IOSR Journal of Electronics and Communication Engineering (IOSR-JECE)
e-ISSN: 2278-2834,p- ISSN: 2278-8735.Volume 10, Issue 1, Ver. 1 (Jan - Feb. 2015), PP 20-31
www.iosrjournals.org

Automotive Diagnostics Communication Protocols Analysis- KWP2000, CAN, and UDS

Muneeswaran. A,
ECU Diagnostics Engineering, Dept. of Electric & Electronics,
Renault Nissan Technology & Business Centre, Mahindra world city,Tamil Nadu, India

Abstract:*The increasing application of embedded electronic components in vehicles brings the need to use diagnostic systems for track and control of parameters. Development, industrial and after-sales are all fields thatuse diagnostic systems' help to execute their tasks.A diagnostic system must, therefore, contain a protocol for connecting the diagnostic tools that the designers, testers and repairers use for checking the ECU's diagnosis information. Each protocol might be suitable for only one diagnostic system and vehicle components and systems need a great amount of effort to implement protocol for one particular diagnostic system. However, there are many types of diagnostic systems defined by ISO and SAE depending on the type of systems and specific diagnostics from the vehicle manufacturers. Moreover, under some conditions, the development time may more than double. Applying communication protocols such as KWP2000, CAN and UDS makes diagnostic device of vehicle network communicate to each other according tostandards.This Paper aims to present an overview about a fewcommunication protocols for diagnostic and services, byshowing their specific tools and applications.*

Index terms:*Controller Area Network, International Standard Organization, Society of Automotive Engineers, OSI Layers, Keyword Protocol, Unified Diagnostics*

I. Introduction

Diagnostic determines, verifies and classifies symptoms aiming to get an overall picture in finding the root cause of a problem in a car. The detection, Improvement and communication strategies applied to abnormal operation of systems is monitored by Electrical and electronic devices. Therefore, the purpose ofDiagnostic is to identify the root cause of abnormal operation so a repair can be performed.Diagnostic requirements for OEM and supplier are defined by a common database which contains the functional diagnostic requirements, its implementation, development, specific dataconcerning to it and also its features. Everyone must have a straight interface with product engineering, manufacturing,Aftersales and supplier. Applicationsof diagnostic can be classified for the following fields as OEM:

Development – in this process, correct functionality of the vehicle's components must be validated. Diagnostic subsystem then takes role at reading out ECU's internal information, and data such as sensor and actuator's values.

Production – the assembly plant uses the diagnostic system for transferring calibration data and software updates to the non-volatile memory of the ECUs, EOL programming and tests included.

Aftersales – in the operating vehicle, diagnostics are mainly used for error detection. Detected errors are stored to a persistent fault memory, and trouble codes are read out at the service station in order to make troubleshootingFeasible.

In the development cycle of an automotive system, diagnostic features can be employed in development, manufacturing and services stages. First of all, diagnostic specifications must take a common database to perform - such as ODX - in which change and data management, and diagnostic development tools should be defined. Diagnostic systems require that computer tools can access a communication protocol and perform all diagnostic systems based on the diagnostic connector.

II. Automotive Protocols

A diagnostic protocol defines a set of conventions used for diagnostic communication between a diagnostic device and an ECU in the vehicle. Diagnostic communication services should be defined in a way to meet development stages, manufacturing and services requirements.The main characteristics of the following diagnostic protocols and services will be investigated. They are all well-known and broadly used by OEM and suppliers at the market:

1) KWP2000 - Keyword Protocol 2000
2) Diagnostics on CAN Protocol
3) UDS - Unified Diagnostic Services.

It will be presented a description of their characteristics and a comparative analysis will be performed. In order to assist OEMs and suppliers for an overall view, some diagnostic tools used for development, test, validation and services will be introduced.

III. Diagnostic Communication Ofvehicle

The Automobile has composed set of generic subsystems as: Power train, Chassis, Body, Communication Control and other related domains as Customers and Standards Requirements.In Automotive, the Systems are managed by Electrical and Electronics Components with Control of Software Standards of Automotive Embedded System Requirements.In Automotive Electronics, Electronic Control unit (ECU) is a generic term for an electronic device that can manage and control subsystems in the automobile. This approach is commonly defined by embedded control systems that demand method, process and tools for development.

Fig 1 presents a block diagram and Architecture of an Engine ECU used to managing and controlling automotive system. It contains analog and Digital input/output signals, Serial Communication Control unit which is enable the interconnection between other automotive System unit like Cluster, Airbag System, Switch module etc.

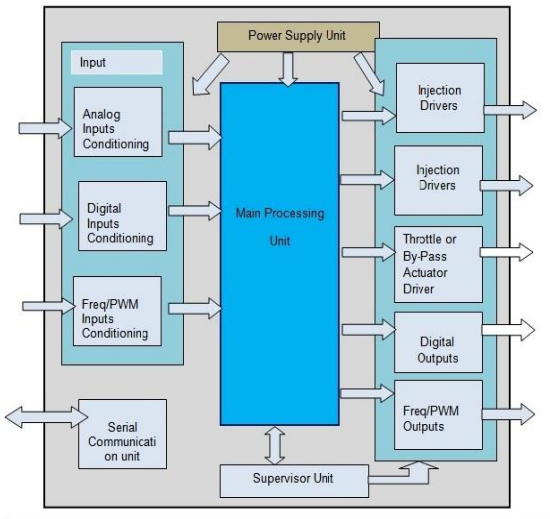


Figure 1: Block Diagram and architecture of Engine ECU with input/output interfaces

Types of ECU include Electronic/Engine Control Module (ECM), Power train Control Module (PCM), Transmission Control Module (TCM), Electronic Brake Control Module (EBCM), Suspension Control Module and other ones, allDefined and named by OEM (Original Manufacturer Equipment) according to its requirements.There are cases when an assembly component incorporates several of the individual control modules. For example: PCM is to manage Engine and Transmission.

