ANALYSIS FOR UDS AND KWP 2000 COMMUNICATION PROTOCOLS WITH UDS BASED DIAGNOSIS DESIGN ON AUTOSAR STANDARDS

MAHİR UMUT DİLEK

FEBRUARY 2024

ÇANKAYA UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

M.Sc. Thesis in

ELECTRICAL AND ELECTRONICS ENGINEERING

ANALYSIS FOR UDS AND KWP 2000 COMMUNICATION PROTOCOLS WITH UDS BASED DIAGNOSIS DESIGN ON AUTOSAR STANDARDS

MAHİR UMUT DİLEK

FEBRUARY 2024

ABSTRACT

ANALYSIS FOR UDS AND KWP 2000 COMMUNICATION PROTOCOLS WITH UDS BASED DIAGNOSIS DESIGN ON AUTOSAR STANDARDS

DİLEK, MAHİR UMUT

M.Sc. in Electrical and Electronics Engineering

Supervisor: Assoc. Prof. Dr. BARBAROS PREVEZE

February 2024, 66 pages

This thesis shows detailed and reflective analysis about unified diagnostic services and keyword 2000. Also not only normal analysis and also check their performances based on AUTOSAR. Thesis firstly starts with introducing critical information's about communication theory, indispensable automotive regulations, aim of the diagnostic systems, controller area network and its features. After, thesis mentions about diagnostic communication details such as diagnostic trouble codes and its format, prioritization of controller area network messages. Also, their robustness, response time periods, way of error memory management considered for these two common diagnostic protocol. Additionally, AUTOSAR introduced detailly with describing its each layer and mentioning about consortium promoter companies and mentioned about their necessity by showing how important to develop software independent from by hardware which means electronic control unit. Analysis continuous with comparing unified diagnostic services and keyword 2000 protocols. This comparison not only emphasize of one of the protocols but also test them with identical conditions. This test includes, their response time period for each transmit and received controller area network messages and with using same electronic control unit a virtual diagnostic trouble code created and measured that how these two common protocol managed error memory. Results of this thesis shows unified diagnostic services and keyword 2000 protocols strong, robust and weaknesses, and mentions how to build unified diagnostic services based on AUTOSAR with remarking critical points. All in all, this thesis results supports researchers, engineers and who will build automotive diagnostic architecture and helps to make conscious decisions.

Keywords: UDS, KWP 2000, Diagnostics, AUTOSAR

ÖZET

AUTOSAR STANDARTINDA UDS TABANLI TEŞHİS TASARIMI İLE UDS VE KWP 2000 HABERLEŞME PROTOKOLLERİNİN ANALİZİ

DİLEK, MAHİR UMUT
Elektrik-Elektronik Mühendisliği Yüksek Lisans

Danışman: Doç. Dr. Barbaros PREVEZE
Şubat 2024, 66 sayfa

Bu çalışma, otomotiv teşhis alanında UDS ve KWP 2000 iletişim protokollerinin kapsamlı bir analizini sunmaktadır. Araştırma, UDS ve KWP 2000'nin performansını, yeteneklerini ve AUTOSAR çerçevesindeki uygunluğunu değerlendirmeyi amaçlamaktadır. Tez öncelikle iletişim teorisi, vazgeçilmez otomotiv düzenlemeleri, teşhis sistemlerinin amacı, denetleyici alan ağı ve özellikleri hakkında kritik bilgilerin tanıtılmasıyla başlamaktadır. Daha sonra tezde, arıza teşhis kodları ve formatı, kontrolör alan ağı mesajlarının önceliklendirilmesi gibi teşhis iletişim detaylarından bahsedilmektedir. Ayrıca bu iki ortak teşhis protokolü için sağlamlıkları, yanıt süreleri, hata belleği yönetiminin şekli de dikkate alınmıştır. Ayrıca AUTOSAR, her katmanını detaylı bir şekilde anlatarak konsorsiyum destekçisi firmalardan bahsederek, donanımdan yani elektronik kontrol ünitesinden bağımsız yazılım geliştirmenin ne kadar önemli olduğunu göstererek bunların gerekliliğinden bahsedilir. UDS ve KWP 2000'nin teşhis işlevselliği, verimlilik ve esnekliğini değerlendirmek için karşılaştırmalı bir analiz yapılır. Gerçek dünya senaryoları ve kullanım durumları, hata tespiti ve çözümünde etkinliklerini değerlendirmek için dikkate alınır. Bu çalışmanın bulguları, otomotiv profesyonellerine ve araştırmacılara, AUTOSAR ortamında protokol seçimine ilişkin bilgiler sunar. Sonuçlar, belirli sistem gereksinimlerine dayalı bilinçli karar verme sürecine yardımcı olan UDS ve KWP 2000'nin güçlü ve zayıf yönlerini vurgular. Sonuç olarak, bu araştırma, AUTOSAR çerçevesi içinde UDS ve KWP 2000 iletişim protokollerinin anlaşılmasına katkıda bulunur. Bulgular, modern otomotiv sistemleri için verimli ve güvenilir teşhis çözümlerinin tasarımını bilgilendirirken, otomotiv teşhisinde gelecekteki ilerlemelere yol gösterir.

Anahtar Kelimeler: UDS, KWP 2000, Teşhis, Autosar.

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to my parents for their support and sacrifice to me. Your memories would ever shine in my mind.

Special thanks to my supervisor Assoc. Prof Dr. Barbaros Preveze for the excellent guidance and providing me with an excellent atmosphere to conduct this research. My special gratitude also goes to the rest of the thesis committee members for the encouragement and insightful comments.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS AND ABBREVIATIONS

ABBREVIATIONS

ACK	:Acknowledge
AUTOSAR	:Automotive open system architecture
CAN	:Controller area network
CANH	:CAN high
CANL	:CAN low
CID	:Common identifier
CRC	:Cyclic redundancy check
CDA	:Central diagnostic access
DCM	:Diagnostic communication manager
DEM	:Diagnostic event manager
DID	:Diagnostic identifier
DRE	:Diagnostic runtime environment
DSD	:Diagnostic service manager
DTC	:Diagnostic trouble code
ECU	:Electronic control unit
ECAL	:ECU abstraction layer
EOF	:End of frame
ID	:Identifier
ISO	:International standards organization
KWP	:Keyword protocol
LID	:Local identifier
MCAL	:Microcontroller abstraction layer
MSB	:Most significant bit
OBD	:Onboard diagnostic
RTR	:Remote transmission request
SOF	:Start of frame
UDS	:Unified diagnostic services

CHAPTER I
INTRODUCTION

1.1 INTRODUCTION

Vehicle diagnostics is of vital importance in the automotive industry. It is a system that protects both the driver and the manufacturer. Thanks to vehicle diagnostics, you can specifically detect faults and potential faults without dis- mantling any part of the vehicle, minimizing the labor cost. In addition, you can detect problems that may arise during the journey and take precautions during routine vehicle diagnostic checks. Diagnostic communication is the exchange of information between a vehicle or mobile machine and outside test equipment (tester). In reality, communication occurs not between the tester and the vehicle or machine, but rather, at a specific point in time, between the tester and one specifically selected electronic control unit (ECU). Automobiles/vehicles are similar to a human body in that the Controller Area Network (CAN) serves as the nervous system, allowing communication. In turn, 'nodes' or 'electronic control units are like body parts that are linked together via the CAN bus. One part's sensed information can be shared with another. The CAN bus system allows each ECU to communicate with all other ECUs without the need for complicated dedicated wiring. Specifically, an ECU can prepare and broadcast data (such as sensor data) over the CAN bus (consisting of two wires, CAN low and CAN high). All other ECUs on the CAN network accept the broadcasted data, and each ECU can then check the data and decide whether to receive or ignore it. In recent years modern automotive E/E (Electrical and Electronics) systems have reached a high level of complexity, leading to a corresponding increase in the complexity of the deployed software [1]. Because of that, CAN communication gained great importance to avoid this complexity. Catching potential errors/faults is vital when developing automotive E/E based on CAN communications. There are two common diagnostic protocols exist in automotive industry. They are, Unified diagnostic services (UDS) and KWP 2000. Both of them standardized from by international standards Organization (ISO). However, it is very critical to comply with the AUTOSAR standard when making any kind of development related to automotive electrical electronics.

Aim is that comparative analysis of UDS and KWP 2000 communication protocols, which are very common in the automotive industry, and UDS-based diagnostic design for a vehicle, taking into account AUTOSAR standards.

1.1.1 Purpose of Thesis

In automotive industry, diagnostic protocols are crucially important for vehicles to provide proper functionality and provide stable fault detection mechanism. Also not only provide diagnostics communication but also these protocols are providing flashing and manipulating software inside of the electronic control unit. Most common protocols in use are unified diagnostic services and keyword 2000. Unified diagnostic services much more improved than keyword 2000. This thesis describes their features, advantages, weaknesses and how to integrate AUTOSAR basic software modules.

First of all, remarking point is that unified diagnostic services developed after keyword 2000. Which means unified diagnostic services involves keyword 2000 protocols functionality and more. Unified diagnostic services standardized based on ISO 14229. Unified diagnostic services define messages to provide communication between electronic control unit and detect faults to repair. Unified diagnostic services also support remote diagnostic communications and features, and this provides remote Access to repair and update the software remotely. Additionally, unified diagnostic services provide flash programming and calibration which is not possible with keyword 2000. In contrast, keyword 2000 protocol standardized based on ISO 9141-2 and mostly preferred from by older architecture vehicles. Therefore keyword 2000 protocol provides physical connection and not possible to provide complex remote features.

Unified diagnostic protocol supports and available to use with many physical communications protocols such as; controller area network, LIN, FlexRay. This versatility makes possible to use unified diagnostic protocol for passenger cars, trucks, buses, agricultural vehicles, construction vehicles. AUTOSAR software architecture supports a lot of diagnostic communication protocols including unified diagnostic services and keyword 2000 and provides standardized software architecture.

The software architecture, which is provided from by AUTOSAR, makes possible to develop unified diagnostic services functions including processing diagnostic messages. Unified diagnostic services based diagnostic services are developed in AUTOSAR software architecture with defining service and parameters of service and services responses. The diagnostic communication module, which is essential part of AUTOSAR, has vital role to process diagnostic services including unified diagnostic services and uses for starting diagnostic services and process diagnostic services feedback. Due to unified diagnostic services has complex, common diagnostic ability and conformality with a lot of physical communication protocols, it is getting more and more popular in automotive industry. Unified diagnostic services provide standardized communication between electronic control unit and diagnostic tool. Which means, makes easy to develop compatible diagnostic communication mechanism for more than one vehicle type for automotive manufacturers. The diagnostic communication protocols which are standardized like unified diagnostic services, has vital importance future of automotive software. Standardized protocols are increases efficiency of diagnostic tools, reduces software development effort, and makes possible to integrate various software variants to one system. Also, manufacturers are uses diagnostic tools and systems for many of the vehicle types. Therefore, this standardization reduces development and integration costs for industry.

All in all, unified diagnostic services, and keyword 2000 are most common and standardized two diagnostic protocols in industry. Unified diagnostic services provide complex and improved diagnostic abilities and able to use with many physical communications protocols and this makes unified diagnostic services versatile protocol. However, keyword 2000 protocol only provides K-line and controller area network communication as physical connection and that makes it limited. AUTOSAR supports standardized approaches including unified diagnostic services and keyword 2000 and makes possible to improve unified diagnostic services systems. Unified diagnostic services are crucially important to reduce development, integration effort and also reduce the costs.

1.1.2 Literature Review

Specialist, engineers, and researchers are makes research and developments for automotive diagnostic protocols. Root cause of this improvements aim is to reduce automotive software and electronics architecture complexity. This complexity effected development effort, integration effort, fault detection time and costs edgeways. Improvement and research efforts, mostly focused to reduce complexity and making standardization of the systems. With standardizing automotive software architecture including diagnostic services, specialists are proving that complexity getting reduce and fault detection time is getting reduce. Therefore, this effects end users as positively. Especially, emergence of unified diagnostic services and keyword 2000 protocols remarks vital milestones. Commencement of this diagnostic protocol improvements and works may vary. However, mostly can be attributed to the evolutionary progress of automotive diagnostics as well as advances in electronic control systems in the second half of the 20th century.

