IJREAT International Journal of Research in Engineering & Advanced Technology, Volume 8, Issue 5, October - November 2020
ISSN: 2320 – 8791 (Impact Factor: 2.317)
www.ijreat.org

Advancement on User Data Protocol Network Management Module 4.2.2 for Ethernet Etiquette using AUTOSAR

Ms. S. G madhugiri
M.Tech Student, School of ECE
MIT World Peace University
Pune, Maharashtra India

Ms. S J. Joshi
Technical Lead,
KPIT Technologies,
Pune, Maharashtra India

Prof. Dr. S.B.Somani
M.Tech Coordinator,
MIT World Peace University
Pune, Maharashtra India

Abstract **– Electronic Control Unit (ECU) and the Embedded System used for a wide variety of purposes from vehicle control to entertainment in modern automotive industries. The industry tries to develop new and efficient systems periodically, to improve the performance of a system. As the number of ECUs in an automobile increases, new technologies are going to develop day by day to interconnect the ECUs. As a result of much efficient ECUs, the data produced by these ECUs and the data to be transmitted between these ECUs increased to a very large scale [1]. So the industry tried to select new connectivity solutions which provide higher bandwidth. The automotive embedded system industry thus ended up in using the well accepted and highly standardized Ethernet connectivity [1]. So that to manage the network between ECUs for Ethernet connectivity UDP Network Management is the best option. This paper shows how to develop the UDPNM Module with using AUTOSAR standards.**

Keywords **- Autosar, Ethernet, User datagram Protocol, Network Management, Udp Network Management, ECU.**

I. INTRODUCTION

The percentage of electronic components and embedded systems is increasing day by day. They are now an essential part of the automotive industry. Due to that, the number of ECUs is increases. In the past, there were different types of connectivity technologies such as Can, Lin, MOST, Flex Ray, etc. used to connect different ECUs in automobiles. But they are not enough for high speed of communication, so Ethernet is used for high speed communication. Ethernet not only provides high bandwidth and scalability, but is also a recognized and mature standard in the electronics industry. Ethernet has the added advantage of being a universal connectivity solution to interconnect almost all electronic systems in the automobile [1]. This makes the automotive system complex in many ways, including networking
For large-scale Ethernet interconnections, we need a network management protocol.

In this paper we are developing network related solutions for Ethernet protocols. Autosar is the standard for developing automobile software. So we are developing UDP network management module using the Autosar standard.

The UDP Network Management (UDPNM) function provides conversions within the Network Management Interface (NM) and a TCP / IP stack (TCP / IP) [3]. The main aim to develop UDPNM is to increase the power efficiency of automobile communications.

II. LITERATURE SURVEY

A. Why AUTOSAR?

With the rapid development of technologies and innovations such as autonomous driving, advance driver assessment system and next-generation hybrid and electric cars, there are a lot of new technologies that are being implemented in vehicles. There was the need of a smart software architecture that needs to be followed by an automobile industry to make their vehicles smarter, safer and intelligent. So, AUTOSAR is a smart software architecture that was designed [8].

Now, a car is having multiple functionalities like Infotainment, wipers, airbags, and electronic injection system, so there is a need of individual ECU for different functionalities. Previously an OEM has to buy all the devices needed for the proper functionality of its car by only one vendor. If the OEM's want an additional functionality, then the OEM's have to be depend only on one particular service provider. That service provider may charge the OEM high amount.

But with the development of AUTOSAR Software Architecture, there is no need to depend on the individual service provider. An OEM's can get different functionalities from different service providers and an AUTOSAR stack makes every ECU compatible with the other ECU, provides a common platform. Integration can also be possible with the help of AUTOSAR. Like one BSW module can be taken

from one vendor other module from other vendor like this. All the modules can be given to one Service Provider for Integrating [6].

