International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169
Volume: 3 Issue: 3 1255– 1260

# AUTOSAR 3.2 based Protocol Data Unit Router Module

Shilpa Das
Department of ECE
Vidya Academy of Science and Technology, Technical Campus
Trivandrum, India
*e-mail: shilpagiresh@gmail.com*

***Abstract***— Automotive Open System Architecture (AUTOSAR) is the new standard software architecture for automobiles. The increased usage of Electronic Control Units (ECU) in automobiles is contributing to higher software complexity in vehicles. Here is the significance of AUTOSAR. The AUTOSAR is a new framework used to reduce the ECU complexity in new generation automobiles. Its core frame work consists of four layers; the top layer is the Application layer and the bottom layer is the Microcontroller layer. Run Time Environment and Basic Software Layer are the two layers sandwiched between these two layers. The Basic Software Layer provides an infrastructural functionality to an ECU. The scope of this paper is limited to a module within the Basic Software Layer named Protocol Data Unit (PDU) Router module. PDU Router module must be instantiated in every ECU. The main task of this module is the routing of Interaction Layer Protocol Data Units (I-PDUs) between the communication services and the hardware abstraction layer modules. It also provides the gateway functionality which means the routing of I-PDUs between the same or different networks. This paper focuses the design, development and testing of AUTOSAR 3.2 based PDU Router module.

***Keywords-*** *AUTOSAR, ECU, PDU Router, I-PDU*

*****

## I. INTRODUCTION

The use of embedded software in automotive industry is growing by each day. In an automotive vehicle of modern day, most of the functionalities such as air conditioning, engine management, antilock braking system etc. are under computer control.

So, off late, modern vehicles are characterized by the presence of high number of ECUs (Electronic Control Units). Luxury cars can easily have up to 70 ECUs. The high number of ECUs will increase the software complexity of the vehicles. In order to handle this enormous complexity in automotive software systems, a new innovative standard was introduced called AUTOSAR (AUTomotive Open System Architecture). AUTOSAR was founded as a development partnership in 2003 by a group of companies.

AUTOSAR is a standardized architecture for Electric and Electronic (E&E) systems. The re-use of software components between different vehicle platforms, like OEMs (Original Equipment Manufacturer) is one of the major goals of AUTOSAR [1]. There are a number of technical factors that have motivated the development of AUTOSAR. The most significant of those are listed below:

- Improving scalability
- Improving quality and reliability
- Improving flexibility
- Enabling the early detection of errors during project's design phase.
- Consideration of availability and safety requirements
- Software updates and upgrades over vehicle life time
- Redundancy activation
- Reduction of costs for software development and service in the long term [2].

AUTOSAR supports the modular development of ECU software. The functions inside the ECU network are easier to integrate and replace. It helps the vehicle manufactures to be more flexible in the use of ECU software [8].

## II. SOFTWARE ARCHITECTURE

The AUTOSAR software architecture is a layered architecture. That has the goal to abstract the software from hardware components [3]. The software architecture of AUTOSAR shown in figure 1 consists of four main layers. The lowermost layer is the microcontroller layer which contains the ECU hardware. The next two layers above the microcontroller layer are Basic Software and the Run Time Environment layer. The uppermost layer is the application layer; which contains all the application specific hardware components [4].

The layers in the AUTOSAR have different functionalities. The Basic Software and the Run Time Environment layers are responsible for the abstraction between the hardware and the application software. The Basic Software is the standardized software; providing necessary services for running the functional part of the software. The Run Time Environment layer connects the basic software with the application software. It enables inter component communication as well as communication from software components to basic software modules.

The Basic Software layer includes three layers called the Service layer, the ECU abstraction layer and the microcontroller layer. It may also contain the Complex Device Driver. It breaks the layered architecture where the direct access of hardware is needed.