1.1.3 Flash Programming and Central Diagnostic Access

The most critical key features of modern vehicles systems are flash programming and central diagnostic access. They effect efficiency of diagnostics and fault repairing. Flash programming means that updating of complete changing software of hardware. This hardware can be an electronic control unit or any hardware which can be diagnosable. This feature can be necessary to fix the bug or update the software or install the new software to activate of deactivate the functionality. Flash programming mostly provides connection via OBD connection socket. For sure this socket needs special diagnostic tools, and this tool includes manufacturer specific interface. This diagnostic tool can be using unified diagnostic service for keyword 2000 protocol. Also, flash programming has importance to fix the bugs or solving the problems without changing hardware or electronic control units. This also reduces technicians' effort and costs as well. Unified diagnostic services and keyword 2000 are most common protocols in automotive industry. Both diagnostic protocols are supports flash programming feature for electronic control units. However, there are differences between unified diagnostic services and keyword 200 about how to implement this flash programming feature to electronic control unit. Unified diagnostic services, provides a lot of standardized message structures for flash programming and these messages can be used many variants of electronic control units which means available for many manufacturers. The messages which are using in unified diagnostic services are standardized and they are not flexible. This approach can make the unified diagnostic services limited for specific flash programming variants.

On the other hand, keyword 2000 provides message structure which increase control of flash programming and this message structure is more detailed. In this way, flash programming can be arranged or modified sensitively, and this can be efficient especially for specific programming style and sensors. However, this complexity inside of the keyword 2000 message structure increases the effort for technicians and engineers and makes difficult to use comparing unified diagnostic services.

The central diagnostic access services are also important feature for both unified diagnostic services and keyword 2000 protocols. However, this service has important differences between unified diagnostic services and keyword 2000. In unified diagnostic services, central diagnostic Access services provides with an array of standardized diagnostic messages. These messages, provides to reach fault diagnostic data's inside the electronic control unit and makes possible to manage diagnostic process and check it for technicians and engineers. Unified diagnostic services messages designed to compatible with many variants of electronic control units and also provides flexibility to Access and checking it. Additionality, unified diagnostic services has standardized fault report mechanism system which makes possible to not only analyze the problem but also revolve the problem from by technicians and engineers.

In contrast, keyword 2000 protocol has limited content for central diagnostic Access. It is mostly focused on to reach basic diagnostic Access. Keyword 2000 diagnostic messages has less details about central diagnostic access with comparing unified diagnostic services. Furthermore, keyword 2000 does not supports standardized fault fixing mechanism during diagnostic process for technicians and engineers. One of the differences between these two protocols is, how to regulate these diagnostic data's and how to reach. In unified diagnostic services, diagnostic data are organized based on diagnostic identifiers (DID) and provides standardized approach for access diagnostic data's. However, for keyword 2000 there is no structured data organization and access, and it makes difficult for technicians and engineers.

All in all, central diagnostic access and flash programming features are available for both unified diagnostic services and keyword 2000. Due to unified diagnostic services more standardized and less complex it makes it more user friendly and time saver protocol. In first look, keyword 2000 has some advantages comparing unified diagnostic services, but at the end of the day it increases the complexity, and this advantage returns to disadvantages.

CHAPTER II
TECHNICAL FOUNDATION

2.1 ELECTRONIC CONTROL UNIT

In this chapter touches the role of complexity and versatility of the electronic control units in automotive industry. Modern vehicles include more than seventy electronic control unit and these electronic control units are works like a central nervous system. This electronic control units have separate and critical roles such as, powertrain control, body function control, engine control, transmission control, advanced driving assistance control and so on. This thesis covers responsibility of electronic control units and their requirements based on automotive standards. Also, mentioning role of the electronic control unit for optimizing of vehicle performance and improving vehicle technology. Electronic control unit consist of some sections and this section can be change based on ECU manufacturers, these sections shown in figure 1. However, if we generalized these sections of electronic control unit mostly these sections can be found in automotive specific electronic control units and they are, analog inputs, digital inputs, controller area network and LIN communication ports, microcontroller itself, analog output, digital output, and pulse width modulation outputs. Also, automotive specific electronic control units are developed to withstand for difficult physical conditions to provide optimal performance.

Furthermore, for automotive specific electronic control units' safety is major topic therefore automotive safety levels (ASIL) integrated for automotive industry. Automotive Safety Integrity Level (ASIL), specified under the ISO 26262 is a risk classification scheme for defining the safety requirements [2]. First of all, it is crucial that to define the ASIL and importance for functional safety. ASIL is an integrated classification safety mechanism as part of ISO 26262. ASIL defines the risks for not only electronic control units but also functions to reach tenable risk level. It defines the potential risks.

An Automotive Safety Integrity Level, according to the ISO 26262 standard, is defined as one of four levels to specify an item or element's necessary ISO 26262 requirements and safety measures as they apply to avoid an unreasonable risk [3]. Starting from ASIL A through ASIL D, each level safety integrity getting increase.

ASIL A (Lowest Safety Integrity): This is the lowest safety integrity level and ASIL A describes small injuries and related with damages which is only includes vehicles.

ASIL B: This is the second safety integrity level and ASIL B describes and defines small injuries but increased severity comparing ASIL A, and gives limited flexibility to apply safety protocols.

ASIL C: This safety integrity level describes outstanding safety integrity with comparing ASIL B and describes and covers serious injuries.

ASIL D (Highest Safety Integrity): This safety integrity level is located top of all ASIL levels, and it describes potential irremediable and mortal injuries and, in this level, there is no flexibility for applying safety protocols.

Defining of an ASIL level for an electronic control units or function is needs hypercritical analysis based on use case scenarios. This classification analysis assists to develop systems which has critical role for safety and also helps to reduce potential risks and apply safety protocols.

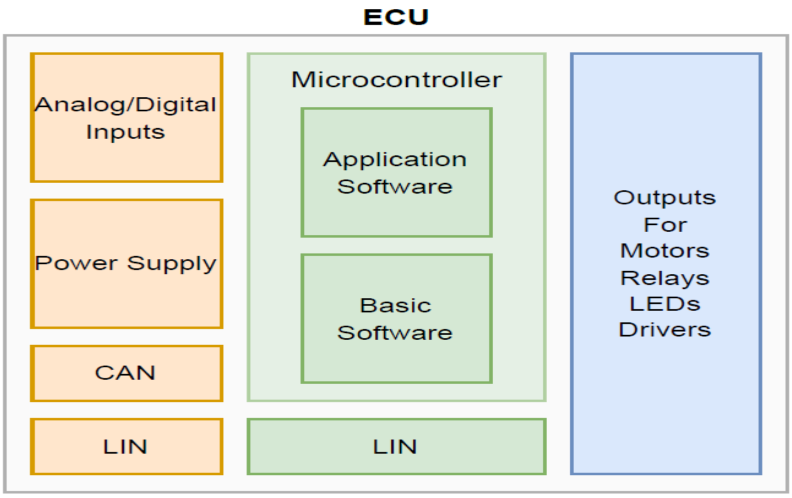

Figure 1: Electronic Control Unit Diagram

2.2 CONTROLLER AREA NETWORK

In the early 1980s, engineers at Bosch were evaluating existing serial bus systems regarding their possible use in passenger cars [4]. Controller Area network is developed from by Robert Bosch in beginnings of the 1980s for automotive industry applications. Controller area network becomes a standard not only for automotive industry but also aerospace industry and industrial automation as well.

Controller are network, provides high speed and robust communication between electronic control units and makes possible to data transfer for both direction. CAN is communication protocol which is used in automotive industry because of its security and reliability [5].

Controller area network protocol works with based on two wires and critical point is these two wires has to be twisted to minimize the electromagnetic effect. These two wires called CAN High (CANH) and CAN Low (CAN L) and they are using for transmitting and receive the data’s between the electronic control units which is located in same network. Controller area network is defined as multi master network which means these electronic control units which is using controller area network does not need central master controller unit.

Controller area network consist of message-based protocol. Which means related data’s transfer with using message frames. This frames can be change depends on controller area network message type but there are some sections which is same for all types. Each message frame includes identifier (ID), data section, control bit section and end frame section. Identifier is using for defining the message, data field includes the information about message itself, control bit section and end frame sections describes the frames beginning and end points and also used for detecting faults. One of the most important key feature of controller area networks detecting faults and fixing fault during data transfer. Controller area network uses specific technique, if there is too much sequential bits which has same value it ads new bits into data flow and this technique calls as bit stuffing. This techniques keep the data’s in synchronized and transmits, receives in a safe way. Other crucial important feature of the controller area network is, prioritize the messages. Controller are network uses a strategy if one messages has high priority and it will transmit first. This feature provides which message transmit and receive speed even in busy communication networks. Controller area network can behave as dominant or recessive. When voltage is existed on the line, dominant signal transmits within 2 volts voltage which is differential voltage of 1,5 volts and 3,5 volts. In contrast, if differential voltage is become 0 volt, 2.5 volts voltage is necessary that transmit recessive signals. This configuration developed to provide the integrity of data communication. These states shown in figure 2.

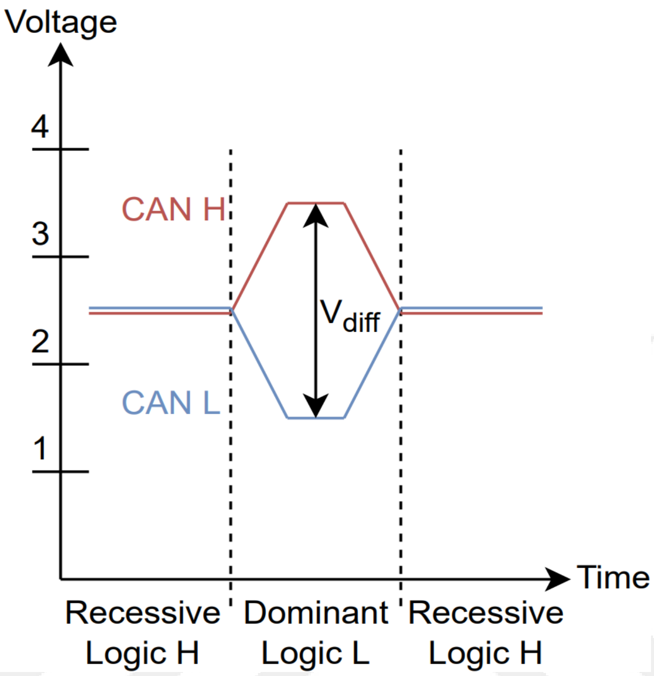

Figure 2: States of CAN Bus

Controller area networks is using in automotive industry to provide connections for electronics control unit, which is developed for control, engines, transmissions, body systems and so on. This control units controls mechanical system, and it shows that controlling mechanical system is not only for automotive industry but also for exist for industrial applications. Therefore, controller area network communication getting more popular for other industries as well. Controller area network is one of the best communication protocols for real time communication applications because of high speed, robustness. Most popular industry for controller area network is automotive industry and has many advantages than other communication protocols.

1. Reliability: Controller area network is developed to withstand to difficult conditions such as electromagnetic field effect, electrical noise, and interference. This robustness provides efficient communications without effecting electromagnetic field of the other components in the vehicle.

2. High-speed communication: Controller area network is the high-speed communication network which is provide speed up to 1 Mbps. With providing high speed between electronic control units, it makes possible real time control for critical systems. The most important factor affecting speed is twisted cable length. Maximum efficiency occurs within the first 40 meters and as the length increases, the speed decreases, you can see in the figure 3.

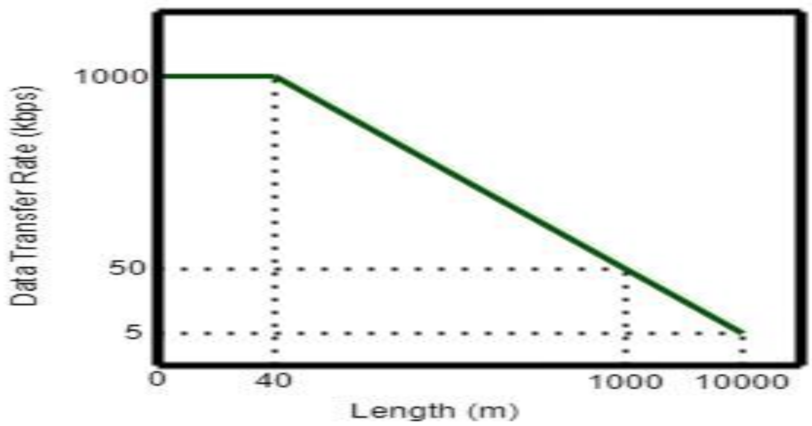

Figure 3: Relation Between Speed and Length

3. Low-cost implementation: Comparing controller area network with other communication protocols like ethernet, controller area network needs less effort and time to integrate which makes inexpensive. This makes controller area network attractive option for automotive architecture.

4. Scalability: Due to controller area network message structures consist of sections it can be scalable. Automotive architectures can consist with few nodes and master’s but also can be consist with complex and multiple master. However, controller area network is compatible for both architectures.