B. Why Ethernet?

In paper [4] it shows the comparison of different communication protocol like CAN, MOST, FlexRay and Ethernet. These all protocols are used as in-vehicle network (IVN) protocols. Also in paper [4] it gives the advantages and disadvantage, where it has been used in an automobile industry is explained. So from that Ethernet can provide the required bandwidth, and it has several advantages, including low cost, high bandwidth, and mature technology [10]. Therefore, Ethernet is expected to become one of the dominant protocols for IVNs in the future. Ethernet can be used for diagnosis, multimedia, and infotainment, and it is expected to be used for ADASs and backbone networks in future automotive systems. CAN, FlexRay, and Ethernet will be simultaneously used in the same vehicle [4]

C. Why UDP?

The User Datagram Protocol (UDP) is a transport layer protocol in OSI model. It provides a best-effort datagram service to an end system. UDP can send the data without acknowledgment of a receiver so data loss is possible, but it is used over a TCP. In the paper [5] it discusses about the TCP and UDP Throughput and it shows that the UDP generating almost double the throughput as TCP, the simplicity of UDP reduces overhead from the protocol and can be adequate for some applications [5].

D. Why Network Management?

Network management is the process of administering and managing the networks. Network Management includes performance management, fault analysis and maintaining the quality of service. In the paper [7] gives the review of why Network Management is required in every communication. Network Management module is to ensure the safety and the reliability of a communication network for ECUs. Have the saving of energy by keeping the network active whenever it is required.

III. AUTOSAR LAYERED ARCHITECTURE

AUTOSAR (AUTomotive Open System ARchitecture) is a software layered architecture developed by automotive manufacturers (OEM's), tool developers. It is open system architecture because it is unlicensed. It separates Hardware from software. So if any changes are made in the hardware there is no need to change the software part. It will add additional functionalities without disturbing the existing model. AUTOSAR architecture follows top-down approach [2]. AUTOSAR layered architecture consists of the Application layer, RTE, BSW, Microcontroller. Figure 1 shows Autosar Layered Architecture.

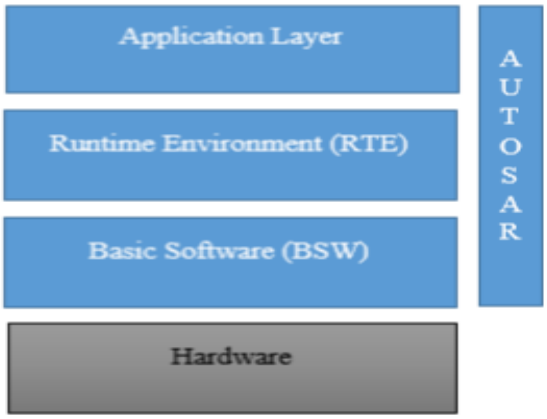

Fig.1. AUTOSAR Layered Architecture.

A. Basic Software layer

It is also known as BSW. It provides services to the software components to run the functional part of the software [8]. Figure 2 shows BSW consisting three software layers. It consists of three layers:

- Service layer
- ECU abstraction layer
- Microcontroller Abstraction layer

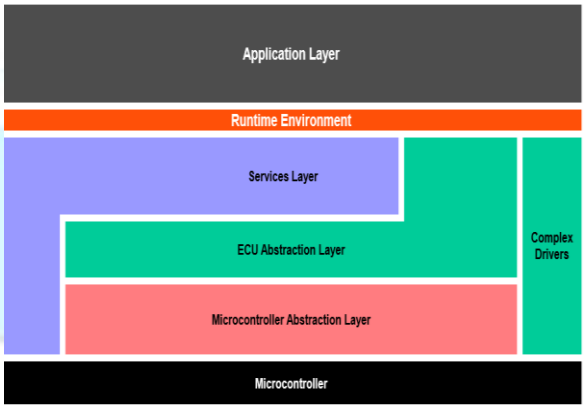

Fig. 2 Autosar BSW Layer comprising of three basic software layers

In AUTOSAR, each layer of BSW is further divided into different functionalities like service layer is divided into memory services, communication services and system services. Figure 3 describes the communication stack. We will concentrate on where the communication stack is present in BSW and how UDPNM is comprised in this COM stack.