The MCAL (Microcontroller Abstraction Layer) is the abstraction of the hardware; to avoid the direct access of microcontroller's register. Abstraction layer provides software interface to a specific ECU. Service layer is the highest layer of the Basic Software and it provides basic

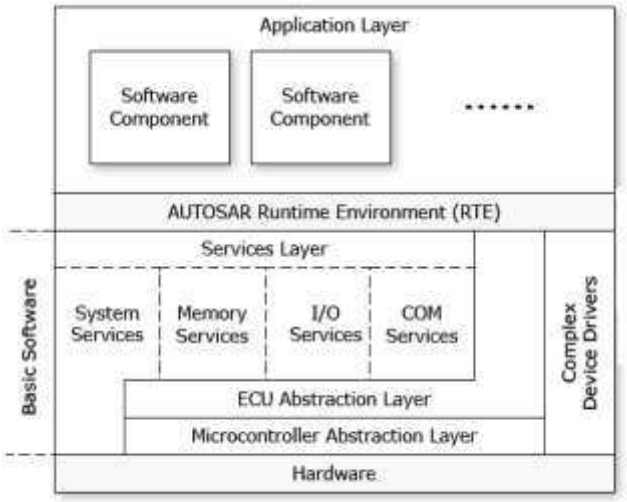


Figure 1. AUTOSAR software architecture layers [4].

services to the application layer, the basic services includes memory services, diagnostic services, communication services, system services etc. [3]. Protocol Data Unit (PDU) Router module is a part of AUTOSAR service layer. PDU Router module provides communication services to other modules [6].

The Operating System of AUTOSAR is based on standard OSEK based operating system. This operating system is widely used by the automotive industry. AUTOSAR OS supports fixed priority based scheduling, facilities for handling interrupts and it also provides inter task communication [5].

## III. PROTOCOL DATA UNIT ROUTER MODULE

PDU is an abbreviation of Protocol Data Unit. Each PDU contains an SDU and a PCI. SDU is Service Data Unit which is the data passed by an upper layer with the request to transmit the data. PCI is an abbreviation of Protocol Control Information; it contains the source and target information. It is added on transmission side and removed on receiving side.

The signals from the AUTOSAR application layer come to the RTE layer; the RTE performs the transformation from the AUTOSAR signal to the signal in COM. AUTOSAR signals are packed and unpacked by using the COM module. The signals are packed into I-PDUs. I-PDU group is a collection of I-PDUs in COM and contains more I-PDUs [1].

The PDU Router module is an important part of AUTOSAR ECU. The main task of Protocol Data Unit Router module is to route the I-PDUs (Interaction Layer Protocol Data Units) between:

- Communication interface modules such as LINIF,CANIF and FlexRayIf
- Transport protocol modules such as CAN TP,FlexRay TP
- AUTOSAR Diagnostic Communication Manager (DCM) and Transport Protocol modules such as CAN TP, FlexRay TP
- AUTOSAR COM and communication interface modules such as LINIF, CANIF or FlexRayIf or I-PDU multiplexer.
- I-PDU multiplexer and communication interface modules such as LINIF, CANIF or FlexRayIf.

PDU Router module can be adapted for gateway operations and for internal routing purposes. It transfers I-PDUs from one module to another module [6].

Each PDU has a static PDU ID; which is used to identify the PDUs. Each PDU Router module finds out the destination path by using the static configuration table and the PDU ID. I-PDUs has a maximum length of 8 bytes.

PDU Router module provides APIs for modules below and above the PDU Router. The modules below the PDU Router module are communication interface modules and the transport protocol modules. The modules above the PDU Router module are COM and DCM (Diagnostic Communication Manager). More over PDU Router module provides an interface to I-PDU Multiplexer (I-PduM) which is a module that is beside the PDU Router.

The PDU Router module consists of PDU Router routing tables and the PDU routing engine. The PDU routing tables can be updated during the post build time in the programming state of the ECU, and it describes the routing attributes for each PDU shall be routed. PDU routing engine describes the actual code performing the routing actions is based on the PDU Router routing tables. The detailed PDU structure is shown in figure 3.

The PDU Router engine provides a minimum routing capability to route specific PDUs without using the routing tables. This supports accessing of the DCM for activation of the ECU boot loader even when the post build time routing table is modified.

