

Developing the interfacing window for UDS protocol services to ECU

Arun Radde

Telecommunication Department

SIT, Tumakuru, India

e-mail: arunpraddi@gmail.com

Dr. T N Chandrika

Telecommunication Department

SIT, Tumakuru, India

e-mail: tnchandrika@sit.ac.in

Abstract— The electronic controller devices in a current automobile are named as Electronic Control Units (ECU). The ECUs backing Controller Area Network (CAN) communication protocol. CAN stands for communication between vehicles ECUs since it is a dependable, robust and it has responsibility confinement quality. When the amount of on-board ECUs numbers is increased, and the diagnosis of a exact ECU becomes a complex in tasks. The analysis of an electronic devices and electrical circuits is to be done along with a standardized communication protocol, UDS (Unified Diagnostic Services). In/Out is tested by UDS service id and it's applications is implemented, the data which the ECU we absorbed is from the sensors generic sensors example angle detection, ambient, temperature and such other sensor this data fo the easy observation for the tester, since UDS have other service ID we can implement other tester easy services also in this paper for some service id has been implemented this has been done using the tool CANoe and backend code implementation is done in CAPL scripting language and it is presented in this paper.

Keywords- UDS, CANoe, ECU, CAPL, ISO-14229, DID

I. INTRODUCTION

In the world of growing automotive industry which is leading the world to shrink is also reducing the manual effort in driving the car by making it autonomous, which is leading to the introduction of Autonomous Driving Assistance System. The concept of autonomous leads to the introduction of more embedded components into an automobile. Embedded being the mix of both hardware and software has to be fused in a proper way for their functioning.

The embedded component used here is referred as Electronic Control Unit (ECU) which forms the major role in the ADAS driving automobile. Since driving is a real time task in order to check the ECU it is necessary for us to check it in a virtual environment for its performance. The developing, testing and analysis of an ECU is very necessary. This is done by the implementation of Unified Diagnostic Services (UDS). The standards of UDS protocol has been defined and documented in ISO- 14229. UDS sevices had made the tester to easily interact with ECU and know its status and condition. There are also other standards like ISO 15765 which is used in sending long length data and for communication over the CAN channel.

II. OVERVIEW

A. Electronic Control Unit (ECU)

Electronic Control Unit is an embedded system of hardware and software. Any autonomous car consists of several ECU numbering from one to 100 or many in upcoming automobiles. The ECU may be a radar ECU, breaking system ECU, steering ECU, airbag ECU etc.

It can be observed that all the ECUs forms the major part of automobile and has to be well functioning in the real time scenarios. ECU basically consists of a sensor and a microcontroller which will be flashed with an operating software according to its working. As a part developing the software it is necessary to check the ECU for its performance before installation. The UDS services are used to read the data, write the data, control the ECU in its off condition.

B. Controller Area Network (CAN)

In the early automobiles the complexity of wiring was considered to be major issue. Also the number ECUs were less. But with growing technology number of ECUs were increased hence the complexity of wiring also increased. This leads to lot many disadvantages such as poor communication, high cost and less reliability. So the major automotive industry came up with solution of CAN protocols. This protocol connected all the ECUs with single wire which led to easy digitised communication among the ECUs. It has advantages like less complexity in connections, high speed data transmission, easy to check if any defect occurs which makes it more reliable.

This CAN protocol has two different frame format for the data transmission based on the speed. They differ in terms of arbitration field ID. The 2.0A has 11-bit arbitration field ID whereas the extended frame format 2.0B has 29 bit ID. The extended frame format is referred as CAN FD due to its flexible data rate. Due to the high transmission bit-rate, CAN FD message always get a smaller transmission and reception time with the increasing of the transmission bitrate.

The complete frames are as shown in Fig. 1 [3].

