ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 3, Issue 1, July 2013

TCP/IP Stack Implementation for Communication over IP with AUTOSAR Ethernet Specification

Robert.S, Dr. Jayasudha J.S, Anurag
M.Tech Student, Prof. and HOD, Dept. Of CSE, Competency Manager
Sree Chithra Thirunal, Sree Chithra Thirunal, Tata Elxsi Pvt. Ltd
College of Engg, Trivandrum

Abstract:- ***The new generation vehicle will provide connectivity and telematics services for enabling vehicle communication. The communication over IP in TCP/IP means enables a connection between external node and in-vehicle nodes using IP protocols. In order to bring IP into the vehicle communication Ethernet technology is needed. Ethernet technology will provide high speed data transmission. A hardware independent Ethernet driver is necessary for providing Ethernet services. In vehicle diagnostics, the diagnostic tools and vehicles are separated by an internetwork. The main aim of using IP into the family of automotive communication protocol is that the development of new in-vehicle network has led to the need for communication between external equipment and onboard Electronic Control Units (ECUs) using many data link layer technologies. Diagnostic Over IP (DoIP) is a protocol mainly used for communication between off-board and on-board diagnostic system. This will improve the opportunities of interconnecting in-vehicle networks with internet for many new applications, including online, remote automotive diagnostics. Due to the limited resource availability in embedded devices, lightweight TCP/IP implementation is adopted. AutoSAR (Automotive Open System Architecture) is a standardized architecture introduced by a group of automobile manufacturers, suppliers and tool developers for developing and integrating software modules from different vendors.***

I. INTRODUCTION

New generation vehicles introduce the concept of providing communications services through the use of advanced software's and microcontrollers. Nowadays many ECUs (Electronic Control Units) are inserted in the vehicle for different purposes and they communicate with each other through buses and act like a LAN (Local Area Network). In order to communicate these units with the outside world (internet) an IP protocol is mandatory. Different vendors use different hardware and software components to create an automotive communication network. This has caused difficulties in developing reliable software and integrating the different components. To facilitate software development, a group of automobile manufacturers, suppliers and tool developers created standardized automotive software called AutoSAR (Automotive Open System Architecture) [1]. AutoSAR is the standard software architecture used for providing a basic infrastructure to assist with developing vehicular software, user interfaces and management for all application domains. This includes the standardization of basic system functions, scalability to different vehicles and platform variants, transferability throughout the network, consideration of availability and safety requirements, integration from multiple suppliers, maintainability throughout the entire product life cycle and software upgrades and updates over the vehicles life time. AutoSAR is jointly developed by automotive manufacturers, suppliers and tool developers. This architecture model is supported by an automated methodology to create the software executable for the ECUs, starting from the design model and the properties and physical topology of the hardware. With the development of computer technology, embedded TCP/IP protocol stack becomes an important part of network information intelligence [2]. TCP/IP represents a protocol stack, which is composed of a series of small and special protocol including IP (Internet Protocol), TCP (Transmission Control Protocol), UDP (User Datagram Protocol), ICMP (Internet Control Management Protocol), FTP (File Transfer Protocol), SMTP (Simple Mail Transfer Protocol) and ARP (Address Resolution Protocol). Embedded equipment for upper network protocol stack usually chooses to use the TCP/IP protocol. TCP/IP protocol can support internal network or networks to communicate and exchange of information with the internet and world-wide networking. The characteristics of embedded TCP/IP are portable, configurable, real time, simplicity and flexibility. Due to the limitations of computing resources in ECUs, a lightweight TCP/IP implementation in embedded system is needed [3]-[4]. The link layer of the TCP protocol stack, network interface layer, is referred to the Ethernet physics interface and the Ethernet interface. The link layer is to drive the Ethernet interface circuit, programming network card driver and Ethernet interface according to interface circuit, so as to realize the system connects network.

The development and introduction of new diagnostic concepts and diagnostic solution [5] offer significant potential to automotive OEMs (Original Equipment Manufacturers) and suppliers for realizing efficiency gains and quality improvement. Growing complexity in automotive electronics can only be mastered technically and economically by use of nonproprietary standards such as ODX (Open Data Exchange), close cooperation and powerful tools. . The main aim of using IP into the family of automotive protocol is that the development of new in-vehicle network has led to the need for communication between external test equipment and

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 3, Issue 1, July 2013

onboard ECUs using many data link layer technologies. Until few years ago, the opinion was that Ethernet would never be used for in-vehicle applications, with the exception of diagnostic access [6]. The Ethernet technology presents a new challenges to automotive OEMs, suppliers and development tool producers, because the internet protocol (IP) and Ethernet represents a new network technology for motor vehicles [7]-[8]. In the existing technologies, many protocols are used for communication between different ECUs inside the vehicle. As far as the number of ECUs increased, it demands high data transfer rate. High data transfer rate is achieved by using high bandwidth cable. the paper is organized as follows. Section 2 elaborates the reason for Ethernet in vehicle communication. Section 3 describes general hardware feature of Ethernet communication. Section 4 describes the process of development of Ethernet driver and its corresponding interface in AutoSAR specification. Section 5 describes the result obtained after implementation and concluded in section 6.