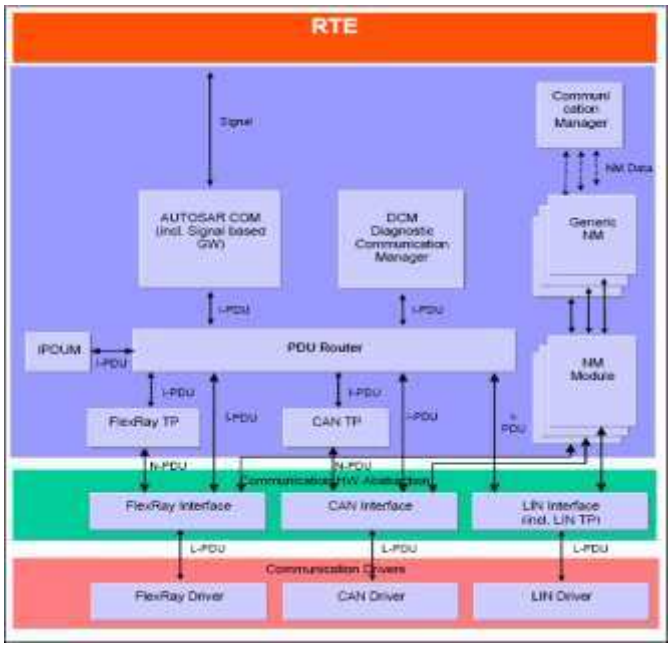


Figure 2. AUTOSAR Communication structure [6]

 ISSN: 2321-8169
1255– 1260

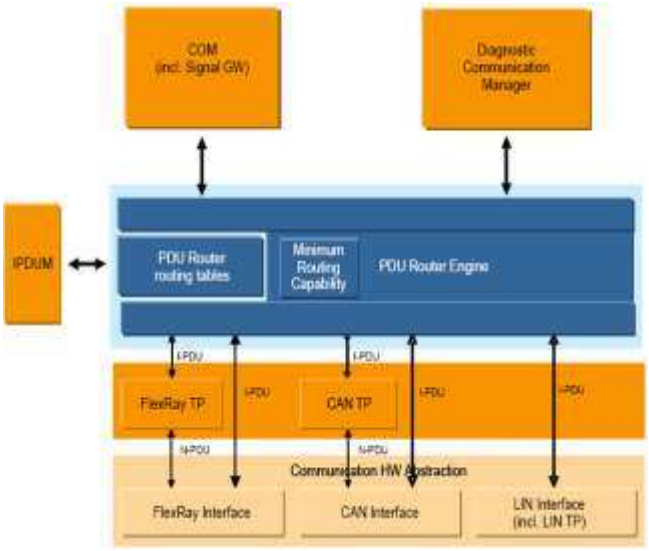


Figure 3. The detailed PDU structure [6]

### A. *Functional Specification of PDU Router*

The PDU Router module can perform three different classes of operations. They are PDU reception, receive the I-PDU and send to the upper layer, PDU transmission; transmit I-PDUs on request of upper layer modules and PDU gateway, receive the I-PDU from an lower layer module and transmit this I-PDUs to the same or another lower layer module [1].

The PDU Router module will refresh the routing tables only when they are not in use. The behavior of PDU Router module functions should be synchronous; although the overall action of a function might be asynchronous.

The PDU Router module provides a special type of gateway functionality known as routing-on-the-fly, which means that the forwarding of data between two communication modules is started before all data have been received.

PDU Router module transfers the I-PDU to one destination module (single cast transmission) or more than one destination modules (multicast transmission).The modules above the PDU Router module such as COM and DCM, can request the cancellation of an I-PDU transmitted from the lower layer modules, if that I-PDU is not needed ,it can be done by using the API PduR_<Lo>CancelTransmit. Cancellation of an I-PDU will only be confirmed by the destination protocol module. Like cancel transmission the modules can request for cancel reception also, for example if the upper layer module want to cancel the reception of I-PDUs from the lower layer modules then the upper layer modules will use the API PduR_<Up>CancelReceive..

It is also possible for the upper layer modules can change the transport protocol parameters for a definite Rx I-PDU.For this upper layer modules call the API PduR_<Up>ChangeParameter [6].