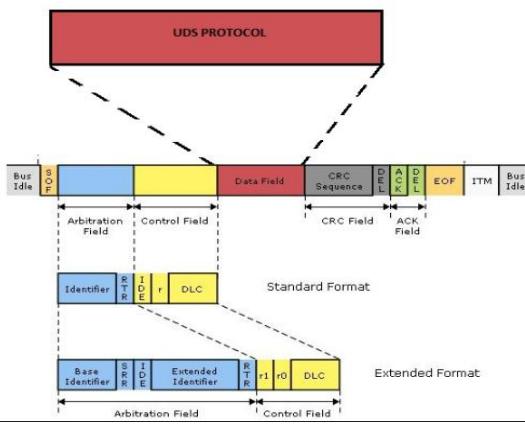


Figure 1. The Frame format of CAN and CAN-FD

The presence of cyclic redundancy check(CRC) bit and acknowledgement(ACK) bit makes sure the transmission is error free [4]. The end of frame and start of frame avoids the collision among the frame making it more reliable for the user. The CAN-FD being the new to the industry is still has the long way to be implemented in all the automotive industry according to the requirement.

III. OSI model UDS implementation

The implementation of UDS services ID's is done according to the OSI model(layers). UDS being implemented and it is found in the application layer of the OSI model. The communication is as shown in Fig. 2.

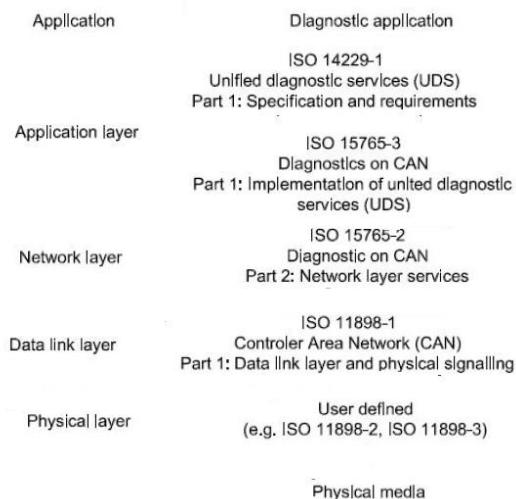


Figure 2. The OSI layer aimed in UDS applications

CAN being the physical part finds its place in the physical layer. CAN utilizes two layers of OSI model one is physical layer and other is data link layer. The data link layer is further divided as the logical link control layer and the medium access control sublayer.

IV. UNIFIED DIAGNOSTIC SERVICES

If we find any trouble in the automobile the holder takes his automobile to the service where a tester will be connected to the CAN channel and the diagnosis of the automobile will be completed. This diagnosis is done by sending the set of services which are defined by the ISO-14229. These services are used for complete analysis of the vehicle status.

Significance of UDS protocol is huge in establishing communication relation between the tester and the ECU. UDS specifies different diagnostic services to achieve diagnostic functions. There is a certain frame format to be followed to establish a convenient communication between a client and the ECU. The properties of unified diagnostic service are as follows:

The tester controls the diagnostic and the ECU only
Responds to the tester request.

The diagnostic service is nothing but information need to be sent to ECU to perform particular action like reading the data from the ECU.

The diagnostic services are denoted by unique service ID.

A. Importance of UDS Application

Once the ECU is installed in the car user cannot access the memory of an ECU. Hence if a tester wants to check the health of an ECU it has to be done through communicating with the ECU. The communication with the ECU is done with the help of UDS services. The ECU is configured such that it understands the request sent by the client and responds to the request accordingly. This request and response is done by utilising the UDS services as defined in the ISO 14229.

B. Types of services under UDS

UDS supports several diagnostic services such as: Diagnostic Session Control, Tester Present, Read data by identifier, Write Data By Identifier, ECU Reset, Communication control, Read DTC information, Security access, Clear DTC, Routine control etc. By using these several necessary tasks can be performed on the ECUs hence establishing the communication between tester and ECU. Some of the services related to input/output to the ECU are as follow:

1) Service- Diagnostic Session Control (0x10)

An ECU support only few services in a particular session. Not all the services are supported in a single session. There are three main sessions in the diagnostic session control. They are default session, programming session and extended session. The default session is the session where normal operations are done which supports basic diagnostic services like read data by identifier