5. Standardization: Controller area network message structure format is standardized. There are three variants for message structures but for these three variants also standardized. Therefore, this standardized message structure increases the compatibility of integrating different vehicle types and this makes also attractive point for manufacturers.

Controller Area Network has three types, and they are, 2.0A, 2.0B and Flexible Data-rate (CAN FD). CAN FD developed for specific architectures which needs increased communication speed, increased and busy communication bus load. CAN FD has same algorithm and mechanism with standard controller area network. However, better solution for complex architectures. Here is the key points which is specific for CAN FD.

Improved Data Transfer Rates: CAN FD has improved data rates comparing traditional controller area network rates. Classical controller area network can provide up to 1Mbps speed. However, in optimal conditions CAN FD transfer up to 15 Mbps. This improvement fulfills the necessity of the augmented bandwidth applications.

Adaptive Data Length: This is one of the crucial features of the CAN FD. Normally, in classical controller area network message structures has constant message format independent from data content it is fixed as 8 bytes. However, CAN FD makes possible to carry data up to 64 bytes for each frame which makes CAN FD flexible.

Interoperability and Legacy Compliance: For complex automotive architecture based on design if CAN FD usage preferred and this is possible to use together with classical controller area networks. Which means architecture designers must not build with choosing only CAN FD but also build the architecture by mixing classical controller area network, CAN FD provides this compatibility. This feature, reduce the integration effort for CAN FD to existing architecture.

Improved Fault Detection Mechanism: To increase the robustness of communication network CAN FD provides complex and improved fault detection mechanism. To utilize from this mechanism complicated algorithms needs to be applied with using cyclic redundancy check (CRC). In this way, ratio of undetectable errors can be reduced.

Optimized Network Bandwidth Utilization: Due to CAN FD has adoptable data length it increases the efficient usage of network bandwidth. Also, due to CAN FD can carry long and large data’s it makes CAN FD effective for the architecture which is necessary to carry large data.

CAN FD takes places when real time, high bus load and increased data transfer is required. Also, CAN FD good choice when it is necessary to integrate new communication platform into existing communication network because it provides versatile and scalable solutions. Differences between CAN Versions shown in table 1.

Table 1 : CAN Bus Versions

	CAN FD	**CAN 2.0A**	**CAN 2.0B**
Speed	Up to 15 Mbps	Up to 125 Kbps	Up to 1Mbps
Range	10 Meters	500 Meters	40 Meters
Identifier	-	11-Bit	29-Bit

All in all, robustness, durability, high speed, and flexibility makes effective controller area network for automotive industry.

2.2.1 Message Prioritization in CAN

Message overlapping problem is not possible with controller area network, due to controller area network is using message prioritization mechanism. In automotive electronic architecture two or more master or electronic control units tries to send message at the same time, arbitration mechanism get involves and it checks which message has higher priority. After, high priority transmits or receives first. Controller area network uses bit wise arbitration. This arbitration compares of messages each sections bit by bit. The message which has high priority will win the arbitration, the node that transmits the message with the highest-priority message identifier (ID) gets to send its message first. A CAN communication's message ID, a special number given to every message, establishes its priority. The priority of the communication increases with decreasing message ID numerical value. This implies that even when a message is in the middle of transmission, the message with the lower message ID will get through first. In the event that two or more CAN bus nodes send a message with the same message ID, "dominant/recessive bit arbitration" takes place. During that process, each node or electronic control unit starts to monitor the messages which sends to network to be sure that all related bits are matched with others. If electronic control unit or node sense a conflict about data transmission, node will stop the process and wait for transmitting of higher priority message. To avoid message conflict for controller area network messages in network, electronic control units or nodes should apply orders, and this is called as CAN protocol. These orders which is defined as CAN protocol defines maximums number of nodes on the communication network architecture, message lengths, and so on. With using software nodes or electronic control units can guaranteed that related controller area network messages are sent properly and also received properly, without any conflict or overlapping. This priority mechanism in controller area network communication is works with message identifier which is defined uniquely depends on message. If number is decrease which means has higher priority. This makes possible even during message transfer of communication network, if identifier value is lower which means it will still be transmitted first and given a higher priority. Instead of defining priority during transmission, arbitration is used. The CAN bus initiates randomly message transmission when more than two nodes or systems try to send a message at the same time.

As mentioned before, each node or electronic control units not only sends or receives the messages but also it continuous to follow the network to be sure that signal received or transmit. Related node catches any overlapping, conflicting or other problem that might be occurred. Immediately stops the Tx/Rx progress and starts to follow highest priority message.

When optimal conditions provide for controller are network communication, transmission is starts and during transmission most significant bit (NSB) transmits first. After it continues to rest of the bits which is located inside of the message and related nodes starts comparing with the corresponding bit of related message. Any overlapping, conflicting, or other issues are detected by the related node. Stops right away. After waiting for the highest priority message. When arbitration progress is done and when a message is decided for transmission, all related nodes or electronic control unit will understand that communication line is busy, and they will start to wait before transmitting or receiving their messages. This mechanism supports robust and healthy communication without any interference.

This message prioritization mechanism can be shown with a simple example. Automotive software architectures include many nodes, mostly more than 70. For now, assuming there are two nodes in the communication line, and they called as node A and node B. Again, assuming both nodes are needs to send their messages at the same time and node A messages identifier number is 50 and node B message identifier number is 100. Due to node A has lower identification number than node. Node A has priority, and it will transmit first. When transmission continuous node A starts transmitting during that time node B starts to follow and compares bit by bit. If Node B detects any overlapping or conflict during communication, it will immediately stop and just starts to wait which means listens to higher priority message.

In conclusion, message prioritizing mechanism has crucial importance for controller area network. If identifier number has lower value than it makes that message higher priority. This prioritization mechanism makes controller area network indispensable for critical safety relevant systems, and it increases the efficiency.

2.2.2 CAN Message Frame

Controller are network can provide communication with using message packages and based on architecture these packages can include many of the CAN message. One singe controller is message structure consist of sections and this section can be change based on CAN variant such as 2.0A, 2.0B or CAN FD. However, in this section most standardized version of the CAN message structure examined.

1. Start of frame (SOF) bit: This section is beginning of the message, and it describes that messages node request and start to communication with another related node.
2. Identifier bit: As mentioned before, controller area network has crucial feature which is called as message priority, and this section determines the priority.
3. Remote Transmission request: This section consist of 1 bit, and it defines that any node is sending or receiving message or not also describes another node is requesting specific data or not.
4. Control field: This field consist of 6 bits, and it works for control mechanism that checks described data length.
5. Data field: This is the main section, and it includes the data which is needed to transmit or receive. This section can be increase up to 8 bytes based on controller area network variant.
6. Cyclic redundancy check (CRC): this section consists of up to 16 bits length and Works for detecting fault inside of the messages.
7. Acknowledgement (ACK): This section can be up to 2 bits and shows that node received or transmit the data’s correctly or not.

8. End of frame (EOF) bit: This is the last section of the message structure, and it can be up to 7 bits length, and it remarks that message is ended.

To summarize, classic controller area network message includes 8 sections to transmit the message properly. Visualization of the message frame shown in figure 4. Usage of this sections of the CAN message structure, it makes CAN communication robust and efficient.

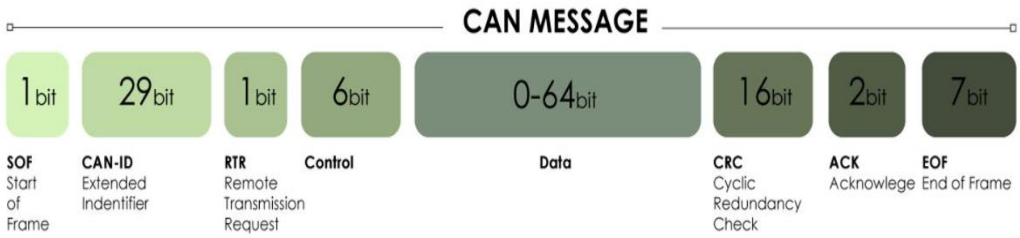

Figure 4: CAN Message Frame

2.3 DIAGNOSTIC COMMUNICATION AND SERVICES

One of the most important points for automotive electronics architecture is diagnostic systems because, it is not only effects development process but also effects production line and after sales. Therefore, industry undergoing from high-tech transformation. The diagnostic option plays an important role to identify and locate the causes of failures [6]. A manifestation of the automotive industry's diagnostic communications services and skills in a comprehensive analysis. For this purpose, it illustrates the basic importance of these two kinds of service, their technological foundations, and where this field is going next. A modern era in diagnostic communication is therefore important to the successful functioning of a modern car. Experience shows that diagnostic communication is not merely time consuming, and its results are not always accurate since with two identical settings or components, there almost nestled inside an infinite whole host of different possibilities available to for our selection. Therefore, diagnostics itself has thereby become a very important aspect of OBDII (On-Board Diagnostic II). That is why diagnostic communications in modern cars also makes a real contribution to environmental protection, safety, and maintenance efficiency.

To have ability to observe status of the electronic systems which is located in vehicle architecture has vital importance. Diagnostic systems are working much in the way of tireless catchment, continuously monitoring a variety of vehicle systems, and quickly spotting any anomalies and potential safety hazards in the fight to prevent accidents and keep occupants safe. Stringent environmental regulations have made emissions control a top priority in the automotive marketplace. Diagnostic systems check in on outputs from systems responsible for managing emissions to make sure that vehicles are within specs, ultimately helping to lessen the environmental burden of the car industry. Diagnostic commands transports layer by layer and general one is shown in figure 5.

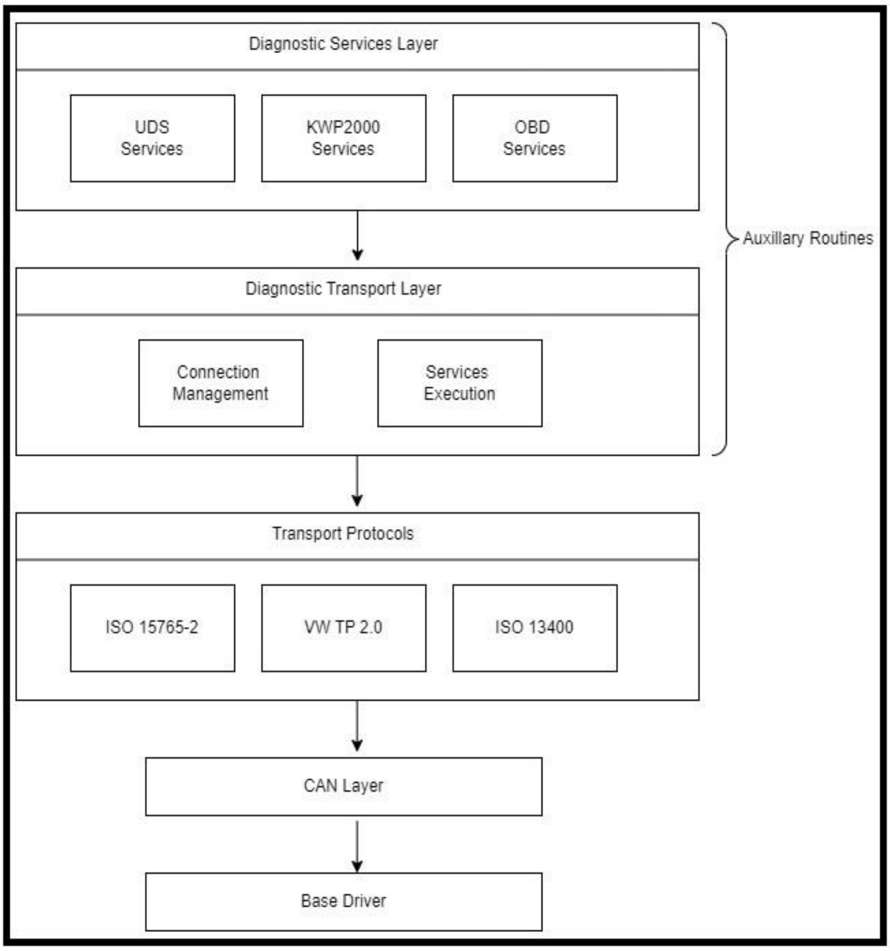

Figure 5: Diagnostic Command Set

Before describing technical details about unified diagnostic services and keyword 2000. It is mandatory to describe technical background information's. They are.

1. On-Board Diagnostics (OBD), is the universal connection diagnostic systems for vehicles independent from manufacturer and vehicle type. Based on automotive regulations it is mandatory that each vehicle must have that port. OBD systems uses many variants of sensor to observe the vehicle control unit systems. Also, OBD is using not only for observing but also flashing. Therefore, it makes user friendly technology for technicians and engineers. Schematic and pinout information of OBD-II socket shown in figure 6.