### B. *PDU Router State Management*

The PDU Router module consists of three states. They are PDUR_UNINIT, PDUR_REDUCED, and PDUR_ONLINE. After power up the PDU Router module shall be in the state PDUR_UNINIT. The PDU Router module shall change to the state PDUR_ONLINE, when the PDU Router has successfully initialized via the function PduR_Init. If the initialization did not succeed the PDU Router module shall change the state to PDUR_REDUCED.PDU Router states is shown in figure 4.

PDU Router module shall perform routing when it is in online state (PDUR_ONLINE), PDU Router shall perform minimum routing and routing when it is in the reduced state (PDUR_REDUCED) and the PDUR shall perform no routing when it is in the uninitialized state (PDUR_UNINIT) [6].

### C. *API Specification of PDU Router module*

In order to initialize the PDU Router module PduR_Init API is used. If this API is successfully initialized then the PDU Router module shall go to the state PDUR_ONLINE.A set of routing paths can be enabled and disabled by using the APIs PduR_EnableRouting and PduR_DisableRouting respectively [6].

### D. *Functional Definitions for lower layer modules*

The modules below the PDU Router modules are CAN,LIN, and FlexRay.Lower interface layer modules calls the function PduR_<Lo>IfRxIndiaction, after an L-PDU has been received. L-PDU is a Data Link Layer Protocol Data Unit which is assembled or disassembled in AUTOSAR hardware abstraction layer. In AUTOSAR; Data Link layer is correspondent to the Microcontroller Abstraction Layer. The maximum length of CAN and LIN L-PDU is 8 bytes and the maximum length for FlexRay L-PDU is 254 bytes.

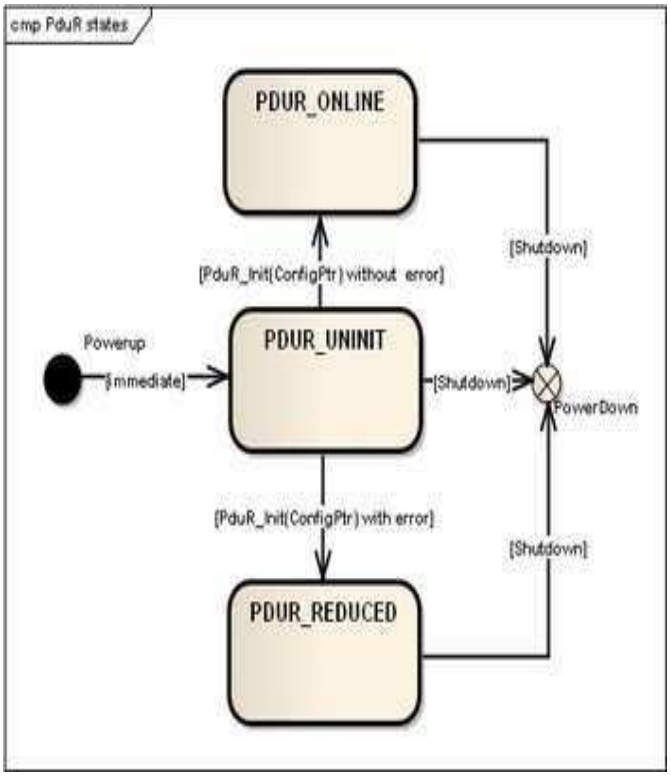


Figure 4. PDUR States

 ISSN: 2321-8169
1255– 1260

PduR_<Lo>IfTxConfirmation and PduR_<Lo>TpTxConfirmation functions are called by the lower layer interface and Transport protocol modules after the PDU has been transmitted. PduR_<Lo>TpProvideTxBuffer , API is used to provide the Tx buffer for the lower layer modules.

NM (Network Management) interface is a new added feature of AUTOSAR 3.2 version. NM module shall act as a bus independent adaptation layer between the bus specific Network Management modules such as CanNm and FrNm. PduR_<Can,Fr>NmRxIndication is used, after an L-PDU has been received. PduR_<Can,Fr>NmTxConfirmation is called after the PDU has been transmitted. In order to trigger the transmission of an NM message PduR_<Can,Fr>NmTriggerTransmit API is used.