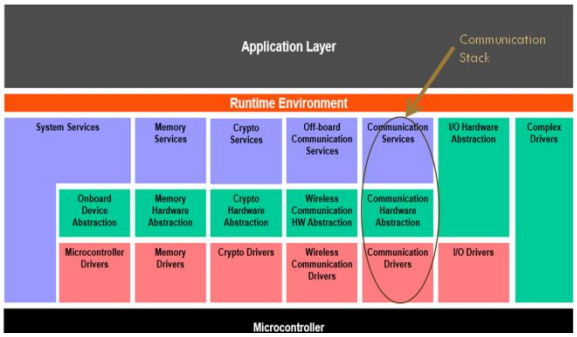

Fig. 3: Communication Stack.

IV. UDP NETEWORK MANAGEMENT

User Datagram Protocol Network Management, it is basically used for managing the network for Ethernet. It is lay on the BSW Service layered of Autosar Layered Architecture.

AUTOSAR UDP Network Management (UdpNm) is intended to work together with a TCP/IP Stack, independent of the physical layer of the communication system used. The AUTOSAR UDP Network Management is a hardware independent protocol that can be used on TCP/IP based systems. Its main purpose is to coordinate the transition between normal operation and a bus-sleep mode of the network. The UDP Network Management (UdpNm) function provides an adaptation between Network Management Interface (NM) and a TCP/IP Stack (TCP/IP) [3]. Figure 4 shows the extended version of AUTOSAR Communication Stack, where UDPNM lie.

User Datagram Protocol Network Management Module lay in the Service Layer of AUTOSAR Layered Architecture.

In a service layer, Ethernet stack is made up of communication services, communication hardware abstraction and communication drivers. In communication services, some of the modules are listed like Communication Manager, Network Manger, UDP NM Module, Socket Adaptor, PDU router and TCP/IP stack. In communication hardware abstraction, Ethernet Interface, Ethernet switch drivers and Ethernet Transceiver drivers are there [3]. In Communication drivers, there is Ethernet Controllers.

Now we will study about these modules one by one.

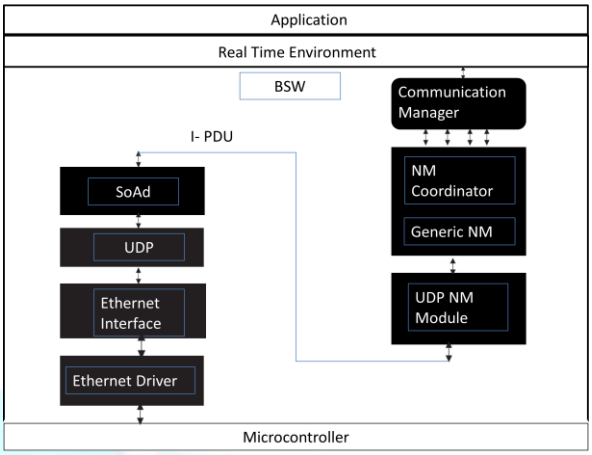

Fig. 4 AUTOSAR Communication Stack for UdpNm.

A. Communication Manager (ComM)

Communication Service having Communication Manager (ComM) it is a Resource Manager. The ComM module collects the bus communication access requests from communication requestors and coordinates all access requests. It simplifies the usage of the bus communication stack for the user. This includes a simplified network management handling. The ComM module uses the NM to synchronize the control of communication capabilities across the network (synchronous start-up and shutdown) [9].

The data is transmitted from Communication Manager to the NM module which contains the Generic NM and NM Coordinator.

B. Generic NM

It is an adaption layer between ComM and Bus specific NM. It is independent of protocols. It handles 'startup' and shutdown of Node/Network, whenever the network is not required any more per network configuration.

C. NM Coordinator

It provides the Coordination between the different Bus Specific Protocols. When the NM-PDU is transmitted through UDPNM, UDPNM uses the Coordination Algorithm to send the periodic NM PDU to the lower layer.

D. Socket Adapter Layer (SoAd)

The Lower Layer is SoAd which is connected through I-PDU (Interaction Layer Protocol Data Unit) to the UdpNm. The main purpose of socket adaptor is to create an interface between communication service layer using PDU's and a socket based TCP/IP stack. So, SoAd (Socket Adaptor) maps PDU id's to socket connections and also Autosar API's to Socket API's.

SoAd supports both TCP and UDP. It can again establish the TCP connection, once it is lost.