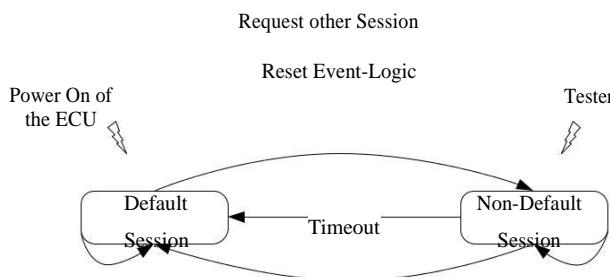


Figure 3. State diagram and session transition logic

In the programming session, a subfunction of the session control is necessary for programming of the ECU can be done. Any session except the default one are supported in the extended session. For session to switch over particular subfunction value has to be sent in the request frame.

2) Service- Tester Present - (0x3E)

The *0x3E* service is being used to tell the ECU that Tester used for the analysis of the ECU is still being connected and a particular diagnostic service is being implemented and the ECU still needs to be in working condition to complete the diagnostic service action to be completed.

This service is utilised to indicate no subfunction to be performed and used only to indicate the tester presence. The tester present request can be sent periodically to the ECU to indicate the presence of the tester. Unlike other services this service does not contain the subfunction. This service is an important as the tester present tells the ECU to be in a particular session i.e to be in default or programming session.

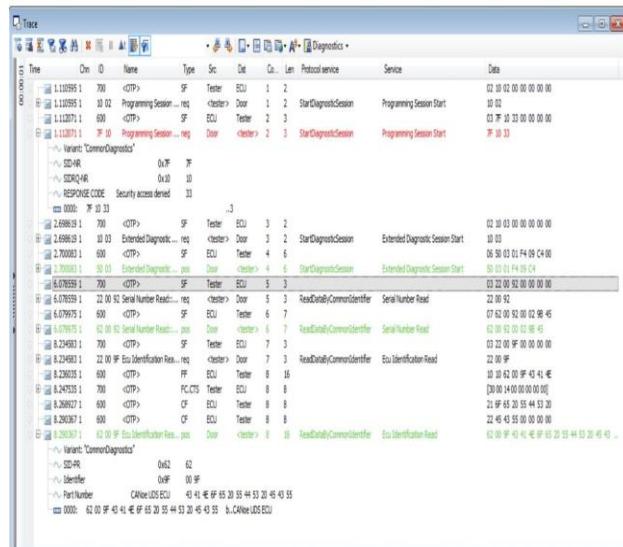


Figure: session failure and successful

1) 3) Service- Read data by identifier- (0x22)

The service read data by identifier is utilised to read the data from the ECU using the Data Identifiers which are referred as DID. These DIDs are used to indicate particular set of data in ECU. DIDs can be represented as 0x5555, 0x5556, 0x104B, 108B etc. The DIDs are used as function name which when requested gives the particular data information such as ECU voltage, current, VIN (vehicle identification number), yaw rate, steering angle, vehicle variant, vehicle speed.

The DID identify data and produce the response which may be a negative or positive response. The positive response is identified when service ID(SID) 22 is logical or with 40. If the response format include 62 then its considered as a positive response. The negative response is identified by 7f byte in the response frame.

V. Tool

CANoe is an user interface software tool used for the development, analysis and then testing of the activities over the CAN bus that is formed when multiple ECUs are connected. It supports developers and testing engineers of Original Equipment Manufacturers (OEM) for developing the software that is to be flashed on the ECU –starting from the setup to interconnection on a virtual CAN network can be created and observed in the simulation setup option. This setup is used to see the behaviour of the ECU over the CAN and also its communication with other interconnected ECUs. Hence before the usage of an ECU it is analysed, tested and also integrated to check for its performance. This helps us to detect problems at the early stage of development and correct them before its growth. Availability of panel window makes it easy to develop user friendly keys for the signals.