### *E. Functional Definitions for Upper layer modules*

The modules above the PDU Router modules are COM and DCM. PduR_<Up>Transmit is called by the upper layer modules to request transmission .PduR_DcmCancelReceive API is used to terminate the currently ongoing data reception. PduR_DcmCancelTransmit is used to terminate the currently ongoing TP data transmission. PDUR shall provide a communication interface API for the CDD.

PduR_CddTransmit is called by complex device driver to request a transmission. PduR_CddTpTransmit is called by an upper layer Cdd to request transmission using a transport protocol.

The IpduM (I-PDU Multiplexer) is placed next to the PDU Router module. PduR_IpduMTransmit requests the transmission for the IpduM. IpduM module is acting as an lower layer module. PduR_IpduMTriggerTransmit is called by the IpduM to request I-PDU contents from upper layer module [6].

## IV. DESIGN OF PDU ROUTER MODULE

Design of PDU Router module consists, of both High Level Design (HLD) and Low Level Design (LLD).HLD specifies, the identification of APIs needed for the implementation of PDUR and the LLD point out the detailed design of APIs.

Figure 5 shows the Low Level Design of PduR_DcmCancelReceive. This function is called by DCM to request a transmission. High level and low level design was done by using Enterprise Architect version 7.5 software.

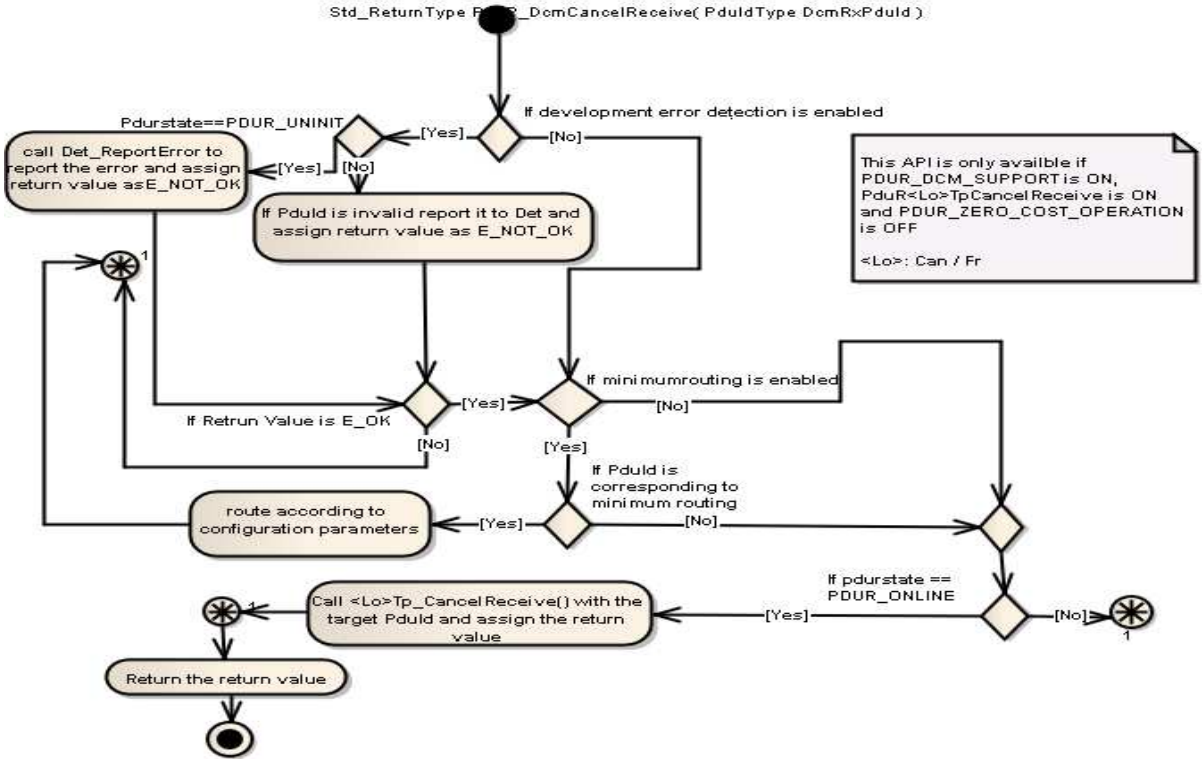


Figure 5. Low Level Design of PduR_DcmCancelReceive

## V. CONFIGURATION

Configuration parameters for the coding were obtained form eZyConfig. This tool was developed by Tata Elxsi Limited. This tool is used for generating the configuration codes for BSW modules. The code file structure of the PDU Router module shall include the following files;PduR_Cfg.c PduR_Lcfg.c and PduR_PBcfg.c.This files represents the pre compile time ,link time and the post build time configuration parameters [7].