E. UDP

Then the data is transmitted to the Ethernet Stack, where the UDP protocol is used for communication. It does not establish any connection before the transmission of data, so it uses a minimum of protocol mechanism, the packet of data is transmitted very fast to the IP and then to the Ethernet Interface. The packet of data send to the Ethernet Interface (EthIf) is called as a Datagram. When there is need to transfer a huge amount of data in real time, UDP is used. When speed is concern then UDP is preferred.

F. Ethernet Interface (EthIf)

It belongs to the ECU abstraction layer. It abstracts the upper layers from hardware. Ethernet Interface remains same for every Ethernet controller and Ethernet Transceiver.

G. Ethernet Drivers

It abstract Eth Controllers (Hardware) from Eth Interface. It provides the data to the Microcontroller.

The AUTOSAR UdpNm module having three operational modes

- Network Mode
- Prepare BusSleep Mode
- BusSleep Mode

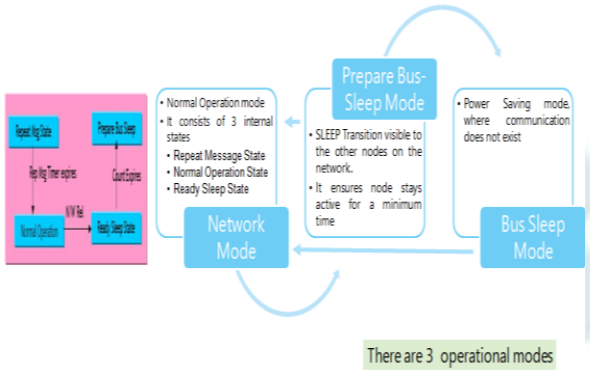

Fig. 5. Flow diagram of UdpNm Operational mode

V. IMPLIMENTAION

Steps for developing the UdpNm module:

1) Analyze UDPNM specific requirements from AUTOSAR UDPNM SRS and Module Specification of UDP Network Management.
2) Develop Embedded Code and Code Generator for the UDPNM.
3) Perform QAC and Polyspace analysis on Embedded Code for compliance to MISRA 2012 standard
4) Develop Test Applications from the Functional Test Plan
5) Execute Functional Test Applications on x86 based PC and generate Functional Test Report.
6) Execute Unit Test Scripts on x86 based PC and generate Unit Test Report
7) Writing the test cases for Unit testing in RTRT Tool.
8) Execute Functional Test Applications on MPC 5748G and generate Porting System Test Report.
9) Writing the test cases for testing the generation tool code.
10) Execute Generation Tool code scripts to generate Generation Tool code Report.

VI. WORK FLOW

The figure 6 shows the flow diagram of development of the UdpNm Module.

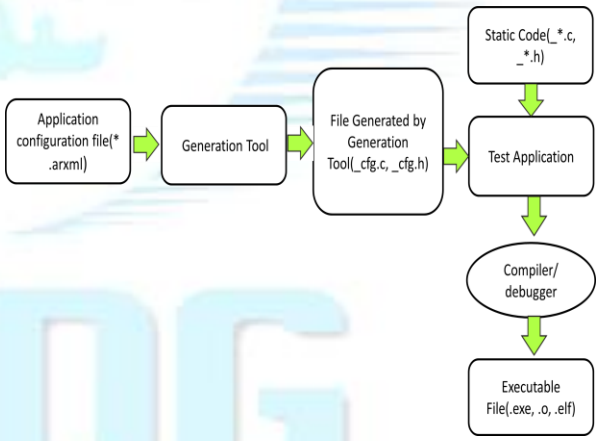

Fig. 6: Work flow to develop the UDPNM Module

A. Application configuration file (.arxml)*

User can load the Custom Parameter definition File or AUTOSAR Parameter Definition File into a K-SAR tool to configure desired configure set(s) for UdpNm module.

B. Generation Tool

Generation Tool will be a command line tool that will parse the ECU Configuration Parameter Definition File, ECU Configuration Description File(s), BSWMD

file and Module Template file as input and generates the C Source and C Header files specific to a module.

C. File generated by GT

Generation tool generates .c and .h files specific to a module. It also generates S-Record File if module is implemented as POST-BUILD variant.

D. Static code

Module specific standard SWS API's definition will be present in static code. Tool generated files will be inputs to static code.

