Turkish Online Journal of Qualitative Inquiry (TOJQI)
Volume 6, July 2021: 6216- 6223

Research Article

Vehicle Diagnostic Process by Reverse Engineering

P. Sivakumar [a], Rajalakshmi R[b]

[a,b]Department of Electronics and Communication Engineering, Kalasalingam Academy of Research and Education, Srivilliputur

Abstract

With the advent of virtual vehicular technology, the digital components of customer vehicles have become progressively associated to aid automotive expertise, driving experience and emissions control. Vehicle ECU simulators play a major role in any diagnostic content development activities. To develop function of automobile by electronic controlled system, the difficulty of failure diagnosis is increasing. Rising functionality and complication of prevailing automotive ECUs raise demands on development. Existing systems available in the market which are capable of logging field data is complex, costly and are not able to determine the overall performance of the vehicle due to its limited functionality. The bus configuration of this bus is not publicly confessed by the car manufacturers. Therefore, here needs to apply Reverse engineering techniques. Nevertheless, performing this approach manually is cumbersome and time consuming. Design Engineer often require vital information related to field operating condition, which can be used to improve and optimize their designs So that, propose an automation of the analysis steps of reverse engineering to upgrade and facilitate the process. This has motivated us to develop a low cost and effective simulator that supports all the CAN protocols and acts as a complete vehicle by simulating multiple ECUs simultaneously.

Keywords: CAN, Electronic Control Units, UDS protocol, Simulation and Onboard/Off board Diagnostics.

1. Introduction

In terms of revenue, automotive industry is one of most significant contributors to world economy. Term that always associated with automotive industry is innovation. Automotive industry invests hundreds of billion dollars each year on research and development. From design rooms to factory workshops, auto companies are delving to new technologies and vehicle concepts that have the ability to transform the Automobile industry. The colossal demand and need for digital technology along with the use of customers around the world in day to day life, is having an extensive impact on the automotive industry. New concepts and intelligent cars are moving from drawing board to the streets. Google has partnered with 3 auto companies – Bosch, Continental and Delphi, to build a multi-brand diagnostic scan tool for car. Converting all these concepts to reality involve the development and automotive diagnostic testing which comes with many challenges. Automobiles these days are equipped with various electronic control units which provide vehicle driving information.

Car diagnostic systems easily and reliably point to problem areas in a car's engine or elsewhere using advanced software, with built-in sensors, processors and microchips. Diagnostic tests may show problems with a vehicle's engine, exhaust system, brakes, transmission and other major components, as well as fuel injector, air flow and coolant, ignition coils, and throttle performance issues. Diagnostic tools can also check the computer system of a vehicle for manufacturer alerts and stored car history records, giving technicians a full picture to perform the best possible repair.

Reverse engineering is widely used in a variety of areas, including digital engineering, entertainment, automobile, household goods, microchips, chemistry, batteries, and mechanical designs. For example, when a

new model vehicle is introduced to the market, competing manufacturers may purchase one model and decompile it to learn how it was built and how it operates.

Real Vehicle Environment the OEM diagnostic tool is connected to the Vehicle's OBD- II connector [2],[3],[4] through VCI which reads the vehicle information like ECU diagnostic trouble codes, real time vehicle parameters and calibrate the ECU's fitted in the vehicle. To detect and diagnose vehicles, OBD is incorporated into ECUs. Faults are detected, and the fault-relevant diagnostic problem codes (DTCs) are set and stored in the ECUs' memory for subsequent off-board, return-to-dealer fault inspection and correction [1]. In-order to simulate the vehicle environment in bench level a virtual vehicle simulator is developed which communicates with OEM diagnostic tool through Peak CAN hardware. Simulated Vehicle Environment. This virtual vehicle simulator works like a virtual vehicle which can simulate multiple ECU's.