Fig 2 presents a block diagram at ECU Managing the Combustion engine.It shows the main input and output signals. Electrical requirements need to be in accordance to the automobile Environment inside the ECU.

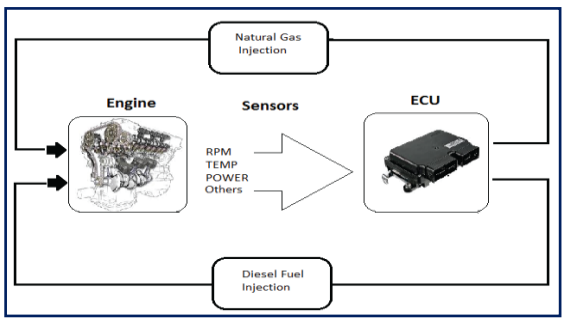


Figure 2: Engine electronic management with an ECU

In vehicle functions were realized able with electro-mechanical means or with discrete electronic components either not at all or only with disproportionately high complexity. In the mid of 1970's the development of integrated circuits in a wide range of applications and revolutionized automotive Engineering.

Until the beginning of the 1990's data were exchanged through point to point links in problems with weight, cost, complexity and reliability. Currently, some modern motor vehicles have up to 60 or more ECU's communicate via several automotive bus systems, such as LIN, CAN, UDS, MOST, Flex Ray etc. with each other via Gateway Control unit.In such automotive bus a network has managing the increasing of complexity of ECU's in vehicle has a challenge for OEM's and automotive suppliers.

Main tasks of the ECU fulfill its controlling their specific functionality in which measures and collects the data need to be perform its main tasks checking and filtering the main signals/ incoming signals. The powerful of automotive networking between the ECU's a good many new performance features can even be achieved completely without additional Hardware.

Factors influencing the design complexity of Distributed systems

- No. of parts
- No. of Nodes/Modules
- Interface Structure
- Communication System
- No. and Nature of I/O signals
- Common mode dependencies
- Power Management

In Error Controllers are used to detectthe fault and enable to perform the Diagnostics using external tester connected with OBD Connector.The Diagnostics part of control unit Embedded Software should fulfill International Standards such as CARB (California Air Resources Board), EOBD (European On-Board Diagnostic), ISO 11898[7], ISO 14229[9], ISO 15031[15], ISO 15765[2] and others.

In Fig 3 shows that a vehicle network topology structure layers and levels for supporting a high level of contents. The different automotive protocol has communicated with central gateway control unit. To reduce the network complexity automotive network topologies is used according to E/E architecture.

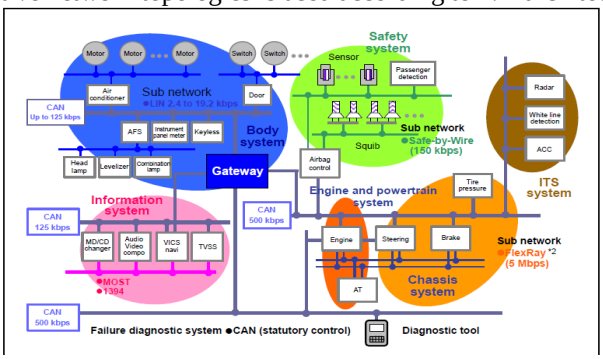


Figure 3: Vehicle network architecture for high level of content

In Fig 4 is represented the Diagnostics approach for vehicle Connected with Electronic subsystem. In the vehicle subsystem isconnected with Diagnostics tester to make the On-Board Diagnostic

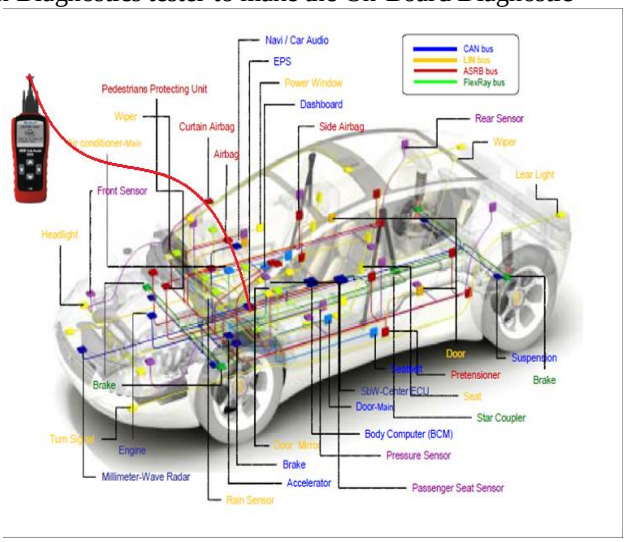


Figure 4: Diagnostic Communication system in vehicle

IV. KWP-2000 – KEYWORD PROTOCOL 2000

The Keyword Protocol 2000, commonly named as KWP2000, is a Communication protocol for On-Board vehicle Diagnostics applied for OEM. There are two primary International Standardized documents such as ISO 9141[10], ISO 14230[4] covered for performing the OBD Communications on K-line between Tester and vehicle.ISO 14230[4] specifies common requirements of diagnostic services which allow a tester to control diagnostic functions in an on-vehicle Electronic Control Unit. In Fig 5 represents the KWP 2000 protocolBus concepts.