E. Application Mak file

Make file contains resources like the include paths or the name of the basic software bundle; Provide the required variables and commands that are used to build the module together with the application.

F. Compiler

It refers to processing of the source file and converting them to object file. The process corresponds to creating Machine language instruction from the high level language source code.

G. Linker

In this phase, linker creates executable file from multiple object files.

H. Executable file

An executable file is ready to be loaded into memory and executed by the system. For Ex: *.abs, *.o or *.exe

VII. CONFIGURATION OF MODULE

For configuring the UDPNM Module we are using the Compose4Ksar tool (C4K)

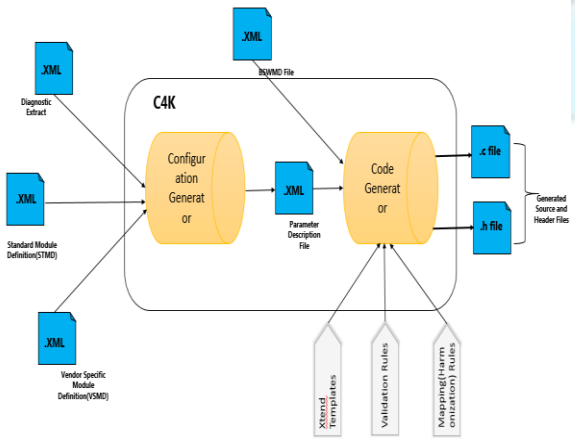

Fig. 7: Code Generator Workflow

A. Input Files

The UdpNm Code Generator accepts ECU Configuration Description File(s), Parameter Definition File, and a BSWMD file as input. Code Generator additionally includes implementation of Configuration validation and Mapping (Harmonization) Rules. Configuration validation is based on explicit/implicit constraints defined in UdpNm Specification. UdpNm module can be automatically configured using C4k with an available System/Ecu/Diagnostic extract which in turn uses Mapping Rules implemented as part of Code Generator. Alternately C4k can be used to manually configure the UdpNm Module.

B. Output File:

UdpNm.exe will generate UdpNm_Cfg.h and UdpNm_Cfg.c files.

We have UdpNm bundles and features but to run it properly, we import bundles and features of dependent modules (ECUC and ComM). (We need to integrate bundles and features of specific modules.)

We have different functional API.arxml files in the test data folder, where we can configure different .arxml files as needed.

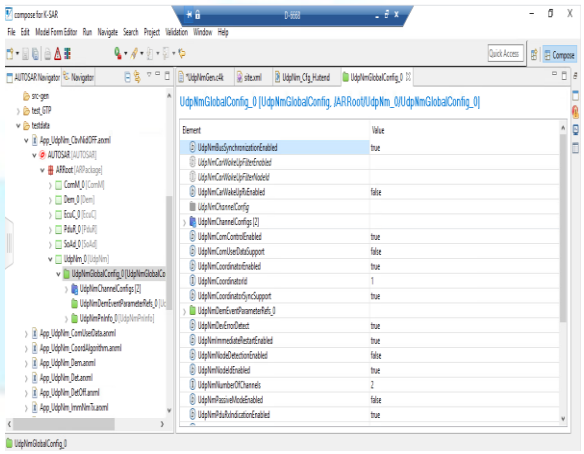

Fig. 8: A sample .arxml file and configuration

After integration we run the .arxml file to check if there is any validity or configuration error. We write validity rules to check the intent. If there are no errors in the configuration after running the file, no error or warning is shown in the console window and the config.c and config.h output files are generated.

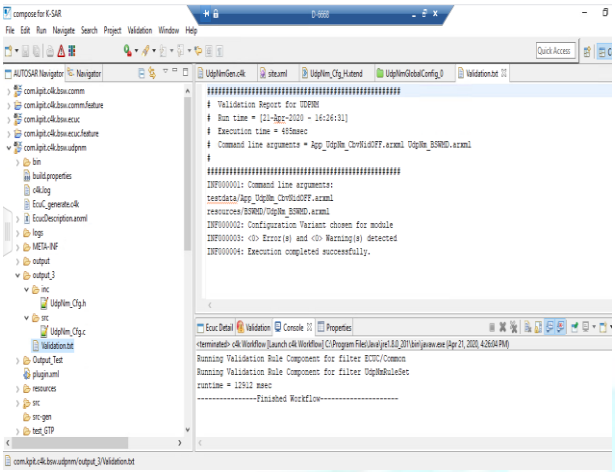

Fig. 9: Configuration output file

Then the path of the c4k will be provided to make file to run the functional test cases and generation tool code testing.