2. Vehicle Networking And Diagnostic Protocols

A protocol is a set of rules that governs the communications between computers on a network. In order for two computers(In Vehicle ECUs) to talk to each other, they must be speaking the same language.

communication protocols:

1. Local Interconnect Network
2. Media Oriented System Transport
3. Flexray

4. Controller Area Network

2.1. CAN – Controller Area Network:

The CAN protocol is a method of communicating between different electronic equipment embedded in a vehicle, such as engine management systems, ABS, gear control, active suspension, lighting control, air conditioning, airbags, central locking, and so on. A concept suggested by Robert Bosch GmbH in[5] to improve the efficiency of vehicles, making them more dependable, safe, and fuel efficient**.** With advancements in the electronics and semiconductor industries, the mechanical systems in automobiles were being replaced by more powerful electronics systems with increased performance. If the number of electronic devices increased, contact signals with more complex interrelationships were used. As a result, vehicle engineers' lives were made more complex when they built devices in which one computer interface needed to coordinate with another in order to function. Recognizing the problem of connectivity between various electronic modules, Robert Bosch develops a new protocol called CAN, which was published in 1986.

Bus Arbitration:

It is a function that resolves conflicts as two or more nodes attempt to send a message at the same time. Using this method, whenever the bus is open, any unit may send a message. If two or more units begin switching at the same time, access to the bus is disrupted; however, this issue can be fixed using identifier arbitration. During arbitration, each transmitter compares the value of the transmitted bit to the value of the bit on the bus. If the bit rate is constant, the node will continue to transmit bits. However, if the transmitted bit rate varies from the bus value, the dominant bit often overwrites the recessive bits. An 11- or 29-bit identifier and a remote transmission (RTR) bit comprise the CAN message arbitration area. Priority is assigned to the identifier with the lowest numerical value. RTR clearly distinguishes between remote frames where RTR is recessive and data frames where RTR is dominant. If both a data frame and a remote frame with the same identifier are started at the same time, the data frame will win. In the arbitration approach, neither details nor time is wasted.

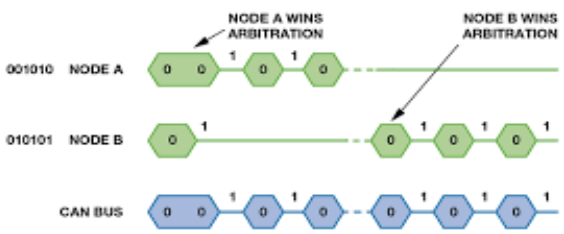


Figure 1. Bus arbitration of CAN

Message Framing

May messages are sent in a format known as frames. A frame is a fixed structure that transports a meaningful sequence of data bits or bytes through a network. The MAC sublayer of the Data Link Layer is in charge of message framing. Frames are classified into two types: regular and expanded. These frames can be distinguished by their identifier fields.

Standard frame

Figure 2. Message Format of CAN

SOF- is an abbreviation for "Start of Frame bit." The frame bit signals the beginning of a message and is used to synchronize the nodes on a bus. The beginning of a frame is marked by a dominant bit in the field.

IDENTIFIER - It has two functions: one is to specify which node has access to the bus, and the other is to classify the type of packet.

RTR- stands for Remote Transmission Request . The request specifies whether the frame is a data frame or a remote frame. When it comes to data frames, RTR is dominant, and when it comes to remote frames, it is recessive.

IDE-The frame format is defined using the IDE – Identifier Extension. The periodic frame has the dominant bit, while the extended frame has the recessive bit.

R0 - Reversed bit that is not actually in operation yet is saved for future use.

DLC – Data Length Code – is a four-bit data length code that indicates the number of bytes being transmitted.

DATA– Used to store up to 64 bits of device data for transmission.

CRC-The checksum of the previous program data is stored in the CRC – 16-bit Cyclic Redundancy Scan – for error detection.

The ACK area is made up of the ACK slot and the ACK delimiter. When the data is correctly collected, the reader rewrites the recessive bit in the ACK slot as the dominant bit.

EOF- stands for End of Frame. A 7-bit field that indicates the end of a CAN frame (message) and disables it.

IFS - Inter Frame Space, which defines the minimum number of bits that must be used to separate consecutive messages.

