the electric vehicle was overheated and the MIL was illuminated to alert the driver. Then external tester was used to read the DTC as repair information. As a result, the tester successfully read the DTC codes of over current and overheats. After repairing the vehicle, the recorded Diagnostic Trouble Codes should be erased. Hence, external tester requested the Clear Diagnostic Information (0x14) service to remove the DTC codes.

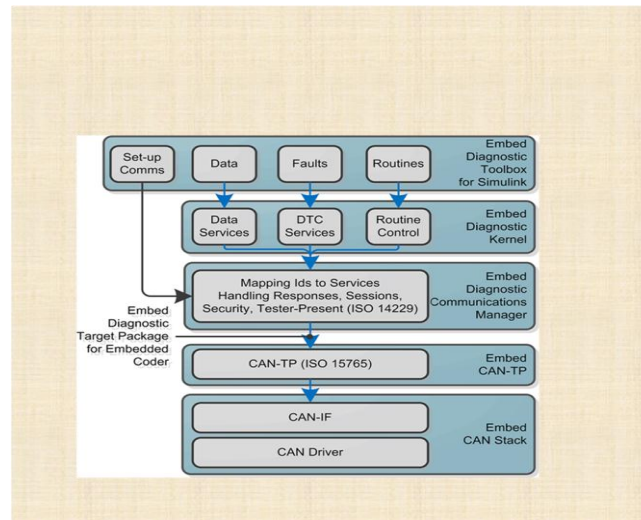


Figure 4 Block diagram of UDS

VI. DEVELOPED REMOTE DIAGNOSTIC SYSTEM

Depending on UDS software, this paper [4] further builds an advanced remote diagnostic system, where 2 Controlled Area Network buses are deployed. Bus 1 is used to transmit information of primary controllers like MCU, BCU, and the gateway. CAN bus two is used to transmit information of other controllers such as AAC. Vehicle sends its information, such as voltage, speed, current, temperature, to the supervisory station periodically. This information is used to monitor the status of the vehicle. If some abnormal situation is detected, compared to previous collected data, the personal at supervisory station can further inspect vehicle immediately by launching the UDS software services. Technicians at the supervisory station can give the driver instructions for safety concerns or provide roadside assistance services by sending out rescue team with proper equipment according to the acquired DTCs. The installation of the developed UDS on a self build electric vehicle. For the purpose of explanation, one diagnostic case is given below. Temperature and current of VCU were below upper limits but more than normal values. The supervisory station recognizes this condition and monitored the duration of events. After duration exceeded a threshold, UDS software is launched to acquire the DTCs. The DTCs indicated that the motor torque was overloaded. Therefore, the vehicle is instructed to pull over immediately and technicians sent out to inspect the vehicle. It turned out that the problem was resulted by mechanical brake components. The problem was fixed Hence the driver safety will be secured.

VII. EXAMPLE – FLASH PROGRAMMING

As a practical example of the UDS diagnostic services, let us consider the typical structure of a flash programming, as illustrated. The diagnostic tester sends a Growl Read Data by Identifier. With this request, it reads the hardware ID and software ID from the controller to see which device it exactly right. Then, the diagnostic tester, the control unit switches to a special diagnostic session. Not the actual diagnosis session for the program, but a session in which there are a number of advanced services available. This is done with the diagnostic service diagnostic Session control. This advanced diagnostic session asks the diagnostic tester, the control unit whether the preconditions are met for flash programming. This is typically that the programming can be done only when the vehicle that the engine must be made, etc.



Fig 1 Basic sequence of flash programming

Then, the diagnostic tester usually with the service Communication Control the fault memory and off the bus communication in other controllers. This advanced diagnostic session has served its purpose. Now the diagnostic tester on the diagnostic service diagnostic Session control to the programming session. At least now is a Security Access necessary. Thereafter, the diagnostic tester usually sends the so-called fingerprint of the control unit. This is information that is stored in the ECU memory permanently, to indicate that programming. It typically has a workshop identifier in the memory of the controller is written, can be enacted so that afterwards who has reprogrammed the ECU. Before the flash memory can be reprogrammed, it must be deleted. This is achieved by calling a routine in the ECU memory of the diagnostic service routine control done. Thereafter, the service request download the actual programming operation is initiated. This service allows the controller will also be notified of which are loaded into memory the data and how much data can be expected. Now starts the actual download of data in a loop with the service transfer data. The storage area is transmitted here in blocks. At the end of the diagnostic tester says the control device transfer exit Now that all data has been transferred. After examining the data transmitted in the control unit now takes place, the actual flash process. Typically, the programming operation will take some time. During this time the controller is not able to process requests from the tester. Therefore, the control unit the service transfer exit usually with a negative response and the error code Response Pending. Reply only when the programming is completed, the controller sends a positive confirmation transfer exit.

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

In order for the other ECUs in the vehicle also restore the status quo, will Communication Control the normal bus communication again and the fault memory in the other control devices is turned on again. Hence the download process is complete.

VIII. DIFFERENCE BETWEEN KWP2000 AND UDS

- Event triggering and periodic transmission are applicable only in UDS.
- Positive response suppression for tester present in KWP 2000.
- Error memory management.
- Transfer of measurement values, only two-byte identifiers are available in UDS.

IX. APPLICATIONS OF UDS

- UDS is currently being launched in many areas of automobile industry ,but it is already in use in other industries
- Diagnostic interfaces: All our EDIC diagnostics interfaces features UDS
- Communication interfaces: For our CAN interfaces, we implemented UDS on the PC
- Customized solutions: Numerous customized implementations of diagnostics stack in the ECU.

X. CONCLUSIONS

We can say that UDS can be very useful for the server to debug or repair the errors in an automotive electronics through wireless communication to ECU. The paper has demonstrated the implementation of UDS software for remotely connected vehicles and its application on diagnostic systems. Depending upon the UDS software, this paper has also presented an advanced remote diagnostic system. Supervisory station constantly monitors the status of vehicles. Upon the detection of any abnormal situation, technicians at the supervisory station can launch the UDS software for further inspection of the vehicle. Hence, defects in vehicles can be repaired as soon as possible and safety of driver is achieved.

REFERENCE

- [1] J. Lin, S.-C. Chen, Y.-T. Shih, and S.-H. Chen, —A study on Remote On-Line Diagnostic System for Vehicle by Integrating the Technology of OBD, GPS, and 3G, World Academy of Science, Engineering and Technology, 2009.
- [2] ISO 14229. Road vehicles - Unified Diagnostic Services (UDS). ISO, Geneva, Switzerland 2006.
- [3] ISO 15765. Road vehicles - Diagnostics on Controller Area Networks (CAN). ISO, Geneva, Switzerland 2006.
- [4] ISO 15031. Road vehicles – Communication between vehicle and external equipment for emissions-related diagnostics. ISO, Geneva, Switzerland 2006.