VIII. RESULTS

A. *Functional Testing*

In functional testing, functional test cases are written for each functionality according to the SRS document. These functional test cases then run on the x86 base system and MPC5748G hardware and give the porting result. Here are the porting results on x86 and MPC5748G

Test Application Name	Test Application Id	PC Variant	No Of Config Set(s)	Total Test Cases	Total Pass Test Cases	Total Fa Cas
App_UdpNm_Det	1	PC	1	111	111	0
App_UdpNm_ImmNmTx	2	PC	1	7	7	0
App_UdpNm_OpModes	3	PC	1	37	37	0
App_UdpNm_StateChangeDis	4	PC	1	1	1	0
App_UdpNm_Init	5	PC	1	10	10	0
App_UdpNm_PartialNetwork	6	PC	1	26	26	0
App_UdpNm_InfoServices	7	PC	1	29	29	0
App_UdpNm_Dem	8	PC	1	2	2	0
App_UdpNm_PassiveModeEnb	9	PC	1	3	3	0
App_UdpNm_TxRx	11	PC	1	55	55	0
App_UdpNm_CoordAlgorithm	12	PC	1	11	11	0
App_UdpNm_Offset	13	PC	1	4	4	0
App_UdpNm_CbvNidOFF	14	PC	1	2	2	0
App_UdpNm_DetOff	16	PC	1	6	6	0
App_UdpNm_ComUserData	17	PC	1	4	4	0
Total Test Cases Tested				308	308	0

Fig. 10: Porting Report of X86

AUTOSAR Module	Compiler Name	Processor Name	No. of Application(s)		
UdpNm	GHS	MPC5748G	16		
Application Name	No. of Compiler Error(s)	No. of Compiler Warning(s)	No. of Generation Error(s)	No. of Generation Warning(s)	Build Status
App_UdpNm_Throughput	0	0	0	0	PASS
App_UdpNm_Det	0	0	0	0	PASS
App_UdpNm_ImmNmTx	0	0	0	0	PASS
App_UdpNm_OpModes	0	0	0	0	PASS
App_UdpNm_StateChangeDis	0	0	0	0	PASS
App_UdpNm_Init	0	0	0	0	PASS
App_UdpNm_PartialNetwork	0	0	0	0	PASS
App_UdpNm_InfoServices	0	0	0	0	PASS
App_UdpNm_Dem	0	0	0	0	PASS
App_Udpnm_PassiveModeEnb	0	0	0	0	PASS
App_UdpNm_TxRx	0	0	0	0	PASS
App_UdpNm_CoordAlgorithm	0	0	0	0	PASS
App_UdpNm_Offset	0	0	0	0	PASS
App_UdpNm_CbvNidOFF	0	0	0	0	PASS
App_UdpNm_DetOff	0	0	0	0	PASS
App_UdpNm_ComUserData	0	0	0	0	PASS
Total Error(s) / Warning(s)	0	0	0	0	SUCCESSFUL

Fig. 11: ESTR Report of MPC5748G

B. *Unit Testing*

Unit testing is used to test the embedded code, which we wrote while developing the module. Here every function, structures, loops and switch statement are examined. To check the intent we wrote the test cases in IBM Rational Test Real Time Studio (RTRT) tool and after running we got the coverage report.