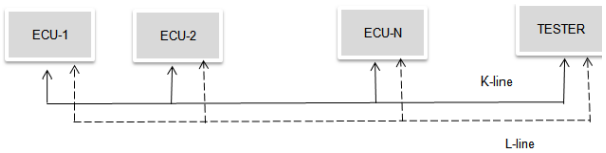


Figure 5: KWP protocol bus Topology model

ISO 9141[10] is a bidirectional serial communication details which specifies K-line is used for communication and initialization; the L-line (optional) is used for initialization only. Special cases are node-to-node-connection, this means there is only one ECU on the line which also can be a bus converter.In ISO 14230-2[5] specifies Data link layer and ISO 14230-3[6]includes all the definitions which are necessary to implement the Diagnostics services.The physical layer may be used as a multi-user bus system, there is necessary to maintain or manage arbitration and bus management system.In Fig 6 described the Vehicle Diagnostics architecture of KWP 2000 which specifies communication among Tester and ECU network with Gateway or Without Gateway access.

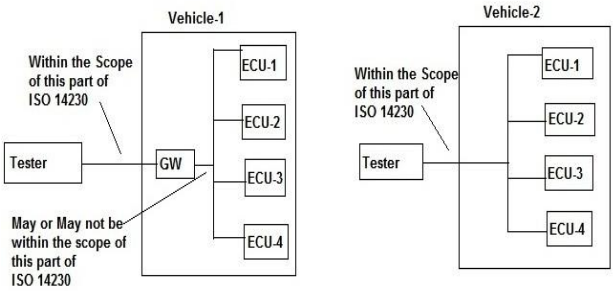


Figure 6: Vehicle Diagnostics Architecture of KWP-2000

In Keyword protocol there is no continuous communication must be initialized without being first requested by tester which means that Wake-Up at 5 baud per seconds.The 5 Baud address byte is transferred from the Tester on K-line and on L-line. After sending the 5 baud address byte the tester will maintain L-Line on High level. After Wake-up procedure, all ECU's are initialized shall use a baud rate of 10,400 Baud for initialization and Communication.[5]

At first tester transmits a Wakeup pattern on K-line and L-line synchronously. The pattern has begins after an idle time on K-line with low time of initialization value. The Tester transmits the first bit of the startcommunication service after a time of wake-up time following the first falling edge, as shown in Fig 7.

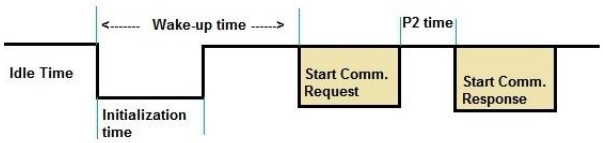


Figure 7: Fast Initialization process of K-line

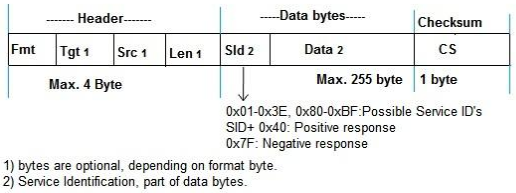


Figure 8: Message and Diagnostics Service structure

The KWP2000 over CAN (ISO 11898[7], ISO 15765[2]) used for diagnostic & continuous communication among ECU's. To achieve better communication, bus arbitration, CAN frame structure is handled by Hardware.

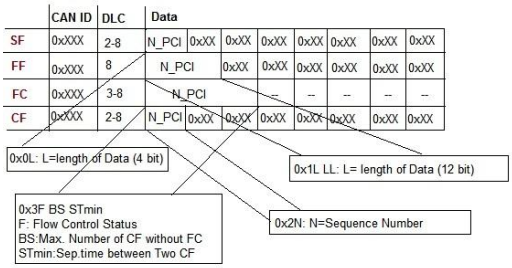


Figure 9: KWP2000 over CAN(ISO 11898, ISO 15765-2)

V. Diagnostics On Can Protocol

CAN which stands for Controller Area Network, is the serial communication protocol internationally standardized by ISO. In automotive industry has adventure to various electronic control systems that have been developed in pursuit of safety, comfort, pollution prevention and low cost.CAN was first developed by Robert Bosch GmbH, Germany in 1986 when they were requested to develop a Communication system between three ECU's in a vehicle Mercedes.In 1993 CAN protocol standardized is ISO 11898 standards "Road vehicles-Interchange of Digital Information Controller Area Network (CAN) for high speed Communication. CAN Operated according to theMulti-masterprinciple. CANtransmits signals on the CAN bus which consists of a CAN high and CAN Low.