It serves as an intermission between two frames and is made up of three recessive bits known as intermission bits. This time allows for internal processing nodes to complete their tasks before the next frame begins.

Advantages:

1. Low price: Since the CAN serial bus has a decent price/performance ratio. They are also relatively inexpensive, since they are powered by high volume production of low cost protocol modules.

2. Reliable: Because of the excellent error detecting and handling systems used by CAN, it provides high efficiency transmission. It is therefore very resistant to electromagnetic interference.

3. Adaptability: May Nodes be quickly linked and separated.

4.Fast: It has a data rate of 1Mbit/s at a bus length of 40m.

5.Multi-master connectivity and broadcast capability: Any node on the bus may access it. Messages may be sent to a single, many, or all nodes.

6.Standardized: The ISO-DIS 11898 (high speed applications) and ISO-DIS 11519-2 standards have standardized CAN (low speed applications).

2.2. Diagnostic Protocol:

2.2.1. KWP 2000:

The 2000 keyword protocol complies with the ISO 14230 standard on-board diagnosis[8] (OBD) protocol. It defines a standardized collection of contact codes used by vehicle ECUs for data sharing in accordance with the OBDII regulatory standard data exchange guidelines. The KWP 2000 in-vehicle networking device is compliant with both K-Line (ISO 9141) and CAN (ISO 11898)[11]. The KWP 2000 protocol employs a physical layer that is similar to ISO 9141-2 for bidirectional serial communication with the controller over K-line. To wake up the automotive ECU, the protocol also employs L-Line (optional) unidirectional communication. The average data rate of KWP 2000 is between 1.2 and 10.4 kilo baud, and data fields within the message will contain up to 255 bytes.

2.2.2. UDS protocol:

An off-board diagnostic device is ISO 14229 [7] (United Diagnostics Services Protocol). It adheres to ISO 14230-3 [10] (KWP2000) and ISO 15765-3 (Diagnostic Communication over Controller Area Network (DoCAN)) standards[6],[9].

The UDS allows for a maximum message size of 8 bytes. For messages larger than 8 bytes, the UDS protocol employs the ISO 15765-2 layer, an international standard for data packet sharing over a CAN Bus. The implementation of UDS diagnostics is independent of the underlying physical layer and is also compatible with LIN and CAN in-vehicle networks.

UDS was created as a diagnostic protocol to unify all previous diagnostic standards and provide a single valid range of diagnostic services for automotive ECUs. As a result, the integration of the UDS protocol stack has meant that the extra costs for the implementation of diagnostic communication applications have been reduced.

S.No	UDS Serivces	Service Identifiers
1	DiagnosticSessionControl	10
2	ECUReset	11
3	SecurityAccess	27
4	TesterPresent	3E
5	ReadDataByIdentifier	22
6	WriteDataByIdentifier	2E
7	ClearDiagnosticInformation	14
8	ReadDTCInformation	19
9	InputOutputControlByIdentifier	2F
10	RoutineControl	31

Figure 3. UDS Service IDentifies

UDS Functional Group:

1) Diagnostic and Communication Management

2) Data Transmission Functional Unit

3) Stored Data Transmission Functional Unit

4) Input Output Control Functional Unit

5) Remote Activation of Routines Functional Unit

6) Upload Download Functional Unit

3. Reverse Engineering Concept

Engineering is the discipline concerned with the architecture, construction, manufacturing, and upkeep of materials, processes, and structures. At a higher degree, there are two forms of engineering: forward engineering and reverse engineering.

Forward engineering is the conventional method of progressing from mathematical designs and high-level abstractions to physical device execution.

Reverse engineering is the practice of replicating an original part or subassembly without the use of sketches, documentation, or a virtual model.

Reverse engineering is the method of evaluating a device in order to:

1.Determine the system's elements and their interdependence.

2.Construct representations of the structure in a different manner or at a higher degree of abstraction.