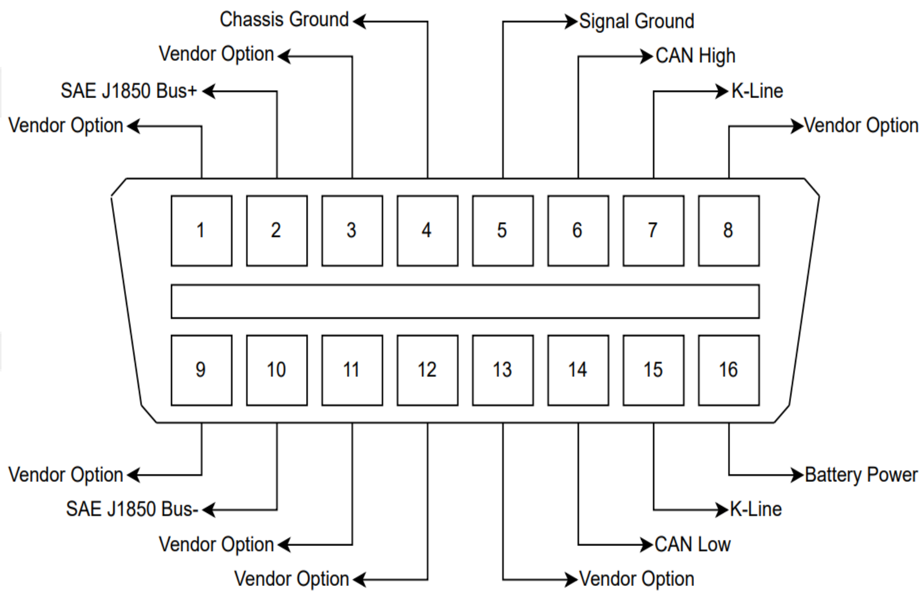

Figure 6: Schematic of OBD-II Socket

The automobile domain's tapestry situates On-Board Diagnostics as critical pivot point On-Board Diagnostics oversees a symphony dutifully choreographing emissions regulation, diagnostic granularities, forward-leaning maintenance, fuel-side parsimony, diagnostics homogenization, shopper empowerment and regulatory alignment These multi-faceted academic-lens lynchpins wield embodied spectra binding automotive dialogue indispensably through technical prowess, legislative entanglement and ecological stewardship.

When diagnostic systems getting more technological features, they refers to previous historical backgrounds On-Board Diagnostics is an essential element of modern vehicular technology and is the basis for self-diagnostic and reporting capabilities of modern automobiles. At the formatives stages of OBD in the 1960s, early vehicles incorporated a simple system of identifying their malfunctions through warning lights. These early warning lights led the way to the sophisticated system of OBD-I in the 1980s. In this era no standardization existed between manufacturers, which made it difficult and time consuming for technicians and mechanics to repair vehicles that were having malfunctions because the procedures were different for each manufacture.

If it needs to going to deeper into the history of these advanced diagnostic systems. An important shift to a standardize OBD system occurred in California in 1991, as the result of a regulatory initiative by the California Air Resources Board. When it introduced its OBD-II program, requiring extensive monitoring of the emission control systems on vehicles, it also adopted a generic DTC. A standardized DTC had been established for use with the new emissions system, as a requirement of the were formulated. The OBD-II standard specified the type of diagnostic connector and its pinout, the electrical signaling protocols available, and the messaging format. It also provided a candidate list of vehicle parameters to monitor along with how to encode the data for each. The U.S. Environmental Protection Agency required that 1996 and newer vehicles pass a modified form of the California Air Resources Board mandated generic DTC. The result was a comprehensive standard set of DTCs. An analysis of these standardized DTCs continued, to points to a breakup of category in the OBD II implementations of the various automobile manufacturers. OBD-II diagnostic services shown in figure 7.

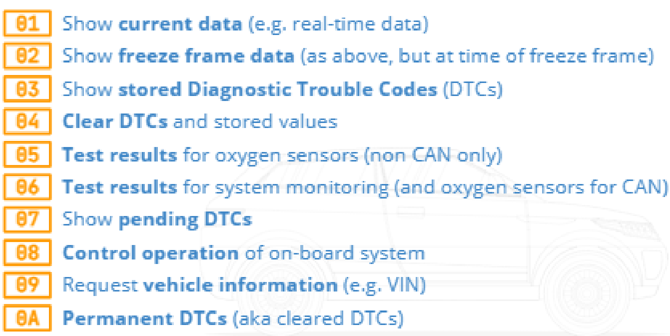

Figure 7: OBD-II Diagnostic Services [7]

OBD-II has been widely adopted worldwide with the main reason being from a practical perspective: to cut down on emissions and to slap a set of standards on those emission values with which everyone must comply. Now, far beyond the United States, the regulations spawned by OBD-II have worked their way all across Europe and into every corner of Asia. More generally, as the OBD-II system began to reach overseas, it got bigger. Communication protocol specs got tighter, data logging capabilities were expanded, and real-time monitoring was added. What began as a system to simply control emissions on the car, grew into one that controlled the cars and their components. Where the OBD connectors are concerned the OBD connectors used in car diagnostics are built in accordance with the OBD-II Standard, and there are essentially two main types – Type A and Type B. The classifications of Type A and Type B are not always universally designated within the standard as "Type A" or "Type B" but instead are distinguished by their physical appearance and pin layouts.

As described before OBD-II is the universal standardized latest version of connection port for vehicles. However, even connector is standardized there are two different types. These types have only pin configuration differences.

OBD-II Type A Connector: The Type A connector has a trapezoidal form with 2 rows of 8 pins. The two center pins are both wider than the others, making it difficult to plug in incorrectly. It is actually a good design for a connector that you may be plugging in and out of the dash many times a day. After then the most important pins for routine diagnostics are a mirror arrangement that makes a lot of sense for cars and light duty trucks. So those pins (4, 5, 7, 9, 12 and 16) are probably all we need to look at. The basic shape suggests many things, some of which did not pan out, and that second, taller row probably accounts for at least a few more.

OBD-II Type B Connector: The Type B connection’s rectangular form presents a problem for double digit pin counts, so they added breakouts in the 5th and 10th positions with a little extra “tooth.” This one also holds 16 pins. At least a half dozen of the “even” pins from the standard diagnostic layout (2, 4, 5, 10, 16, etc.) get used here. Type B’s are common in larger vehicles that do not seem to be so common. Well, around here.

They function as standardized diagnostic interfaces within the OBD-II standard. However, dedicated pins enable emissions-related diagnostic capabilities. Nevertheless, it is critical to understand that manufacturers may add extra pins to support registered systems or specific features. The wider diagnostic context thus permits some diversity within the OBD-II standard. It ensures a baseline homogeneity for emissions-related diagnostics.

OBD-II is current universal standardized port for manufacturers. However, third generation of OBD is developing. It is not suggesting that oodles of standards and wishful thinking are tantamount to real progress, of course. That said, it's the reasons behind the speculated OBD-III's creation that are intriguing. Although OBD-III is still in the early stages and may yet take years to implement, it is not that it as a direct replacement. Instead, it's a more accurate diagnostic standard. The expectation is to not only standardize diagnostic procedures, but to improve the communication between systems. The increasing interconnectedness of advanced driver assistance and self-driving features makes it harder to diagnose a vehicle as if its systems existed in isolation. Continuing to develop OBD this long after its introduction not only underscores the dramatic years-long shift for OBD, but its establishment as a cornerstone of the modern car's technological landscape.

2. Other standardized indispensable technical mechanism is diagnostic trouble codes, known as DTCs, constitute an alphanumeric system for the identification of specific automotive malfunctions. These codes consist of formats and even format is the same but meaning of the numerical values can represents different point depends on manufacturer.

DTC codes help you understand what needs to be fixed so you can keep your vehicle operational and safe [8]. DTC message has a format which includes information about root cause of the trouble and shown in figure 8.

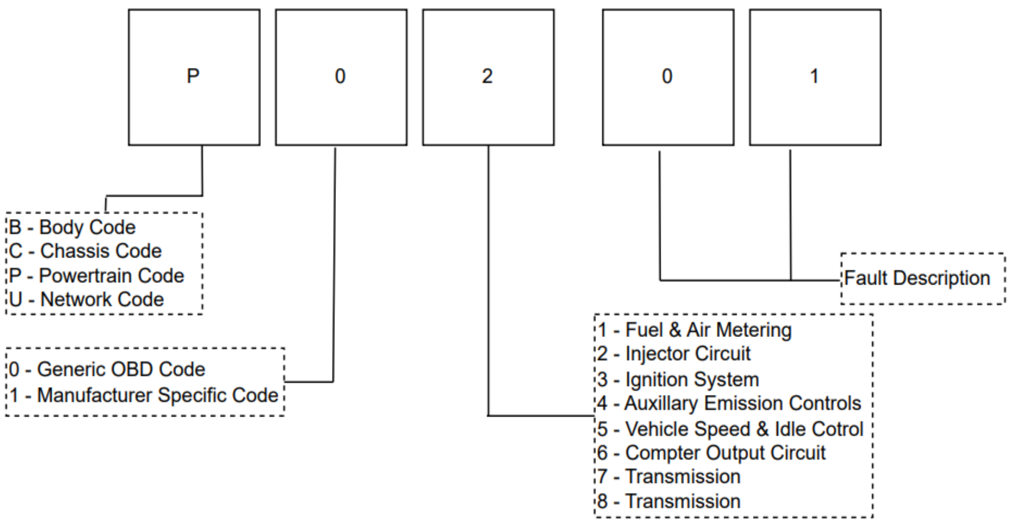

Figure 8: Format of DTC

3. Day by day fault detection and fixing mechanism getting more importance. Telematic mechanism also supports that. Usage of telematic board modules inside of the electronic architecture of the vehicle makes possible remote diagnostic features. Remote diagnostic features are increasing the efficiency and reduce the costs.

4. Last point for this topic is, to include data analytics and artificial intelligence into diagnostic systems. If it is come together with remote diagnostics, for future it might be possible to apply architectural changes, modifications without any technicians or engineer intervene.

To wrap up, it is evident that both diagnostic communications and services are essential components of the automotive sector. Their contributions are not only vectors of the industry's evolution, but significant aspects in the augmentation of vehicular safety, environmental compliance, and maintenance practices. OBD and DTCs are just a few of the accompanying technologies, along with telematics.

Automotive industry trying to be stay in safe zone by using standardized systems for their vehicle electronics architecture. This standardized systems mostly focused on diagnostic services due to diagnostic services effects not only production or engineering department but also effects after sales as well. In that case, unified diagnostic services, and keyword 200 protocols getting importance.

2.3.1 Unified Diagnostic Services

In this section, unified diagnostic services will be examined which is focused point. Unified diagnostic services are powerful, strongest, and robust diagnostic network mechanism currently for automotive industry. Also, due to diagnostic communication services is most standardized protocol, it is more effortless to integrate into systems. UDS is a unified diagnostic services protocol standardized by the International standards Organization (ISO) in the ISO 14229 [9], these standardized services which is provided from by unified diagnostic services includes reading errors which occurred not only in controller area network but also in electronic control units. Also clearing error codes if they are passive errors. Unified diagnostic services also have a message structure and generalized version of this structure shown in figure 9.

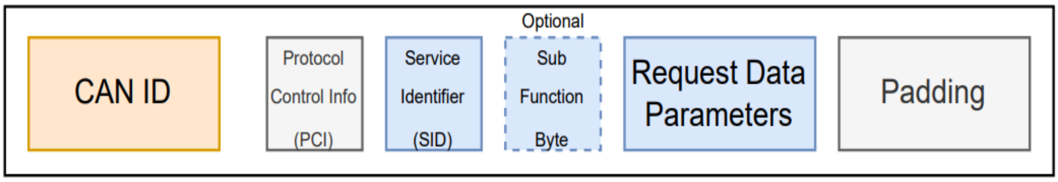

Figure 9: UDS Message Structure

One of the major advantage points for UDS is to be using standardized communication platform between electronic control unit and diagnostic tool. As mentioned, figure 9, UDS also has message structure and this structure mostly consist of 6 sections. Only one section which is called as “Sub Function Byte” is optional and rest of them completely standard for UDS on CAN mechanism and this makes easier for technicians and engineers. Unified diagnostic services allow for multi directional communication and this multi directional communication provided from by sub services of unified diagnostic services. This services also makes possible to read error, clear the error, contact with ECU, flash programming of ECU and so on. As mentioned before unified diagnostic services has error detecting sub services but the point is these services also can work simultaneously during diagnostic communication which means online communications, and this makes unified diagnostic services faster for error handling process. Other important feature of unified diagnostic services is, it not only works with CAN communication, but also works with LIN, FlexRay, Ethernet and so on. Therefore, manufacturers which select UDS in their architecture design have ability to integrate UDS with many variants of communication platform.