These 2 buses are carrying signals opposite to each other to overcome noise interruption (EMI) that simultaneously on the bus. The CAN protocol is based on the two logic states: The bits are "recessive" (logic 1) or "dominant" (logic 0).

A. CAN Protocol Features:

Content based Addressing – CAN uses message-based addressing. This involves assigning fixed identifier to each message. The Identifier classifies the content of the message.

Priority Assignments – CAN uses bit arbitration technique which is the priority of accessing the bus determined by the11-bit or 29 bit identifier. Due to the architecture that "dominant" bit will always override "recessive" bit, the node with lower Identifier will have higher accessing priority [8]. This is part of the input from CSMA/CA which stands for Carrier Sense Multiple Access with Collision Avoidance.

System flexibility –The units connected to the bus have no identifying information like an address. As a result, when an ECU unit is added or removed from the bus, there is no need to change the software, Hardware or application layer of any other unit connected to the bus.

Communication speed –Any communication speed can be set that suits the size of a network. Within oneNetwork, all units must have the same message speed.

Error Confinement -There are two types of errors occurring in the CAN: a temporary error where data on the bus temporarily becomes erratic due to noise from the outside or for other reasons, and a continual error where data on the bus becomes continually erratic due to a unit's internal failure, driver failure, or disconnections.

The CAN has a function to discriminate between these types of errors. This function helps to lower the communication priority [1] of an error-prone unit in order to prevent it from hindering communication of other normal units, and if a continual data error on the bus is occurring, separate the unit that is the cause of the error from the bus.

The Maximum CAN bus data transfer rate of 1Mbps can be achieved with a bus length of up to 40 meters when using a twisted pair, which is the most common bus medium used for CAN. Other transmission mediums, such as coaxial cable or fibre optics could be used. The relation between data transfer rate and bus length, for twisted wire pair is shown in Fig 10. For bus lengths longer than 40 meters the bus speed must be reduced, and at the bus lengths above 1000 meters special drivers must be used[11].

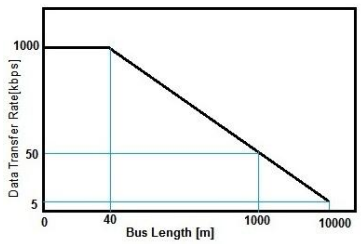


Figure 10: Relation between data transfer rate and bus length

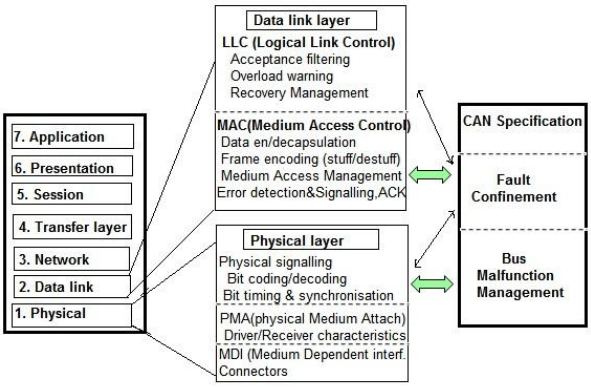


Figure 11: ISO/OSI layer CAN Protocol layers

In Fig 11 shows the CAN protocol Model based on ISO 11898[7] for physical layer, ISO 15765-2[3] specifies Transport and Network layer services.

B.CAN Message Format:

The CAN protocol supports two message formats, the only important difference being in the length of the identifier. The so called CAN standard frame as well known as CAN 2.0[1] A, supports a length of 11 bits for the Identifier, whereas the so called CAN Extended frame, also known as CAN 2.0 B, supports a length of 29 bits for the Identifier.

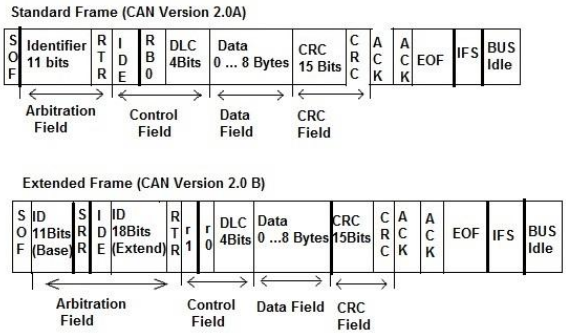


Figure 12: Standard & Extended CAN Frame Format

Start of Frame: The start of frame of Data frame consists of a single dominant bit.
Arbitration field: The field in which arbitration takes place consists of Identifier bits and the bit immediately following called RTR (Remote Transmission Request). The length of the identifier is 11 bits for standard Data frame and 29 bits for Extended Data frame. The RTR bit must be dominant.
Control field: In the 2 bits reserved bits and 4 bits of control field is the Data length code indicates the number of bytes contained in the data filed.
Data field: The Data filed is the location of the useful data to be transmitted. It can consist of anything from 0 to a maximum 8 bytes, transmitted with the MSB leading.
CRC field: CRC stands for cyclic redundancy check. It checks the validity of the SOF, arbitrationfield, control field and data field. It can detect maximum of 5errors in the message. This field ends with a recessive bit delimitation called the CRC delimiter [11].
ACK field:The first bit of the acknowledgement field is dominant if no error related to the CRC hasbeen detected by other nodes on the bus. Otherwise, an error frame is sent. The secondbit is always recessive and is the acknowledgment delimiter. This bit is followed by aflag sequence of seven recessive bits known as the end of frame.
EOF Frame: The data frame is terminated by a flag consisting of a sequence of 7 recessive bits, which it should be noted is 2 bits longer than the bit stuffing standard length [1].
IFS:This frame is used to separate the data and remote frames. Whichever frames may have been transmitted prior to them, are separated from the preceding frame by an inserted interframe space. However, no interframe spaces are inserted preceding overload and error frames.