3. Make a physical image of the device.

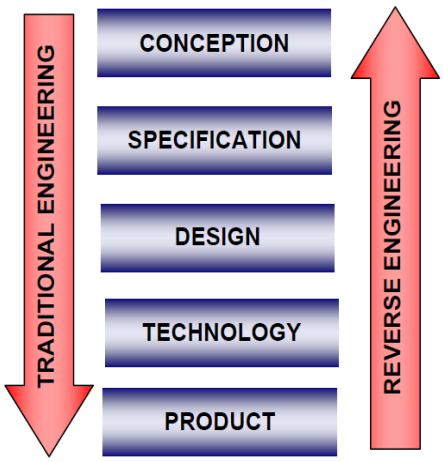


Figure 4. Reverse Engineering flow

The aim of reverse engineering is to reduce product production time. In today's highly competitive world market, producers are constantly looking for innovative approaches to and the time it takes to get a new product to market. Rapid product production refers to recently designed technology and strategies that help suppliers and designers fulfill the demands of shorter product development cycles. Injection molding industries, for example, would greatly reduce tool and die production times.

By collecting the physical dimensions, characteristics, and properties of the component, reverse engineering allows the replication of an existing device. A well-planned life-cycle analysis and cost/benefit analysis should be performed before undertaking reverse engineering to justify the reverse engineering programs. In most cases, reverse engineering is cost effective.

Advantages:

1. Lesser maintenance

2.Competitive advantage

3.Quality improvement

4.Cost saving for developing new products

4. Design And Methodology

Block diagram:

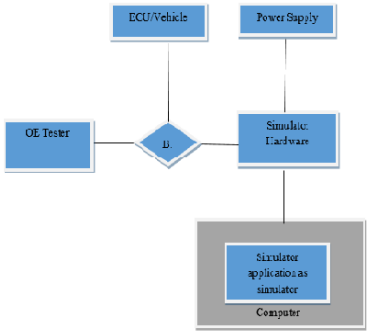


Figure 5. Connection between vehicle ECU and External OE tester

1. Wake up the system: We need to wake up the system with help of VIN number and VDIAG number.

2. To simulate the module with KWP/UDS services IDs.

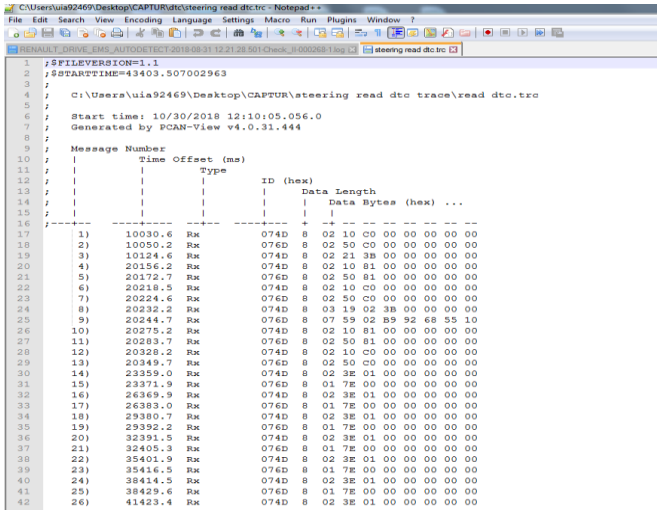
3. This communication helps to identify which diagnostic protocol has used in that ECU. To send and receive messages, devices that communicate using the CAN protocol are linked together using a standard serial bus. For data transmission between nodes to occur, they must have the requisite hardware and software integrated within them.

4. To read the information with corresponding request response message. If any request means we need to add in sim file or XML code.