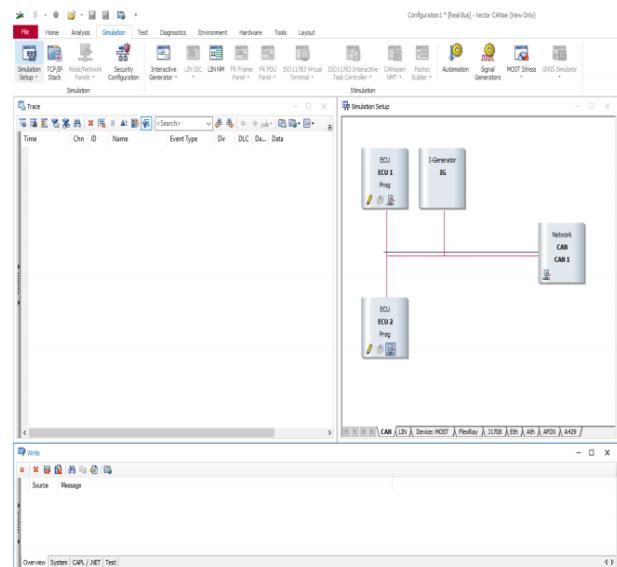


Figure: CANoe Interface

Comparable to the C programming language, CAPL, or Communication Access Programming Language, is the event-oriented programming language used in the CANoe environment. The main idea behind CAPL scripting is based on the behaviour of an ECU. It is event driven that is it responds only when an event occurs, including: on timer, on key, on start, on message, on occurrence of particular signal. Usually it is not possible to have two procedure with same events. However, there is a possibility that same event can lead to two difference procedures.

The creation of CAPL is done in the CAPL browser and Using CANoe in combination with CAPL makes it possible to create customised tools necessary for the analysis of the ECU. The signals transmitting and receiving over the CAN bus by the ECU can be observed in the trace window as shown in the figure 4. The below result is obtained by the implementation of diagnostic service read data by identifier (0x22).

- [4] Controller area network (can) overview. National Instruments. [Online]. Available:<http://www.ni.com/white-paper/2732/en/>. [Accessed: 06-Mar-2018].
 - [5] 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.
 - [6] "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].
 - [7] ISO 15765. Road vehicles - Diagnostics on Controller Area Networks (CAN). ISO, Geneva, Switzerland 2006.
 - [8] Marscholik, C. and Subke, P. "Road Vehicles Diagnostic Communication Technology and Applications". Laxmi Publications Ltd. ,2009, pp 105-145
 - [9] 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

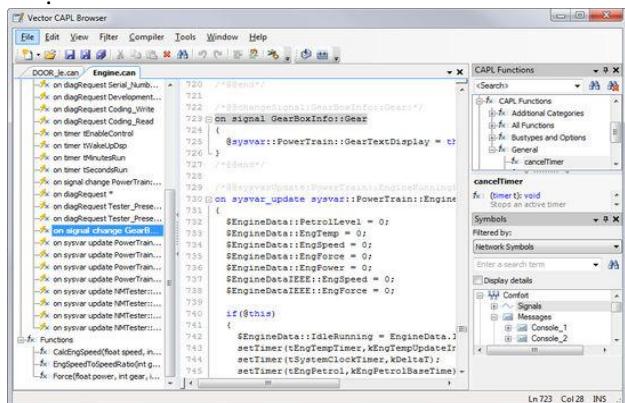


Figure: CAPL Scripting

VI. CONCLUSION

The development of a window using CAPL scripting helps the user to interact with the ECU in an easy way. Ultimately the implementation of an UDS service is made more easy for the analysis and testing of an ECU.

ACKNOWLEDGEMENT

The authors would like to acknowledge Mr. Swaroop G, Mr. Sumith N, Mrs. Nikita Gandhi of Continental Automotive, BLR for giving proper support to implement the above work in the automotive electronics.

REFERENCES

- [1] ISO 14229. Road vehicles - Unified Diagnostic Services (UDS). ISO, Geneva, Switzerland, 2013.
 - [2] 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.
 - [3] 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.