Unified diagnostic services not only have advantages for communication platform layer. Vehicles needs to a specific tool for providing diagnostic communication between technician and vehicle. This tool calls as diagnostic tool or tester. Each manufacturer has their own specific diagnostic tool. If they used unified diagnostic services in their automotive architecture, it makes diagnostic tool available for all variant of manufacturers vehicles. This is also reducing the costs. Unified diagnostic services have many advantage’s with comparing other diagnostic protocols and one of the important advantages is providing high speed for diagnostic communication. However, this speed can be limited based on communication platform which means if UDS implemented based on CAN communication and UDS can provide maximum speed which CAN communication is provide. Therefore, choosing communication platform also as vital role. Due to UDS has many standardized sub services, it can handle many functions less steps than other services and it makes faster of UDS.

There are a lot of standardized UDS subservices exist. However, in this section most generalized and standardized 9 sub services will be examined with detail which is defined from by the ISO 14229-1:2013 standard.

1. Diagnostic Session Control (0x10): With this sub services tester have ability to active and deactivate the diagnostic services. Without this sub services, if diagnostic services start automatically, it can increase the load on bus.

2. ECU Reset (0x11): As mentioned before with using diagnostic services electronic control units can be repaired and other vehicles ECU can be used for another vehicle. However, to provide this flexibility, tester must reset the electronic control unit and this sub services provides that feature.

3. Read Data By Identifier (0x22): This sub services uses for to start reading electronic control unit with using unique identifier.

4. Read Memory By Address (0x23): Electronic control units especially which is developed specifically for automotive industry has memory which is consist from by sections. Each memory sections defines for specific functions. With using this sub services this memory section reading feature can be available.

5. Write Data By Identifier (0x2E): Electronic control units includes variables which has unique identifiers. With using this services tester will be able to write data into specific identifier which is located in the ECU.

6. Tester Present (0x3E): This sub services mostly used for beginning of the diagnostic session to be sure that ECU and also tester are ready to start.

7. Security Access (0x27): This sub services uses for ECU which has specific and highly critical duty. With using this sub services manufacturers can be sure that unauthorized access is not possible to critical ECUs.

8. Communication Control (0x28): This sub service is using for to pause the communication between ECU and tester. Also, can be used for to change parametrizing during the diagnostic communication.

9. Routine Control (0x31): Depends on manufacturer architecture, some of the electronic control units can have routines. These routines represent calibration of sensors which is connected to related ECUs. With using this sub services these routines can be activate or deactivate.

These mentioned nine general standardized UDS sub services provides generalized and again standardized approach for all manufacturers vehicle variants, and it makes possible to perform diagnostic functions independently from manufacturers.

Below, full list of sub services of unified diagnostic services can be found which is defined from by ISO 14229-1:2013 standard:

1. 0x10 - Diagnostic Session Control
2. 0x11 - ECU Reset
3. 0x14 - Clear Diagnostic Information
4. 0x19 - Read DTC Information
5. 0x22 - Read Data by Identifier
6. 0x23 - Read Memory by Address
7. 0x24 - Read Scaling Data by Identifier
8. 0x27 - Security Access
9. 0x28 - Communication Control
10. 0x2A - Read Data by Periodic Identifier
11. 0x2C - Dynamic Define Data Identifier
12. 0x2E - Write Data by Identifier
13. 0x2F - Input Output Control by Identifier
14. 0x31 - Routine for Control
15. 0x34 - Request for Download
16. 0x35 - Request for Upload
17. 0x36 - Transfer Data
18. 0x37 - Request Transfer for Exit
19. 0x3D - Write Memory by Address
20. 0x3E - Tester Present
21. 0x83 - Access Timing Parameters
22. 0x84 - Secured Data Transmission
23. 0x85 - Control DTC Setting
24. 0x86 - Response on Event

Day by day with developing technology these services can be modify and improved. However, each service will have specific role and makes easier testers duty while providing communication between electronic control unit.

Also unified diagnostic services effected layered mechanism of Open Systems Interconnection (OSI) model. Before describing detail of the OSI model of UDS. It is mandatory to describe OSI itself first. Open Systems Interconnection model is defined from by ISO to standardized telecommunication frameworks and also computing systems as well. OSI consist of 7 layers and each layer has their specific role. Layers also makes easier to understand the topology behind. Here is the 7 layers of OSI:

Physical Layer (Layer 1): This layer represents physical connection between the devices with using cables connectors and so on.

Data Link Layer (Layer 2): After physical layer with using this layer it makes possible to be sure that related data’s will starts to follow in a safe and secure way. Also, provides error detection and access control.

Network Layer (Layer 3): With using this layer related data packages can be addressed to related networks. During that addressing process also optimal data transfer path can be defined.

Transport Layer (Layer 4): This layer provides transportation of data to related network. During that transportation end to end mechanism is performs and with using end to end mechanism this layer can be sure that related data delivered or not.

Session Layer (Layer 5): This is the 5th layer in OSI, and this layer makes possible to provide communication between different variants of the machines. Also provides synchronization and ending.

Presentation Layer (Layer 6): This is the 6th layer of the OSI. Main duty of this layer is to apply encryption of the data and provide the format of data.

Application Layer (Layer 7): And this is the last layer of the OSI model. This layer provides network services to end users and it has many networks application including HTTP, FTP and SMTP. OSI model of UDS shown in table 2 for each layer.

Table 2 : OSI Model of UDS

Layer	Layer Name	Standards	Description
7	Application Layer	ISO 14229-1	UDS: Specification and Requirements
6	Presentation Layer	Not Applicable	-
5	Session Layer	ISO 14229-2	UDS: Session Layer Services
4	Transport Layer	ISO 15765-2	Diagnostic on CAN: Network Layer Services
3	Network Layer	ISO 15765-2	Diagnostic on CAN: Network Layer Services
2	Data Link Layer	ISO 11898-1	CAN: Data Link Layer and Physical Signaling
1	Physical Layer	ISO 11898-2	CAN: High Speed Medium Access Unit

OSI model helps to understand communication and telecommunication platform and makes easier to to design a communication network. Possible faulty network communication platform design can be occurred without applying OSI model structures.

Unified diagnostic services is also applies OSI model structure. However, applying OSI mode for UDS is nor directly the same because based on cases application areas are different. Therefore, UDS implies 3 parts of OSI layers and rest of them is standardized. Layers are shown with using basic automotive architecture in figure 10. Here is the 3 specific OSI layers which UDS implies.

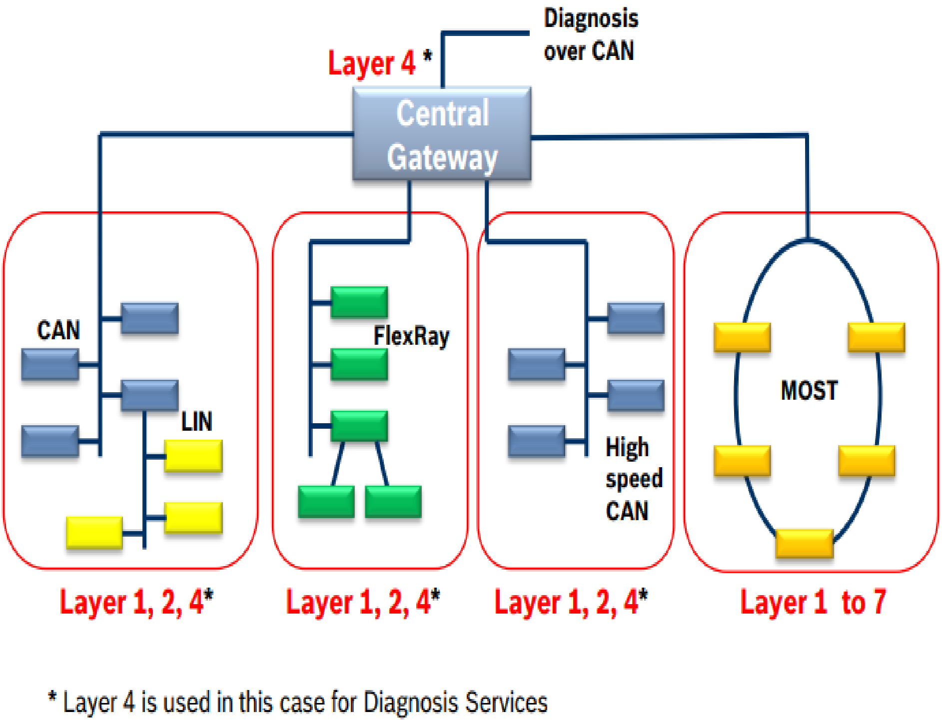

Figure 10: Basic Automotive Architecture [10]

Application Layer (Layer 7): Unified diagnostic services follow the last layer of the OSI. With this layer UDS defines diagnostic services to provide communication between tester and electronic control unit.

Transport Layer (Layer 4): This is the 4th layer in OSI and it work for UDS as to carry related diagnostic network data's to related systems. This transportation protocol is also part of ISO 14229 standard.

Network Layer (Layer 3): This is the 3rd layer of OSI model. This layer also works for UDS as addressing mechanism. UDS has specific sub diagnostic services which works based on network layer of OSI. With using this layer UDS sub diagnostic services address related messages to related systems. Important point is which communication platform is used during diagnostic network such as CAN, LIN, FLexRay and so on.

Remarkable point is that unified diagnostic services matches mostly with the high level of the OSI layers. Matching of the lower layers of the OSI model to UDS is depends on the selecting communication network platform. Based on platform number of matching layers can be change. Visualization of OSI layers for UDS is shown in figure 11.

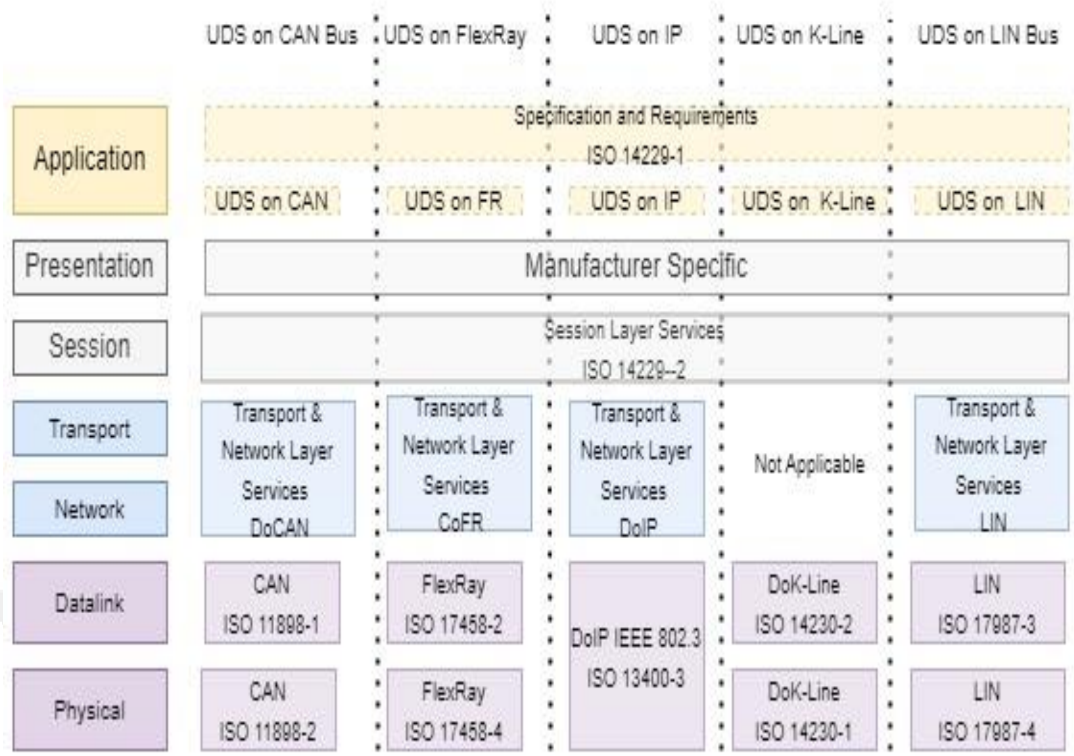

Figure 11: Visual of OSI Layers of UDS

Unified diagnostic services are a diagnostic protocol which is standardized from by ISO to provide communication between testers to ECUS independently from manufacturers. As mentioned, usage of UDS follows standardized approach and this approach reduces the integration and development efforts and also reduces the costs as well. Way of standardization of UDS Is also includes OSI model which is mentioned before.