5. After that will go to find live data, fault codes(DTC) and guided routines.

5. Result And Analysis

1.PCAN view simulator: It displays the request response of the tester and ECU.



```
;$FILEVERSION=1.1
;$STARTTIME=43403.507002963
;
;   C:\Users\uia92469\Desktop\CAPTUR\steering read dtc trace\read dtc.trc
;
;   Start time: 10/30/2018 12:10:05.056.0
;   Generated by PCAN-View v4.0.31.444
;
;   Message Number
;   |         Time Offset (ms)
;   |         |        Type
;   |         |        |        ID (hex)
;   |         |        |        |     Data Length
;   |         |        |        |     |   Data Bytes (hex) ...
;   |         |        |        |     |   |
;---+--   ----+----  --+--  ----+---  +  -+ -- -- -- -- -- -- --
     1)     10030.6  Rx        074D  8  02 10 C0 00 00 00 00 00
     2)     10050.2  Rx        076D  8  02 50 C0 00 00 00 00 00
     3)     10124.6  Rx        074D  8  02 21 3B 00 00 00 00 00
     4)     20156.2  Rx        074D  8  02 10 81 00 00 00 00 00
     5)     20172.7  Rx        076D  8  02 50 81 00 00 00 00 00
     6)     20218.5  Rx        074D  8  02 10 C0 00 00 00 00 00
     7)     20224.6  Rx        076D  8  02 50 C0 00 00 00 00 00
     8)     20232.2  Rx        074D  8  03 19 02 3B 00 00 00 00
     9)     20244.7  Rx        076D  8  07 59 02 B9 92 68 55 10
    10)     20275.2  Rx        074D  8  02 10 81 00 00 00 00 00
    11)     20283.7  Rx        076D  8  02 50 81 00 00 00 00 00
    12)     20328.2  Rx        074D  8  02 10 C0 00 00 00 00 00
    13)     20349.7  Rx        076D  8  02 50 C0 00 00 00 00 00
    14)     23359.0  Rx        074D  8  02 3E 01 00 00 00 00 00
    15)     23371.9  Rx        076D  8  01 7E 00 00 00 00 00 00
    16)     26369.9  Rx        074D  8  02 3E 01 00 00 00 00 00
    17)     26383.0  Rx        076D  8  01 7E 00 00 00 00 00 00
    18)     29380.7  Rx        074D  8  02 3E 01 00 00 00 00 00
    19)     29392.2  Rx        076D  8  01 7E 00 00 00 00 00 00
    20)     32391.5  Rx        074D  8  02 3E 01 00 00 00 00 00
    21)     32405.3  Rx        076D  8  01 7E 00 00 00 00 00 00
    22)     35401.9  Rx        074D  8  02 3E 01 00 00 00 00 00
    23)     35416.5  Rx        076D  8  01 7E 00 00 00 00 00 00
    24)     38414.5  Rx        074D  8  02 3E 01 00 00 00 00 00
    25)     38429.6  Rx        076D  8  01 7E 00 00 00 00 00 00
    26)     41423.4  Rx        074D  8  02 3E 01 00 00 00 00 00
```

Figure 6. PCAN Simulator Trace

2. Live data and DTC from OE tool(Renault):In UDS 22 and 19 02 3B service identifiers are used to read the live data and DTC.

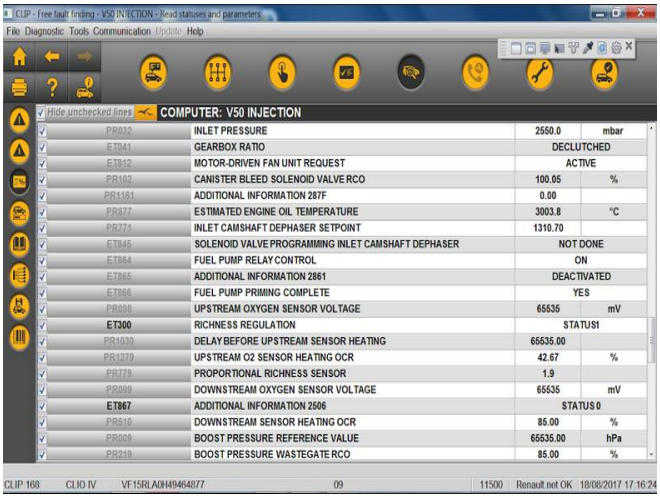


Figure 7. Reverse Engineering OE Tester Live data displa**y**

3.Tester display result: the taken reverse engineering data base has been stored in the check-II device memory or cloud. Now we can test with virtual vehicle by fast check or global scan.