C. CAN Bus Arbitration scheme:

Arbitration is the mechanism that handles bus access conflicts. Whenever the CAN bus is free, any unit can start to transmit a message. During the arbitration phase [1], each transmitting device transmits its identifier and compares it with the amount monitored on the bus. If multiple units started sending a message at the same time, they are arbitrated for conflict by using the arbitration field of each transmitted frame beginning with the first bit in it. The unit that output a "dominant level" successively in a greater No. of bits than any other units is allowed to send. The units that lost in this arbitration go to a receive operation beginning with the next bit.In Fig 13 shows Bus Arbitration mechanism.

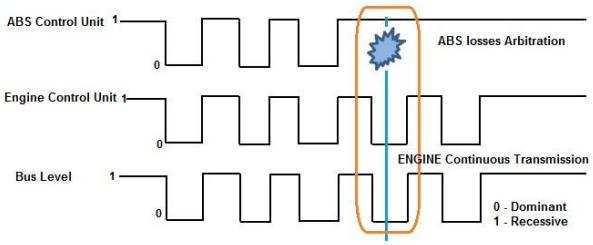


Figure 13: Bus Arbitration Scheme

D. CAN Control System Resistive Structure:

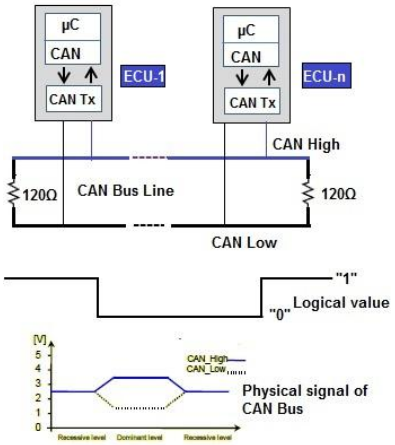


Figure 14: CAN Physical layer ISO 11898(125 kbps to 1 Mbps)

In Fig 14 shows that the structure of CAN Control system Resistive Model.The ISO-11898[7],CAN High Speed, standard assumes the network wiring technology to be close to a single line structure in order to minimise reflection effects on the bus line. It requires the use of a twisted pair cable, shielded or unshielded, with 120 ohm nominal impedance, and terminated with a 120 ohm resistor at either end to suppress signalreplications along the bus. There are not many cables in the market today that fulfil this need, but a shielded twisted pair cable with 60 ohm nominal impedance is readily available and low-cost.

So it was decided to use a 60 ohm cable for this model system rather than 120 ohm cable as required by the standard. To suitably terminate such a cable, one would expect to have 60 ohm resistors connected at its ends. This is true if CAN transceiver IC was designed for such a setup. Unfortunately, the transceiver is designed to drive a minimum of 45 ohm bus load, which cannot be reached using 60 ohm resistors ateither end, giving a total of 30 ohm bus load. This would increase the load to 60 ohms so that the transceivers can function properly but would generate reflected signals at the ends of the cable. However, for a 20 metre long cable with 125kbps data transfer rate, the signal reflections are very small and would not cause major Communication problems.

CAN High and CAN Low (CAN_H and CAN_L), represent the differential transmission system for the CAN bus. In figure 14 shows the differential voltage among CAN_H and CAN_L is what is used by the CAN protocol. Initially, when there isno data being linked on the bus, both CAN_H and CAN_L are set to 2.5Vdc.[8] For example, logic 1 is transmitted along the bus, CAN_H voltage increases by 2.5V and CAN_L voltage drops by 2.5V, which gives a difference of 5V that is logic 1.

The Advantage of such a technique is that if voltage spike was induced in the cable both conductors would be affected equally and the voltage difference among the two conductors would be maintained, thus providing the CAN Network with a good degree of noise immunity.

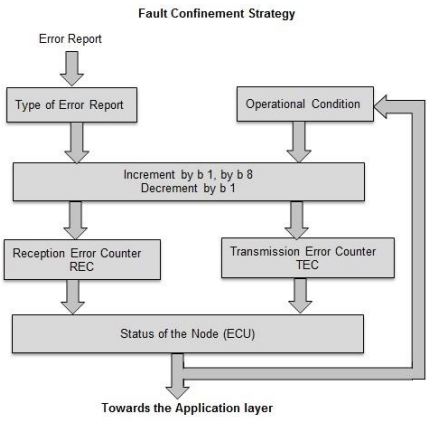
E.CAN Bus Error Handling:

The CAN protocol is designed for safe data transfers and provides mechanisms for error detection, signalling and self-diagnostics, includes the fault confinement [13].
Error detection is done in five distinct manners in CAN: [1]
(i) Bit monitoring, (ii) bit stuffing, (iii) Frame check, (iv) ACK check and (v) CRC check.
Cyclical redundancy errors are detected using a 15 bit CRC computed by the transmitter from the messagecontent. Each receiver accepting the message recalculates the CRC and compares it against the transmitted value. A discrepancy between the two calculations causes an error flag to be set. Frame checks that will flag an error are the detection by a receiver of an invalid bit in the CRC delimiter, ACK delimiter, EOF or 3-bit interframe space. Finally, each receiving node writes a dominant bit into the ACK slot of the message framethat is read by the transmitting node. If a message is not acknowledged (perhaps because the receiver has failed), an ACK error is flagged.
In the bit Level Error, each transmitted bit is "Read back" by the transmitter. If the monitored value is different than the value being sent, a bit Error is detected. In More bit errors are detected by stuffing: After five consecutive identical bits have been transmitted, a bit of the oppositepolarity will be inserted ("stuffed") by the transmitter into the bit stream. Receivers automatically "de-stuff" the message. If any node detects six consecutive bits of the same level, a stuff error is flagged.

F. CAN Fault Confinement:

The Error detection, signaling and fault confinement defined in the CAN standard makes the CAN bus very reliable. The purpose of this mechanism is not only to enable hardware faults and disturbances to be detected, but also especially to locate them so that accurate interference is possible. The rules for processing confinement errors are described in CAN protocol in such way of that microcontroller closest to the place of occurrence of the error react with highest priority and as quickly as possible.

All microcontrollers confirming to the CAN protocol have two separate internal counters, "Transmit Error Counter"(TEC),"Receiver Error Counter" (REC)[11] with the task of recording the errors occurring in transmission and reception respectively. The CAN protocol strategy of Fault Confinement outline shown in the Fig15, Fig 16.



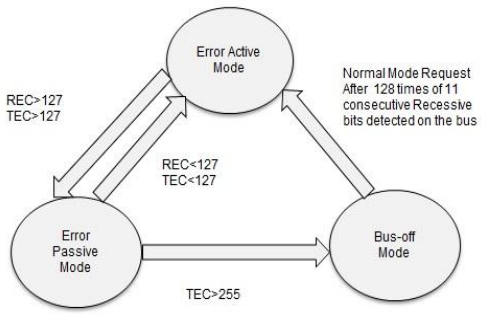


Figure 15: Fault Confinement Strategy of CANFigure16: CAN Error State Machine

In Respect to the Fault Confinement mechanism, the ECU Unit may be in one of the three following status,Error Active, Error Passive and Bus-off [11].

In Error Active mode is working normally. It can takepart in communication and send error flags when it detects an error frame, thereby terminating the transmitted frame. In this state the counts of both the error counter would be less than 127. When the error count reaches zero, these node will return to normal mode form error active mode.

The Error passive state is a state in which the unit tends to cause an error. While the unit in an error passive state can participate in communication on the bus, it can't notify other units of an error positively during a receive operation in order not to delay their communication. Even when the unit in an error passive state has noticed an error, if no errors are detected by other units in an error active state, it is assumed that no errors have arisen anywhere in the bus. Once unit in an error passive state has detected an error, it transmits passive-error flag. Furthermore, the unit in an error passive state can't start transmission immediately after it has finished sending. A suspend transmission period (comprised of 8 recessive bits) inserted in an interframe space before the next transmission can start.

In the Bus-off mode, the Error counter of a node reaches 255, and then the node/unit cannot participate in the communication bus. When in this state, the unit is disabled from all transmit/receive operations.When the

error counts reach 128 or below then the node come back to error passive mode and take part in communication again. Otherwise by resetting the error counters and the CAN controller, they can take part in communication after 128 occurrences of 11 consecutive bits.
The values of the Transmit and Receive Error Counters change according to the prescribed conditions.

In the Table-1 [1] summarizes the conditions under which the error counter values change. It is possible that many conditions will overlap in one data transmit/receive operation. The time at which the error counter counts up coincides with the first bit position of the error flag.

Table1. Error Counter Conditions

S.No	TEC/REC change conditions	TEC	REC
1	When the receive unit has detected an error Except in the case where the receive unit noticed a bit error while it was sending an active-error flag or overload flag.	-	+1
2	When the receive unit has detected a dominant level in the first bit that it received after sending an error flag.	-	+8
3	When the transmit unit has transmitted an error flag	+8	-
4	When the transmit unit has noticed a bit error while sending an active-error flag or overload flag	+8	-
5	When the receive unit has noticed a bit error while sending an active-error flag or overload flag	-	+8
6	When any unit has detected the dominant level in 14 consecutive bits from the beginning of an active-error or an overload flag, and each time the unit hasnoticed a dominant level in 8 consecutive bits later	For a transmit unit+8	For a receive unit+8
7	When any unit has detected a dominant level in additional 8 consecutive bits after a passive-error flag, and each time the unit has detected a dominant level in 8 consecutive bits later.	For a transmit unit+8	For a receive unit+8
8	When the transmit unit has transmitted a message normally (ACK returned and no errors noticed until completion of EOF).	-1 ±0 when TEC = 0	-
9	When the receive unit has received a message normally (No errors detected until ACK slot and the unit was able to return ACK normally).	-	–1 when $1 \leq REC \leq 127$ ±0 when REC = 0,When REC > 127, a value between 119 to 127 is set in REC
10	When the unit in a bus-off state has detected a recessive level in 11 consecutive bits 128 times.	Cleared to TEC = 0	Cleared to REC = 0

There are various tools available to check the CAN protocol software layer components. In most recent years all OEM's are used the Vector's software tools for evaluating ECU Diagnostics functions. In Table-2 summarizes tools availability of CAN protocol [13].