2.3.2 Keyword Protocol 2000

Keyword 2000 (KWP200) protocol is one of the important and common protocol which is used for diagnostic communication. International Standard Serial communications KWP2000 (ISO/WD 14230) is used to define the general requirements of diagnosis based on serial communication system [11]. Keyword 2000 protocol also has standardized mechanism to provide diagnostic communication between electronic control unit and tester. Even keyword 2000 protocol has standardized and complex structure this complexity might be a negative point for KWP2000. Also, standardized structure is not much standardized as UDS. KWP2000 provides communication speed, from 1.2 kbps to 10.4 kbps.

This communication speed also depends based on communication platform. However, due to KWP200 only provides diagnostic based on K-Line and CAN communication platform it is limited. KWP200 also provides efficient data transfer comparing previous architecture. It uses a message format which consist of header and payload. KWP2000 favors symmetrical communication sequence where the number of request and response messages between the testing device and server are symmetrical [12]. As mentioned before, supported communication lines are limited for KWP2000. It is only supporting communication networks of K-Line and CAN. KWP2000 communication over CAN shown in figure 12.

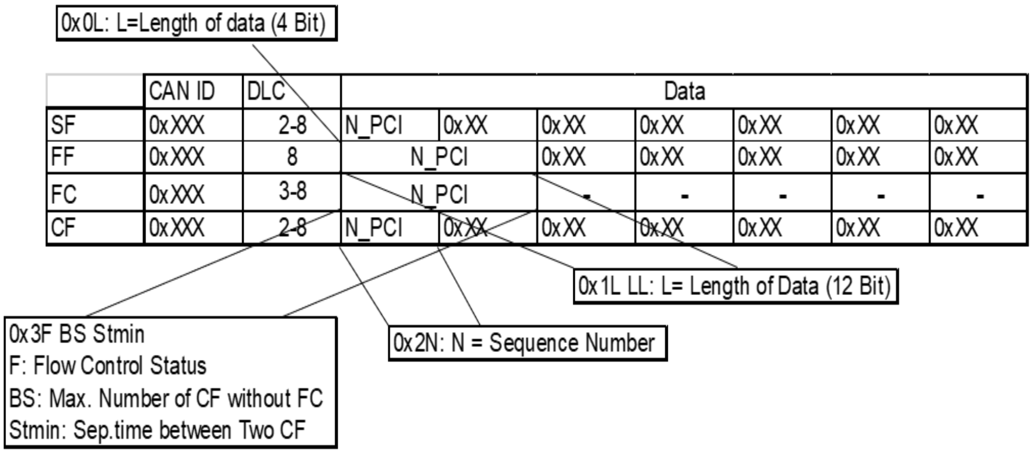

Figure 12: KWP2000 Over CAN Bus

KWP2000 protocol, uses again sub diagnostic services to communicate and make possible to apply flash programming for electronic control units and for that reason KWP200 defines set of services. This diagnostic service includes standardized functions such as read, clear the diagnostic trouble code, freeze the data frame, and check the real time data.

KWP2000 communication services provides ability to test communication network, resetting the communication interface. So, these abilities make possible to re-flash programming the electronic control units which is already flashed before. There are few physical layers for communication, and they are, ISO 9141-2, J1850 VPW, J1850 PWM, and CAN which is supporting from by KWP2000. KWP2000 also can be configurable with the other diagnostic protocols. Also, KWP2000 presents standardized communication network way to communicate with electronic control units, and also KWP2000 diagnostic protocol can be used for flash programming and configuration not only for aftersales but also for production line. All in all, these features make popular KWP2000 as well.

As mentioned before, KWP2000 has many advantages for diagnostic communication but also has some disadvantages as well, and most critical negative point is to provide limited and slow communication speed. This limited speed causes long diagnostic analyze time and longer flash programming time. This limited speed reduces the number of produces vehicles and implicitly effects the cots negatively.

Security features getting importance day by day. Especially cybersecurity protection mechanism getting mandatory standard for automotive industry as well. However, other negative or weak point of KWP2000 is that has limited ability to provide security. Which means that can make defenseless the systems. Also KWP2000, has limited support for futures new technology such as hybrid and electrical vehicles powertrain systems.

As described before, KWP2000 provides some services to provide robust communication between ECU and tester. These services can examined under three sections:

1. Diagnostic services: Diagnostic services makes possible to detect and analyze the failures in the ECU. Also, diagnostic services include sub services. These sub services are not detailed like UDS. Here are the diagnostic sub services:

• Read Diagnostic Trouble Codes (DTCs): Each electronic control unit keeps diagnostic troubles codes in their memory to localize the error. This service works to reach DTCs of ECUs.

• Clear Diagnostic Trouble Codes: Diagnostic trouble codes generalized under two section like active and passive. Clearing of DTC is possible with using this service. If the DTC is passive.

• Read Data by Local Identifier (LID): With using this service, local identifier can be defined and makes possible to get data from ECU.

• Read Data by Common Identifier (CID): This service can define common identifier and this identifier works for user to get data from ECU.

• Read Freeze Frame Data: If an error occurred and depends on that error related DTC will be generated. This service makes possible for user to analyze related DTC code by freezing data frame.

2. Communication services: Communication services works for to start communication and also provide robust communication between ECU and tester. This service has also subtitles. They are:

• Start Communication: With using this services communication line can be activated.

• Stop Communication: With using this services communication line can be deactivated between ECU and tester.

• Tester Present: This service works for electronic control units and says that there is a diagnostic tool is available on the line.

3. Programming services: As mentioned before KWP200 can support flash programming feature. This feature is providing with using programming services. They are:

• Request Download: When related software is ready in the server, after using this service-related software can be request.

• Request Upload: With using this service, existing software can be upload.

• Transfer Data: While flash programming starts, this service starts to data transfer.

• Request Transfer Exit: If any problem occurred during download or upload transfer without a problem it can be the user request, transfer process can be cancelled.

Described diagnostic, communication, and programming services supports KWP2000 to keep standardized protocol and provides limited flexibility. These services work for provide robust communication and detect the error and after fixing error they also provide flashing new software. However, these all services require physical connection and that makes KWP2000 not possible to use for remote diagnostic feature and this limits the flexibility.

Even KWP2000 has weaknesses, still it keeps being popular within old architecture generation vehicles. However, day by day new diagnostic communication protocols like UDS getting more popular due to it provides new technology.

2.3.3 Automotive Open System Architecture

To increase the standardization and efficiency, Automotive open system architecture (AUTOSAR) created. AUTOSAR, is a consortium to make more standardized software architectures and libraries and reduces to effort to developed same software libraries again and again. AUTOSAR represents the first widespread approach to establish a standard covering virtually all aspects of the development workflow of automotive embedded software [13]. AUTOSAR consortium founded in 2003. Almost all manufacturers are part of AUTOSAR consortium. Some of them part of AUTOSAR consortium as core member, some of them as developer member and some of them as supporting member. To give few examples like, BMW, Daimler, VW, Ford, GM, Toyota, PSA, Continental, and Bosch.

Main target for AUTOSAR is to make software independent from hardware which means electronic control unit. AUTOSAR slogan says that “Cooperate on standards, compete on implementation.” Electronic control units include embedded software modules and with using standardized AUTOSAR technique it become possible to build a software which is independent from hardware. Without applying AUTOSAR standardized technique automotive architectures can faced problem for future. In corona pandemic effects shows that chip crisis might be occurred. And, if the automotive software architecture developed without AUTOSAR which makes that software dependent to a single electronic control unit. After, if manufacturer needs to change hardware or electronic control unit because of a reason such as chip crisis. Software itself needs to be developed from beginning again. However, if the AUTOSAR technique is used, then manufacturer can change electronic control unit without making major changes in the software.

AUTOSAR itself is compatible with almost all known automotive specific regulations and standards. AUTOSAR also reduces the developing effort, integration effort, maintenance cost as well. Not only reduce the costs but also increases the code quality as well. AUTOSAR is also compatible with new technologies of powertrain and body function systems. Electronic control units include developed software and that software roughly consist of two sections and they are: application software and basic software. AUTOSAR following layered structure in the basic software level and with using layered mechanism basic software can be adopt one ECU to another ECU. AUTOSAR layers consist of 5 main layers and software modules isolated in each layer inside and each layer communicate with other layer. Standard AUTOSAR layers shown in figure 13.

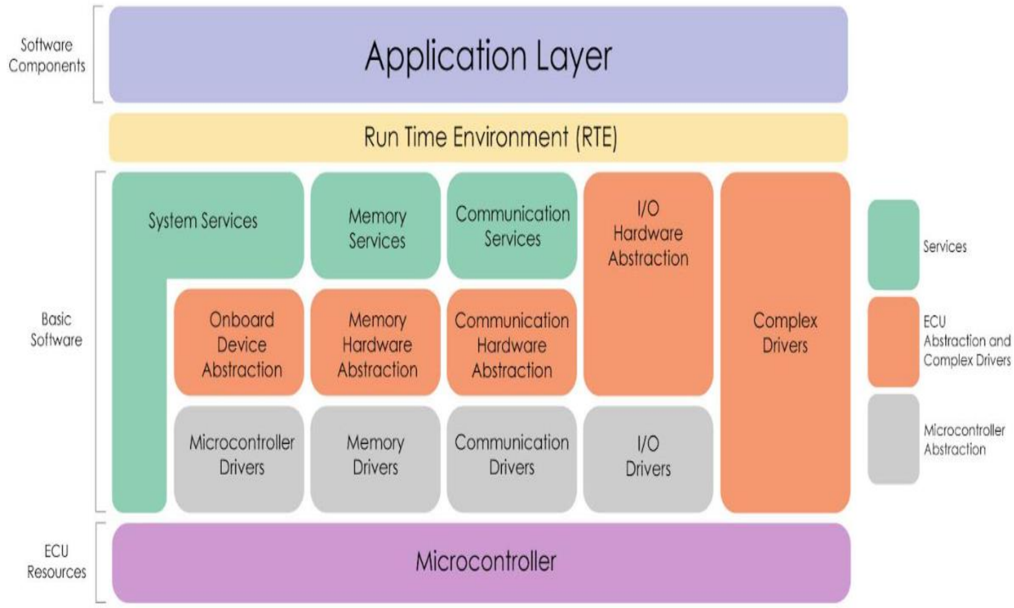

Figure 13: AUTOSAR Layers

Microcontroller Abstraction Layer (MCAL): This is the first and bottom layer and it includes drivers. Main duty for this layer is to separate higher layers from microcontroller. This layer includes,

ECU Abstraction Layer (ECAL): This layer's acts like an interface for MCAL layer. With using this layer software getting independent from ECU which means hardware. This layer also includes specific drivers like injection control and so on. Main duty of this layer to provide independency for rest of layers.

Services Layer: This layer includes most important and critical software's. Which is necessary for implementation of communication systems, memory management, diagnostic services, fault detection mechanism.

AUTOSAR Runtime Environment (RTE): Based on specialist and automotive researcher this layer is called as "heart" of AUTOSAR. As mentioned before automotive software roughly consist of two section and they are application software and basic software. Basic software mostly developed using C languages. However, application software's are developing with using model-based software tools like Simulink, LogiCAD and so on. This layer provides communication between application software and basic software which means act like a bridge.

Application Layer: Application Layer: This layer includes application software which is developed from using model based software development toolchain. This layer defines functions which can varies from manufacturer to manufacturer.

AUTOSAR itself consist of two sections, classic and adaptive. Until this section described version is classical version. Working principles, layered structures and methodologies are same for classic and adaptive. AUTOSAR adaptive layers includes much more complicated driver strategy and software. The reason behind of that, AUTOSAR adaptive developed to catch new technologies such as, advanced driver assistance systems, autonomous driving systems, hybrid powertrain mechanisms, remote technologies and etc.… Here is the major advantages characteristic specification of AUTOSAR adaptive.

Modularity and Decoupled Software Components: AUTOSAR adaptive has much more flexibility and modularity comparing AUTSAR classic. This modularity includes sub layer inside of the AUTOSAR classic layer. This flexibility makes easier to adapt complicated systems into architecture.

Dynamic Configuration Capabilities: AUTOSAR classic has static configuration mechanism. However, AUTOSAR adaptive has complicated configuration mechanism, and this mechanism makes possible to apply software changes into system without re-compiler. With using this mechanism, integration time of complex system such as autonomous driving software modules can reduces.