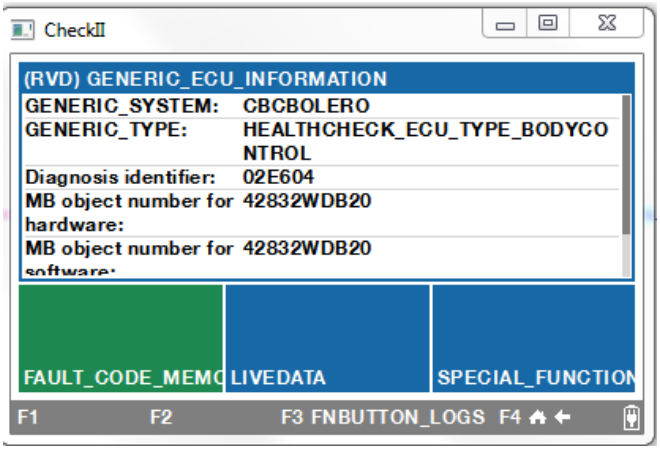


Figure 8. Multi-brand external tester(Check-II) display

6. Conclusion and Future Work

The special feature of CAN that allows different electronic units to connect with one another has made it relevant in the healthcare sector and outlines challenges, possible solutions and technological advances in the reverse engineering process for automotive mechatronic systems that communicate through CAN bus. Traditional methodologies cannot accommodate the increasing complexity and diversity of mobile devices. Also as, reverse engineering have to build up methodologies from OEMs need to develop new technologies with help of existing system, specifically focused on the vehicle inspection and diagnostics activities. Finally the multi-brand diagnostic simulator was designed and continuously implemented/integrate with new models. And issues to be corrected. Here am explaining one small part of the system diagnostics with the help of controller. This feature is not included in the majority of currently available professional programs. We also implemented a simulator of the vehicle on-board computer and its diagnostics over CAN bus.

As future works, the project team will face the following tasks:

- Implement the remaining communication protocols in the OBD-II standard to enhance the robustness of the device without compromising the project viability;
- Implement a wireless communication interface, via Bluetooth, Wi-Fi, or XBee, between the established scanner and the PC; and

Continue the research in new technologies to incorporate the current requirements of the third generation OBD standard.

References

[1] Jittiwut Suwatthikul, 'Fault detection and diagnosis for in-vehicle networks', National Electronics and Computer Technology Center (NECTEC), Thailand.

[2] Jheng-Syu Jhou ,Shi-Huang Chen ,Wu-Der Tsay and Mei-Chiao Lai (2013), 'The Implementation of OBD-II Vehicle Diagnosis System Integrated with Cloud Computation Technology', IEEE Transaction Robot, Vision and Signal Processing.

[3] Hasan N N, Arif A, Pervez U, Hassam M and Ul Husnain S S 2011 Micro-Controller Based On-Board Diagnostic (OBD) System for Non-OBD Vehicles Proc. - 2011 UKSim 13th Int. Conf. on Modelling and Simulation pp 540–544.

[4] Türker G F and Kutlu A 2016 Survey of Smartphone Applications Based on OBD-II for Intelligent Transportation Systems J. Eng. Res. Appl. www.ijera.com 6 p 2248.

[5] Bosch. "CAN Specification", Version 2.0, Robert Bosch GmbH, 1991.

[6] ISO 15765-2:2011, Road vehicles – diagnostic communication over controller area networks (DoCAN) – part 2: Transport protocol and network layer services.

[7] ISO 14229-1:2013, Road vehicles – unified diagnostic services (UDS) – part 1: Specification and requirements.

[8] ISO 14230-2:2013, Road vehicles – diagnostic communication over k-line (DoK-line) – part 2: Data link layer.

[9] ISO 15765-2 - Road vehicles – Diagnostics on CAN – Part 2: Network layer services.

[10] ISO 14230-2:1999 - Road vehicles – Diagnostics Systems - Keyword Protocol 2000 Part 2: Data link layer.

[11] ISO 9141 - Road vehicles – Diagnostic Systems – Requirements for interchange of digital information.