Table2. Tools for CAN Protocol

Tools	Purpose
Network Designer CAN	Tools for Designing the CAN Network Architecture
PREEvision	Design Process of Distributed Automotive Systems
CANalyzer	Analysing Tool of ECUs and Networks
CANoe	Tools for Development, test, simulation, Diagnostics and Analysis of ECUs and Networks.
CAN Scope	Integrated Oscilloscope solution for CANoe and CANalyzer
CANape	Measurement & Calibration via CCP/XCP and

	Diagnostics
CANDelaStudio	The Software Environment for Improving the Diagnostics Development Process of Automotive ECU's

VI. UDS – Unified Diagnostics Service: ISO 14229-1

The ISO 14229-1[9]Unified Diagnostic Services (UDS) Protocol is a communication protocol in the automotive field, developed based on the idea of Keyword Protocol (KWP2000) to fulfill common requirements for diagnostic systems on CAN buses. In UDS Protocol was created by combining the ISO Standards 14230-3 (KWP 2000) and 15765-3 (Diagnostics on CAN). This carried out to greatly reduce the costs which to date have arisen for the development of diagnostic communication. It has been the various protocol landscapes with vehicle manufacturers and suppliers.The standard provides a unified set of diagnostic services for ECUs. The diagnostic services for CAN have been adapted on this new basis. UDS also reduces the effort involved in creating test applications. In the ideal case Test scenario, the diagnostic tool can communicate with other ECUs and vehicles without any changes.There are five types of Diagnostics functions described in the specification.

Table 3. UDS Diagnostics Functions

Diagnostics functions	**Examples**
Communication Management	Session Control, Device Reset, Security Access, Communication Control
Data	Read Identifiers or Memory Write Identifiers or Memory
Stored Data	Read Diagnostics Information Clear Diagnostics Information
I/O Control	Control Input or Output
Reprogramming	Download and Upload of Data

UDS protocol is only an application layer in the OSI model. Normally, the intercommunication inside various vehicles uses Controller Area Network (CAN) buses as a Physical Layer of the protocol. There are a number of standards used for different kinds of diagnostic services as mentioned in the Diagnostic Service section. In the ISO-15765[3] is one of the principal ideas for developing a communication protocol of enhanced diagnostics on CAN buses. The standard defines a specification of the Network Layer and Transport Layer for Diagnostic Services. Moreover, this standard represents the latest protocol and is a mandatory standard for modern vehicles.In ISO 14229-1[9] state that UDS is Protocol- independent specification. To implement the message transfer rules the Network layer service specification is required. In Table 4 shows the required specification of OSI 7-layer Model of UDS protocol.

Table 4. OSI Layer specification of UDS

OSI Layers	**Enhanced Diagnostics on CAN (UDS)**
Application Layer	ISO 14229-1 & ISO 15765-3
Presentation Layer	N/A
Session Layer	ISO 15765-3
Transport Layer	ISO 15765-2
Network Layer	ISO 15765-2
Data Link Layer	ISO 11898/SAE J1939 - 15
Physical Layer	ISO 11898/SAE J1939 - 15

In the Fig 17 shows the other possibilities of ISO 14229-1 protocol specification.

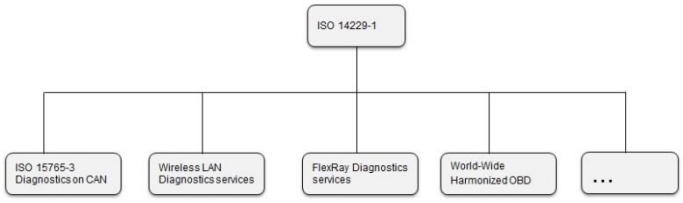


Figure 17: Available Standards & possible future Implementation of ISO 14229-1 UDS[9]

VII. ODX – Open Diagnostics Data Exchange

ODX is the abbreviation for Open Diagnostic Data Exchange and it's standardized in ISO 22901[12].The ODX data model of ECU diagnostics and programming data is specified in UML ("**U**nified **M**odelling **L**anguage"). The implementation format is defined in XML. This language device the diagnostic interface of an electronic control unit is described in electronic form. This format includes detailed specifications of diagnostic protocol complete communication parameters and detailed service requestsresponses with scaling formulas and physical units.

The purpose of ISO 22901 is to define the data format for transferring Electronic Control Unit (ECU) diagnostics and programming data between the system supplier, Vehicle manufacturer and service dealerships and diagnostic tools of different vendors.

An ODX user is to have a standardized diagnostic description format which can be used OEM, Suppliers, production, After-Market for ECU Development. In that cause ODX form is the advantage and benefits in the development of automotive electronic components.

In Fig 18 shows the vehicle development process using ODX standard will be clarified.