Service-Oriented Communication Architecture: With using this advantage AUTOSAR adaptive managing of the complex software components getting easy. AUTOSAR adaptive also supporting from by service-oriented architecture (SOA). Due to modern vehicles architecture complexity getting increase this point makes selectable advantage for AUTOSAR adaptive.

Over-The-Air (OTA) Update Capabilities: The most remarkable advantage of AUTOSAR adaptive is to support over the air technology. With using this technology, independent from place and tie all necessary software updates can be applying wirelessly.

Compatibility with Dynamic and Open Environments: AUTOSAR adaptive has also able work in an environment with open source technology to reduce the complexity of the modern vehicles architecture.

To summarize, even AUTOSAR adaptive has same methodology and layered structure with AUTOSAR classic, it provides much more flexibility and modularity comparing with AUTOSAR classic. This makes AUTOSAR adaptive more logical choose for complex systems such as ADAS, Automonous driving, OTA and etc. The adaptive AUTOSAR diagnostics follows the UDS 14229-1 specification to implement the functionalities of diagnostic services [14]. Key differences between AUTOSAR classic and adaptive shown in table 3.

Table 3 : AUTOSAR Comparison

Subject	**AUTOSAR Adaptive**	**AUTOSAR Classic**
Application	Distributed	ECU Defined
Communication	Services – Ethernet+SOME/IP	Signals – LIN, CAN, FlexRay
Requirement	Flexibility, Performance, Update, Upgrade	Efficiency, Safety and Security
Hardware	MPU, SoC, Accelerators - Changing	MCU - Steady
HW Architecture	Centralized and Connected	Distributed
SW Architecture	Modular	Monolithic
SW Configuration	Dynamic	Static

CHAPTER III
COMPARATIVE ANALYSIS OF UDS AND KWP2000

3.1 ANALYSIS RESULT OF UDS AND KWP2000

Until this chapter, detailed description is provided for UDS and KWP 2000. Also, before making analysis and tests, all necessary technical background information provided. As mentioned, unified diagnostic services and keyword 2000 protocols are the most common protocols in automotive industry. During this analysis, identical conditions are provided. Same electronic control unit used, Controller area network platform used, and same teste tool used. Details of control unit can be found in table 4.

Table 4 : The Features of ECU

Core	Infineon AURIX TC299 32 Bit
HSSL	320 Mbit/s
EEPROM	768 Kbyte
CAN Bus	6
ASIL Level	D
Program Flash	8 Mbyte
Data Flash	766 Kbyte

It is already proofed that unified diagnostic services has more sophisticated protocol compared to KWP2000. Several key advantages also shown in table 5.

Table 5 : UDS and KWP2000 Comparison

Subject	UDS	KWP2000
DTC Functions	21 Sub Functions Under the ReadDTC Service	3 Sub Functions Under the ReadDTC Service
Supported Communication	CAN, LIN, and Ethernet	K-Line or CAN
ISO Standard	ISO 14229-1	ISO 14230
Error Memory Management	$14 (clearDiagnosticInformation) and $19 (readDTCInformation)	$14 (clearDiagnosticInformation), $18 (readDTCByStatus), $17 (readStatusOfDTC), and $12 (readFreezeFrameData)

Enhanced Diagnostic Capabilities: Unified diagnostic services present much more variant of diagnostic sub services with comparing KWP2000. This includes comprehensive services for accessing fault memory, manipulating ECU parameters, and executing complex diagnostic routines. Especially for the error management UDS has faster response and it is more efficient, in our case using CAN communication same diagnostic trouble code tried to analyze with using identical electronic control unit. Figure 14 shows that UDS can analyze same diagnostic trouble code with using only two diagnostic trouble code service ($14 and $19). However, KWP2000 needs four diagnostic trouble code services ($12, $14, $17, and $18) for same issue. Which means spending time to analyze for KWP2000 is much more.

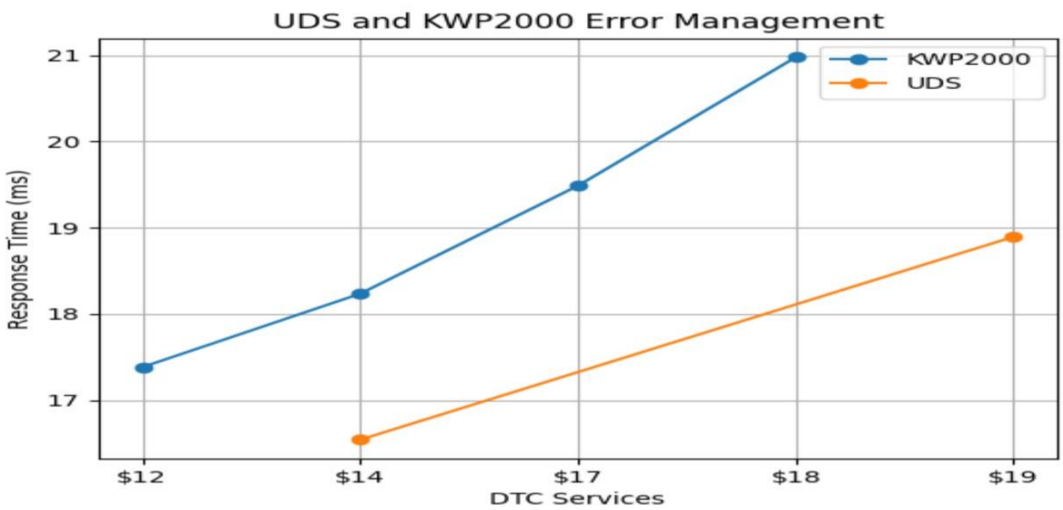

Figure 14: Diagnostic Trouble Code Services Performance

Advanced Security Features: Unified diagnostic services provide high level security mechanisms robustly. UDS also prevent unauthorized access to electronic control unit or hardware. This is critical benefit for manufacturers. With using this feature unauthorized workshops cannot be able to reach control units.

Standardization and Adoption: UDS always implies ISO 14229 standard. With this standardization approach makes UDS universal and communize protocol for all manufacturer and vehicle variants especially modern vehicles which needs complex and high-tech solutions.

In contrast, KWP2000, being an earlier protocol, and it has some limitations which prompt to manufacturer change the KWP2000 with UDS. KWP2000 was an ultimate solution for old generation architecture vehicles. However, day by day KWP2000 cannot fulfill the requirement of the new modern complex vehicle's technologies. The UDS protocol offers a benefit wherein the tester can solicit multiple measurement values through a single UDS service request using its 2-byte (16 bits) long data identifiers. In contrast, the KWP 2000 standard employs a 1-byte (8 bits) Local identifier. This results in enhanced efficiency in the exchange of data. When these protocols are tested under the same conditions and the CAN trace values are graphed, as can be seen from figure 15, the efficiency of UDS is dominant.

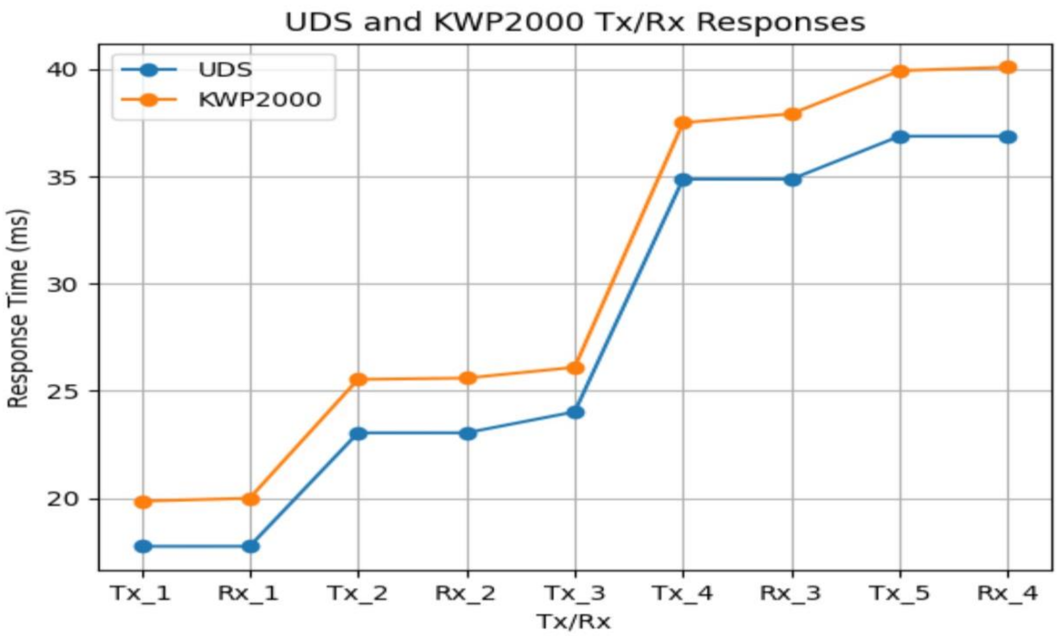

Figure 15: Response Times of UDS and KWP2000

When a connection is provided to vehicle, specific tool and software are mandatory to analyze the data which is earned by using these protocols. Figure 16 shows how important diagnostic tool is in communication and shows that it is not mandatory for services but also production line and research and development centers as well.

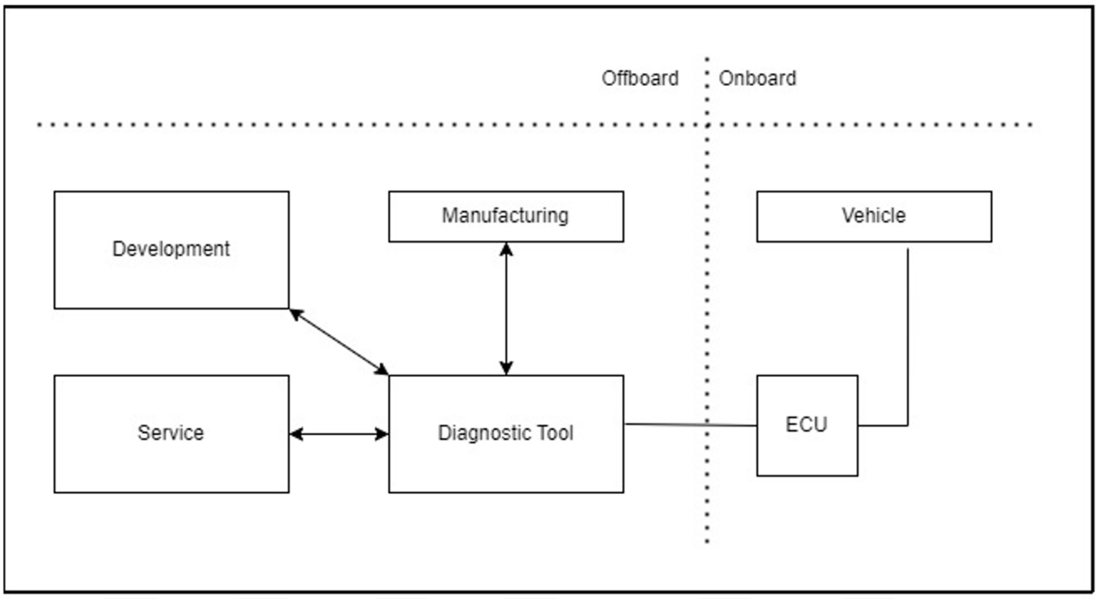

Figure 16: Diagnostic Diagram

With this academic work, technical background provided by giving analytical examples and after with providing identical environment. Unified diagnostic services and KWP200 protocols are tested. Test case include two main sections. These sections: firstly, to show transmit and received responses. This response time period data collected, and it was proven that UDS is faster by showing it into same graph for both two protocol responses. Secon section is cover about diagnostic trouble codes. In that test case, same virtual diagnostic trouble code created and triggered. After, analysis style and behavior examined separately. Second test cases showed that error memory management strategy of UDS is working efficient than KWP2000. This result also collected and showed in graph.

3.1.1 How Should UDS Based Diagnostic Design Based On AUTOSAR ?

AUTOSAR based unified diagnostic design implementation can be help with applying steps below.

1. Identify Diagnostic Requirements: The first step is to determine the diagnostic requirements for the system. The UDS protocol can be used when specifying diagnostic services such as fault detection, fault reporting and troubleshooting.

2. Understand AUTOSAR Diagnostics Stack: With understanding the AUTOSAR diagnostic stack, that will give access to the software modules that manage diagnostic communication and services. Relevant modules consist of Diagnostic Communication Manager (DCM), Diagnostic Event Manager (DEM), Diagnostic Service Manager (DSD) and Diagnostic Runtime Environment (DRE).