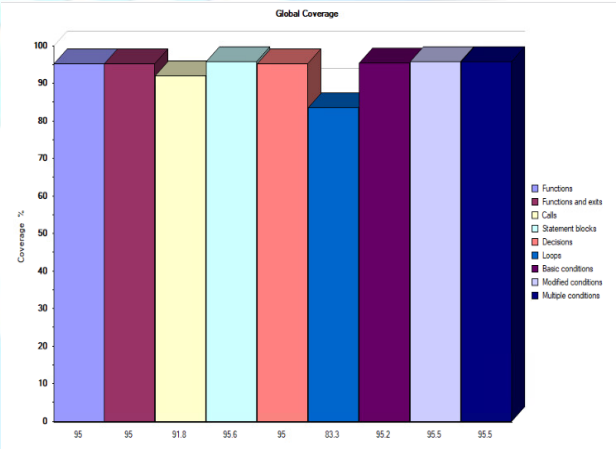

Fig. 12: Graphical Coverage Report

C. *Generation Tool Code Testing*

Generation tool code testing is used to test the validation rules which we write in c4k tool. We write the test cases for each validation rules like for checking the errors, warning and information. After running different configuration files the particular validation error will be appeared.

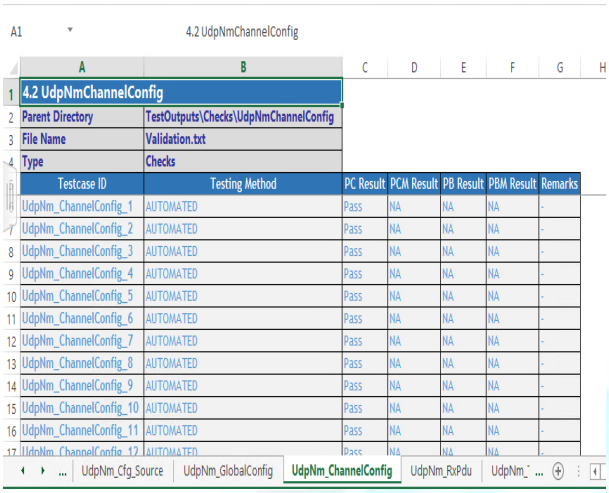

Fig. 13: Generation Tool Code Report

IX. CONCLUSION AND FUTURE SCOPE

So after completing the development of UDPNM Module we can conclude that the UDPNM module for Autosar version 4.2.2 is stabilized now and the complexity of the embedded code is reduced. After this development the network will properly manage for Ethernet protocol. And the efficiency of power is increases by configuring UdpNm Module into Bus Sleep Mode.

As the Autosar is a growing platform, it will identify the technological trends and key features are added accordingly. So the new AUTOSAR version release may happen with more features and functionality. It will enhance the Ethernet support in Automobile communication. So the development of UDPNM module for Autosar latest version is required. The next development will be for Autosar 4.3.1 release version.

ACKNOWLEDGMENT

The work is presented in this paper is supported by KPIT Technologies, Pune and School of ECE MIT World Peace University, Pune, Maharashtra, India

REFERENCE

[1] Gopu G.L., Kavitha K.V. and James Joy, "Service Oriented Architecture based connectivity of Automotive ECUs," International Conference on Circuit, Power and Computing Technologies [ICCPCT] - 2016

[2] AUTOSAR Layered Software Architecture

[3] Specification of UDP Network Management AUTOSAR Release 4.2.2

[4] Jin Ho Kim, Suk-Hyun Seo, Nguyen-Tien Hai, Bo Mu Cheon, Young Seo Lee, and Jae Wook Jeon, Member, "Gateway Framework for In-Vehicle Networks Based on CAN, FlexRay, and Ethernet," IEEE Transactions on vehicular technology, vol. 64, no. 10, October 2015

[5] Sangrok Han and Hyogon Kim, "On AUTOSAR TCP/IP Performance in In-Vehicle Network Environments," IEEE Communications Magazine • December 2016

[6] www.kpit.com/ECODE.

[7] Ratanang Thupae , Bassey Isong , Naison Gasela, Adnan M. Abu-Mahfouz, "Software Defined Wireless Sensor Networks Management and Security Challenges: A Review," IECON 2018

[8] AUTOSAR Technical overview v2.2.2 R3.1

[9] Specification of ComM Manager AUTOSAR Release 4.3.1

[10] Young Seo Lee, Jin Ho Kim, Seok Jin Jang and Jae Wook Jeon, "Automotive ECU Software Reprogramming Method Based on Ethernet Backbone Network to Save Time," IEEE Transactions on C.2.2 [Computer-Communication Networks]: Network Protocols- 2015