## VI. DEVELOPMENT AND TESTING OF PDUR MODULE

The development of PDU Router module was done by implementing all the APIs intended for it. The file structure of the PDU Router module consists of all the header files and the code files. Coding was done in C and by using Microsoft Visual C++ 2008 express edition. It is successfully compiled and builds without error.

Testing of PDU Router module consist of Unit testing and Integration testing. Unit testing refers to the process of testing modules against the detailed design. Integration testing mainly deals with testing for the correctness of the interface. Test Systems 2.3 (TESSY) is used for unit testing of PDU Router. Validation platform used for integration testing is MPC5668G, a 32 bit MCU built in dual core power architecture technology.

## VII. RESULTS

In order to test and verify PDU Router module unit testing, Batch testing, and integration testing were done.

### *A.* Tessy Testing Result

After executing the test in Tessy ,if the expected and the actual values are same the test case will pass by indicating it in green colour.

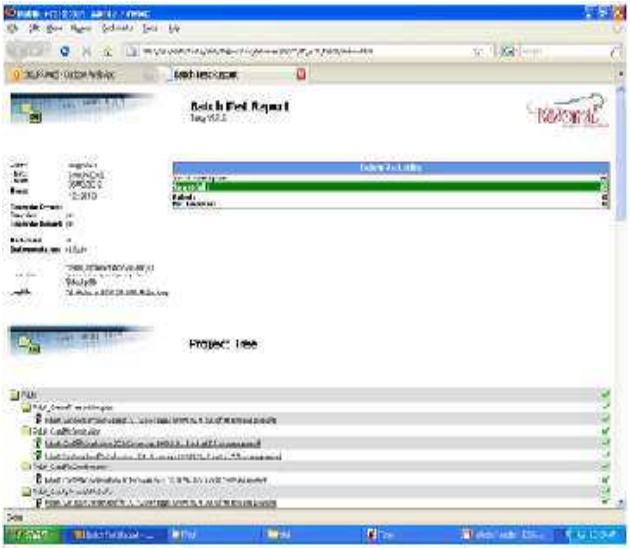

Figure 6. Batch Test Report inTessy

Test report will be generated in XML or HTML format. Figure 6 shows the Batch Test Report in Tessy.

### *A. Integration Testing Result*

The hardware setup consist of Evaluation board (MPC 5668G), CANoe tool, CANCaseXL, CANcardXL, Lauterbach debugger,PC etc. The PC will act as one CAN node and board will act as another CAN node. Only 3 CAN channels of the target board is used for testing. PC is connected to the board through CANCaseXL and CANCardXL. The transmitted and received messages can be viewed in Trace window of CANoe. Four messages are selected for integration testing. They are:

BDY152HA – Message to COM
BDY660HA -Message to COM & Gateway
DNT7E7HA – Message to DCM
DNT7DFHA -Message to DCM & Gateway

Figure 7 shows the layout of BDY152HA. This is a 8 byte message, which is transmitted to COM.