3. Configure Diagnostic Communication Manager (DCM): Enable USD communication by configuring the DCM module in AUTOSAR. Use the USD protocol as communication protocol to set up diagnostic services. Configure the communication arguments like communication transport platform like CAN, LIN, etc. And diagnostic addressing modes.

4. Implement Diagnostic Services: Enforce UDS diagnostic services as per your system's diagnostic necessities. UDS diagnostic services include Read Data by Identifier (RDI) Write Data by Identifier Control DTC Setting, among others. Furthermore, carry out the required service handlers.

5. Define Diagnostic Data: Specify the diagnostic data elements that need to be tracked, read, or changed during a diagnostic interaction. This incorporates the definition of diagnostic data identifiers (DIDs) and their associated data components. Connect DIDs and map them to your automotive system's appropriate internal variables or sensors.

6. Handle Diagnostic Event Reporting: It is also necessary to manage diagnostic event reporting, with usin the Diagnostic Event Manager (DEM). Design the DEM to create diagnostic events considering predefined rules and regulations.

7. Integrate with ECUs: Connect the UDS based diagnostic design to the Electronic Control Units (ECUs) in your vehicle system. Guarantee that the appropriate AUTOSAR communication interfaces, like Diagnostic Communication Interface (DCI), is set up and connected to the appropriate ECUs.

8. Test and Validate: Lead careful testing validation of the UDS-based diagnostic design. Conduct tests to make sure that services are working as expected.

9. Ensure Compliance: Verify that it provides the relevant automotive regulations and appropriate standards such as ISO 14229 (UDS) ISO 13400 (AUTOSAR).

With applying the steps given above in steps, it makes possible to successfully create a UDS-based diagnostic design. In addition, efficient and standardized diagnostic capabilities to automotive system will be provided.

CHAPTER IV
CONCLUSION

In conclusion, in automotive industry vehicles getting more complex electronics architecture. When architecture getting more complex, diagnostic systems getting crucial importance. Because, with using diagnostic communication root cause of the problem can detect without any dismounting of the hardware which is located in the vehicle. In that case, costs will reduce. Diagnostic communications also provide routine vehicle checks mechanism and with using that mechanism, it can take precautions which has possibility. These diagnostic systems connect to the vehicle with using a communication platform. That platform mostly created from by controller area network. In this thesis most common two diagnostic protocols analyzed and described in a detailed way. After these two common protocols tested with using identical CAN communication network. Given the increasing complexity of modern automotive E/E systems and the corresponding software, CAN communication has become integral in managing this complexity and detecting potential errors during development. During the test cases same electronic control unit used. Also, this control unit includes AUTOSAR compatible basic software and application software. Later, with using same diagnostic tool (tester) connected to the electronic control unit and started to send controller area network messages to the ECU and also starts to receive again controller area network messages from electronic control unit. In the controller area network trace section response times are observed and this time periods showed in a graph. Then, again with providing same conditions, a virtual diagnostic trouble code triggered after these two protocols are tested for their ability to analyze and how they managed. This error management process handled by their related diagnostic sub services and this error management result showed also within graph. As a result of the test and analysis results applied under equal conditions, the advantages of UDS were clearly stated.

REFERENCES

[1] DAEHYUN Kum, GWANG-MIN Park, SEONGHUN Lee and WOOYOUNG Jung (2008), "AUTOSAR Migration From Existing Automotive Software", *International Conference on Control, Automation and Systems*, pp. 558-562, Korea.

[2] EMBITEL (2018), *Understanding How ISO 26262 ASIL is Determined for Automotive Applications*, /https://www.embitel.com/blog/embedded-blog/understanding-how-iso-26262-asil-is-determined-for-automotive-applications, DoA.10.01.2024

[3] CHANDEL Ashutosh (2020), *Understanding an ASIL in the Functional Safety Standard ISO 26262*, https://www.lhpes.com/blog/what-is-an-asil, DoA.10.01.2024

[4] CIA (2011), *History of CAN technology*, https://www.can-cia.org/%20can-knowledge/can/can-history/, DoA.12.01.2024

[5] DEKANIC Stjepan, GRBIC Ratko, MARUNA Tomislav and KOLAK Igor, (2018), "Integration of CAN Bus Drivers and UDS on Aurix Platform", *Zooming Innovation in Consumer Technologies Conference*, pp. 39-42, Croatia.

[6] SALCIANU Monica and FOSALAU Cristian, (2012), "A new CAN Diagnostic Fault Simulator Based On UDS Protocol", *International Conference and Exposition on Electrical and Power Engineering*, pp. 820-824, Romania.

[7] CSS ELECTRONICS (2023), *OBD2 Explained - A Simple Intro*, https://www.csselectronics.com/pages/obd2-explained-simple-intro, DoA.12.01.2024

[8] EDMUNS Jeff (2019), *Interpreting and understanding DTC codes and DTC meanings*, https://gomotive.com/blog/dtc-codes/, DoA.12.01.2024

[9] KHARCHE Parag, MURALI Meera and KHOT Geetanjali, (2018), "UDS Implementation For ECU I/O Testing", *In, International Conference on Intelligent Transportation Engineering*, pp. 137-140, India.

[10] DIARRA Aboubacar (2013), *OSI Layers in Automotive Networks*, https://www.ieee802.org/1/files/public/docs2013/new-tsn-diarra-osi-layers-in-automotive-networks-0313-v01.pdf, DoA.10.01.2024

[11] BIN Sun, YUE-LI Hu and KUN Wang, (2011), "Diagnosis Design Based on KWP2000 in Body Control Module", *In, International Conference on Measuring Technology and Mechatronics Automation*, pp. 1071-1074, China.

[12] EMBITEL (2018), *KWP 2000 and UDS Protocols for Vehicle Diagnostics: An Analysis and Comparison*, https://www.embitel.com/blog/embedded-blog/kwp-2000-and-uds-protocols-for-vehicle-diagnostics-an-analysis-and-comparison, DoA.10.01.2024

[13] HONEKAMP Uwe (2009), "The Autosar XML Schema and Its Relevance for Autosar Tools", *Journal of IEEE Software*, Vol.26, no.4, pp. 73-76.

[14] SINGH Gaurav, KAMATH Narayan and SHARMA Rajendra Kumar, (2018), "Implementing Adaptive AUTOSAR Diagnostic Manager with Classic Diagnostics as APIs", *International Conference on Intelligent Computing and Control Systems*, pp. 894-898, India.

APPENDICES

APPENDIX A Sample C Code For Negative And Positive Responses

Here is an example C code for handling NRC and positive response of Read Data By Identifier (0x22) service in UDS. This code defines a “uds_request_t” structure to hold the request information, and a “uds_response_t” structure to hold the response information. The “handle_read_data_by_identifier” function takes a pointer to a request object as input, and returns a response object.

```
#include <stdio.h>
#include <stdlib.h>

#define NRC_SERVICE_NOT_SUPPORTED 0x11
#define NRC_CONDITIONS_NOT_CORRECT 0x22
#define NRC_REQUEST_OUT_OF_RANGE 0x31
typedef struct {
    uint8_t service_id;
    uint16_t data_id;
    uint16_t data_length;
} uds_request_t;

typedef struct {
    uint8_t service_id;
    uint8_t response_code;
    uint8_t data[8];
    uint8_t data_length;
} uds_response_t;
uds_response_t handle_read_data_by_identifier(uds_request_t* request) {
    uds_response_t response
```

```
    uint16_t data_id = request->data_id;

    // Check if the requested data identifier is supported
    if (data_id != 0x1234 && data_id != 0x5678) {
        response.service_id = request->service_id + 0x40; // Set response SID to negative
response SID
        response.response_code = NRC_REQUEST_OUT_OF_RANGE; // Set negative
response code
        response.data_length = 0;
        return response;
    }
    // Fetch the data corresponding to the requested data identifier
    uint8_t data[8];
    if (data_id == 0x1234) {
        data[0] = 0x01;
        data[1] = 0x02;
        data[2] = 0x03;
        data[3] = 0x04;
        data[4] = 0x05;
        data[5] = 0x06;
        data[6] = 0x07;
        data[7] = 0x08;
    } else {
        data[0] = 0xAA;
        data[1] = 0xBB;
        data[2] = 0xCC;
        data[3] = 0xDD;
        data[4] = 0xEE;
        data[5] = 0xFF;
        data[6] = 0x00;
        data[7] = 0x11;
    }

    // Construct positive response
```

```
    response.service_id = request->service_id;
    response.response_code = 0x00; // Positive response code
    response.data_length = 8;
    for (int i = 0; i < 8; i++) {
        response.data[i] = data[i];
    }

    return response;
}

int main() {
    uds_request_t request;
    request.service_id = 0x22;
    request.data_id = 0x1234;
    request.data_length = 0;

    uds_response_t response = handle_read_data_by_identifier(&request);
    if (response.response_code != 0x00) {
        printf("Negative Response: 0x%x\n", response.response_code);
    } else {
        printf("Data: ");
        for (int i = 0; i < response.data_length; i++) {
            printf("%x ", response.data[i]);
        }
        printf("\n");
    return 0; }
```

APPENDIX B Sample C Code For Service $27, NRC $33 and $35

Here is an example C code for implmentation of service $27, NRC $33 and $35.

```
if (0x27) {
    else if ( 0x27 ):
        if (received_data.received_data[2]) {
            else if (0x01):
                if (message_length == expected_message_length_security_2) {
                    responseCode = positive_response;
                } else {
                    responseCode = IMLOIF;}
                security_access_byte1 = received_data.received_data[3];
                security_access_byte2 = received_data.received_data[4];
                security_access_byte3 = received_data.received_data[5];
                security_access_byte4 = received_data.received_data[6];
                break;
            else if (0x02):
                if (message_length == expected_message_length_security) {
                    if ((received_data.received_data[3] == security_access_byte1) &&
                        (received_data.received_data[4] == security_access_byte2) &&
                        (received_data.received_data[5] == security_access_byte3) &&
                        (received_data.received_data[6] == security_access_byte4)) {
                        security_access_gained = true;
                        responseCode = positive_response;
                    } else {
                        security_access_gained = false;
                        responseCode = IK;
                        /* NRC 0x35: invalidKey */
                    }
                } else {
                    responseCode = IMLOIF;
                    /* NRC 0x13: incorrectMessageLengthOrInvalidFormat */
```

```
            }
            break;
        default:
            responseCode = ROOR;
            break;
    }
    break;
default:
    responseCode = SNS;
    /* NRC 0x11: serviceNotSupported */
    break;
}
```

APPENDIX C Sample C Code For AUTOSAR Microcontroller Drivers

This sample introduces a more modular structure with GPIO_ConfigType representing configuration parameters, GPIO_DriverType encapsulating the driver with its configuration, and standardized functions like GPIO_Init and GPIO_WritePin following MCAL principles. In a full AUTOSAR environment, this would be just a small part of the complete system, and additional layers such as Hardware Abstraction Layer (HAL) and services like communication stacks, diagnostics, and operating system services would be implemented. Always refer to the specific microcontroller's documentation and AUTOSAR standards for a complete and accurate implementation.

```
#include <stdint.h>

// AUTOSAR-like configuration
#define PORTA_BASE_ADDRESS 0x40004000
#define PORTB_BASE_ADDRESS 0x40005000
#define GPIO_PIN_0 (1U << 0)
#define GPIO_PIN_1 (1U << 1)
#define GPIO_PIN_2 (1U << 2)

// AUTOSAR-like data structures
typedef enum {
    GPIO_LOW = 0,
    GPIO_HIGH = 1
} GPIO_LevelType;

typedef struct {
    uint32_t portBaseAddress;
} GPIO_ConfigType;

typedef struct {
    GPIO_ConfigType *config;
} GPIO_DriverType;

// AUTOSAR-like functions
void GPIO_Init(GPIO_DriverType *driver) {
    // Initialization of GPIO (simplified for educational purposes)
    // Configure port based on the provided configuration
    // ...
    // Set initial pin states, directions, etc. }
```

```
void GPIO_WritePin(GPIO_DriverType *driver, uint32_t pin,
GPIO_LevelType level) {
    // Write to a specific GPIO pin
    // ...
}

int main() {
    // Example usage of AUTOSAR-like microcontroller driver
    GPIO_ConfigType configA = {PORTA_BASE_ADDRESS};
    GPIO_DriverType driverA = {&configA};

    GPIO_Init(&driverA);

    GPIO_WritePin(&driverA, GPIO_PIN_0, GPIO_HIGH);
    GPIO_WritePin(&driverA, GPIO_PIN_1, GPIO_LOW);

    // More complex interactions with the GPIO driver
    // ...

    return 0;
}
```