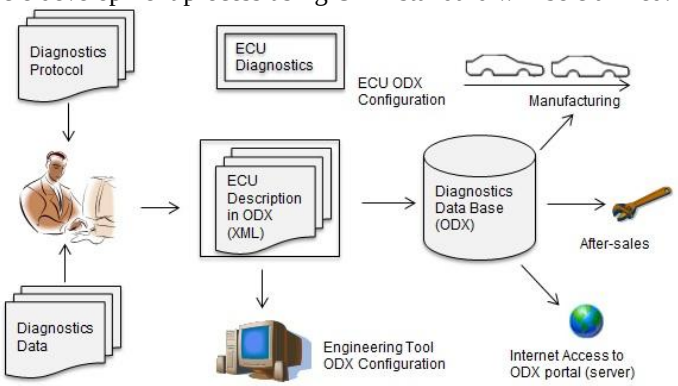


Figure 18: ODX usage of ECU & Vehicle Life Cycle(Reference: ISO 22901)

A. Benefits of ODX in Automotive Development Process

Reduce effort for Implementation tests and ECU tests of diagnostics features (Time/Efficiency). Increase the test complexity, reproducibility of tests and to get the better quality. Better support by specialized diagnostics tools[14].There is possible use of ODX for configuration of software components. The generation of documentation is possible from XML Description. In the vehicle manufacturing side the re-utilization of verified data in the development, End Of Line (EOL) and After-sales service Tools. The Loss cost in the diagnostics data distribution.

VIII. Comparison Analysis

In the Table-5 shows key structures of diagnostics protocols for Automotive Standards at the ISO Organization. In ISO 14230[3], ISO 11898[7] are the diagnostics communication protocols with the highly used for global Automotive market and next generation of Automotive/Vehicle Communication Applications. The New Trend of automotive diagnostic communication is used the UDS protocol and its standards 14229-1[9]for passenger car and commercial vehicles being possible to run over several Networks technologies according to Standards as K-Line, CAN Bus, LIN Bus, FlexRay, Ethernet and MOST.

Table 5. Comparative Analysis of diagnostics protocols

	KWP	**CAN**	**UDS**
Standards	ISO	ISO	ISO
Application	ISO 14230-3	ISO 15765-3	ISO 14229-1
Presentation	N/A	N/A	N/A
Session layer	N/A	N/A	ISO 15765-3
Transport	N/A	N/A	ISO 15765-2
Network layer	N/A	ISO 15765-2	ISO 15765-2
Data link layer	ISO 14230-2	ISO 11898	ISO 11898
Physical layer	ISO 14230-1	ISO 11898	ISO 11898

IX. Conclusion

The implementation of embedded electronic circuits theusage of communication protocols and standards has becomeessential to ensure the proper working and monitoring ofautomotive systems. This work provides the fundamental concepts of diagnostics communication protocol and its standards in the automotive field. This work describes the automotive standards ISO 14230-3, CAN 2.0 Specification, ISO 11898, and ISO 14229-1 (UDS) needed to fulfil the OSI layers in order to extent the related understanding of communication system. This work has provides the better understanding on the service of automotive diagnostics standards, ECU Communication with Tester Tool.

References

[1]. Bosch. "CAN Specification", Version 2.0, Robert Bosch GmbH, 1991.
[2]. ISO 15765-3 - Road vehicles – Diagnostics on CAN – Part 3: Implementation of diagnostic services.
[3]. ISO 15765-2 - Road vehicles – Diagnostics on CAN – Part 2: Network layer services.
[4]. ISO 14230-1- Road vehicle – Diagnostics Systems – Keyword Protocol 2000Part 1: Physical layer, 1999
[5]. ISO 14230-2 - Road vehicles – Diagnostics Systems - Keyword Protocol 2000 Part 2: Data link layer, 1999
[6]. ISO 14230-3- Road vehicles – Diagnostics Systems - Keyword Protocol 2000Part 3: Application layer, 1999
[7]. ISO 11898-1-Road vehicles – Controller Area Network (CAN) – Part 1: Data link layer and physical signalling, 2003.
[8]. CAN in Automation. "Controller Area Network (CAN)" [Website], http://www.can-cia.org/index.php?id=systemdesign-can
[9]. ISO 14229-1- Road vehicles – Unified diagnostic services (UDS) – Part 1: Specification and requirements, 2005
[10]. ISO 9141 - Road vehicles – Diagnostic Systems – Requirements for interchange of digital information.
[11]. Multiplexed Networks for Embedded Systems: CAN, LIN, FlexRay, and Safe-by-Wire by Dominique Paret. 2007.
[12]. ISO 22901-1 Road vehicles – Open diagnostic data exchange (ODX) – Part 1:Data model specification, 2008.
[13]. Vector: "Introduction to CAN" [Website], http://elearning.vector.com/index.php?&wbt_ls_seite_id=489557&root=378422&seite=vl_can_introduction_en
[14]. ODX Process from the Perspective of anAutomotive Supplier, Dietmar Natterer, Thomas Ströbele, Dr.-Ing. Franz Krauss and ZF Friedrichshafen AG
[15]. ISO 15031 -6 Road vehicles – Communication between vehicles and external equipment for emissions-related diagnostics – Part 6: Diagnostic trouble code definitions.