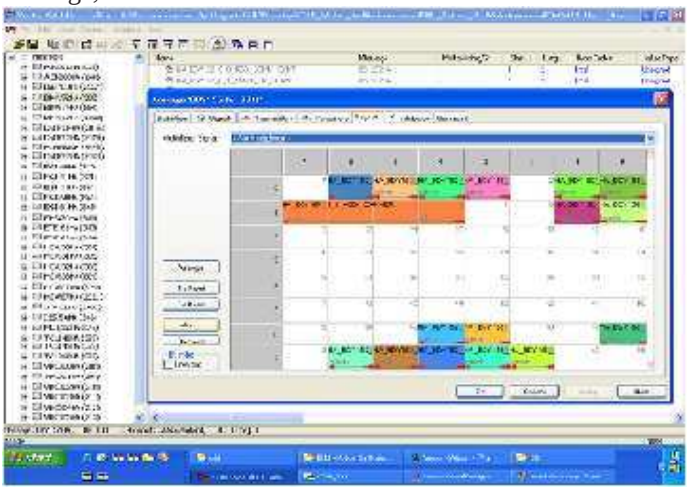

Figure 7. Layout of BDY152HA

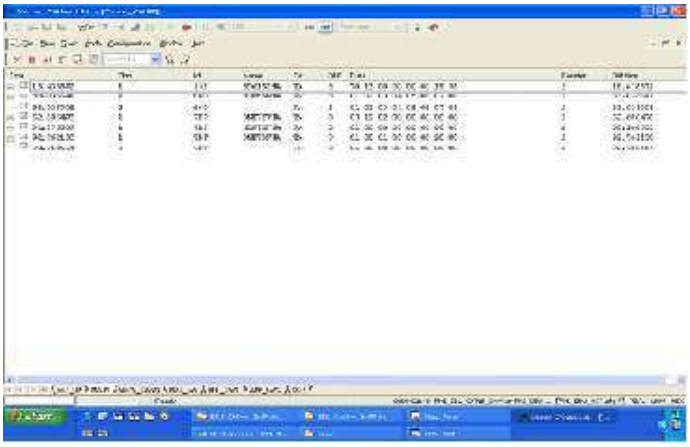

Figure 8. Snap shot of Messages in Trace Window

The transmission of message from PC to board, reception of message from board to PC and gateway of message from one CAN channel to other is successfully tested. The transmitted and received messages can be viewed in Trace window of CANoe. Fig8 shows the snap shot of messages in Trace window.

## VIII. CONCLUSION

This work has resulted in a working AUTOSAR PDU Router module. This was achieved by analyzing the requirements and preparing the high level design. Based on the high level design, low level design was done by drawing flow charts for all APIs. After the design, coding was done by using Visual C++, the correctness of the APIs was tested in unit level and integration level using the tool Tessy. Batch test report is generated using the Tessy software. Integration Testing was done on MPC5668G Evaluation Board. Messages were successfully transmitted and received in the integration testing by using the MPC5668G Evaluation Board and CANoe tool.

Believed that with the constant improvement of AUTOSAR, the PDU Router module as an important part of AUTOSAR, will play an increasingly important role in the development of automotive electronics industry. With the increasing demands for a safe and convenient car, the complexity of automotive ECU is constantly increasing. AUTOSAR enables the development of sophisticated ECU software with high quality and efficiency. In the future, AUTOSAR can be widespread to deal with the currently faced challenges, such as reducing time to market, managing the complexities, improving efficiencies, and so on.

## REFERENCES

[1] Gosda,Johannes.”*AUTOSAR Communication Stack”* .Technical report,Hasso-Plattner Institute fur Softwaresystemtechnik ;pp.13-23;2009

[2] AUTOSAR Partnership,”AUTOSAR_Technical Overview V2.2.2R3.2Rev1”.
[Online].Available.http://www.autosar.org/download/R3.2/AUTOSAR TechnicalOverview.pdf 2011

[3] Warschofsky, Robert,”*AUTOSAR Software Architecture*”,Technical report,Hasso-Plattner Institute fur Softwaresystemtechnik ;2009

[4] Schreiner Dietmar “Component based middleware for AUTOSAR”Master's Thesis,Vienna University of Technology,pp.42-45,2009

[5] Leppla,Gareth”*Mapping Requirements to AUTOSAR software components”*;Master's Thesis;Waterford Institute of Technology Awards Council;pp.51-58;June 2008

[6] AUTOSAR GbR:”Specification of PDU Router”V2.4.0R3.2Rev1
[Online]Available:http://www.autosar.org/download/R3.2/AUTOSAR_ SWS_PDU_Router.pdf 2011

[7] eZyConfig manual,Tata Elxsi Limited,November 2009

[8] Han Wu;”Vehicle Diagnostic with AUTOSAR”Master's Thesis;KTH Computer Science and Communication ;pp.11-14;2008