II. ETHERNET IN AUTOMOTIVE

Many years ago, there was an opinion that Ethernet would never be used for in-vehicle applications. Camera based driver assistance system was the first applications to utilize Ethernet technology as a system network [9]. Ethernet could contribute homogeneous future communication architecture for high bandwidth applications with cost-effectiveness. There is several communication architectures used in existing vehicles. The common architectures used are LIN (Local Interconnect Network) [10], CAN (Controller Area Network) [11], FlexRay [12], I^2C (Inter Integrated Circuits), MOST (Media Oriented System Transport) [13] etc. The in-vehicle interdomain communication between multiple not directly compatible networking technologies requires the support of complex gateways. For example, if a device connected to CAN has to send a request to configure a parameter to a device connected to Flex Ray, the communication requires a mediating gateway to coordinate the activities of these heterogeneous architectures. A homogeneous networking technology for in-vehicle control units would solve the problem and simplify the complex architecture of the vehicle. New vehicles contain more than 70 ECU's for different applications such as engine control, airbags, antilock breaking, cruise control, electric power steering, mirror adjustment, battery and recharging system for hybrid or electric cars and so on. This means that the traffic requirement is steadily growing and there is a need for high bandwidth. Ethernet technology can provide a homogeneous and high bandwidth for data transmission. Due to the scalable feature, the Ethernet can meet the scalability requirements imposed by the today's automotive systems, where the number of nodes to interconnect steadily increases. The standardization and openness of Ethernet also force the automotive communication to adopt the Ethernet technology in vehicles. The Ethernet offer a support to the TCP/IP stack. This ensures the internet connectivity will be provided in-vehicle, opening the way to enhanced navigation functionalities, remote diagnostics, location based services and smart charging in electric cars [14]-[16]. The table .1 shows the data transfer rate of various communication protocols in vehicle.

Table 1: The data Transfer rate of different communication protocols

Protocol	Data Transfer Rate
I^2C	100Kbps – 3.4Mbps
CAN	5Kbps – 1Mbps
LIN	19.2Kbps
FlexRay	Upto 10Mbps
MOST	25Mbps, 50Mbps, 150Mbps
Ethernet	10Mbps, 100Mbps, 1Gbps

III. HARDWARE COMPONENTS

The figure 1 shows that the MCU (Micro Controller Unit) and Ethernet Controller make any device connected to the internet. Now the designers of embedded system incorporate Ethernet connectivity into their systems. Nowadays large number of high quality chips, supporting Ethernet technology is available in the market. Some of them are ENC28J60, PIC18F97J60, ENC624J600, PIC32MX6XX, MPC5668G/E, and TMS570LS3137 and so on. The stand-alone Ethernet controller contains both Ethernet MAC and Ethernet PHY can communicate with microcontroller unit (MCU) to provide single port network connectivity. TMS570LS3137 [17] is used as the MCU that contains Ethernet MAC and dp83640 as the PHY device for the TCP/IP stack implementation for communication over IP with AUTOSAR Ethernet specification.

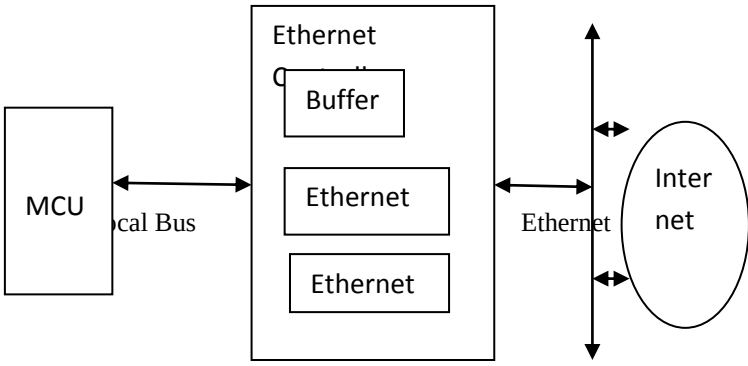

Fig 1: Hardware connection representation

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 3, Issue 1, July 2013

The buffer area is used to store data before transmission and store data after reception. The MDC (Management Data Clock) is used to synchronize the communication between microcontroller and PHY device. An RJ45 connector is used to connect the microcontroller to the internet.

IV. DEVELOPMENT PROCESS

Development of correct requirements at the beginning of a software project is considered an important precondition for successful software development. Rigorous implementation of requirements in embedded software development is especially critical, since requirements affect both software and hardware. The goal is to identify elements for effective requirements implementation in embedded software development. The AUTOSAR software requirement specification for Ethernet Driver, Ethernet Interface, Ethernet Transceiver Driver, Ethernet State Manager, Communication Manger, hardware requirements of the Ethernet controller and the PHY device are captured. The hardware requirements are captured according to the specifications of the microcontroller. The Embedded driver design in AUTOSAR is entirely different from designing Linux device drivers. The driver development is divided into Static part and Dynamic part. The static part remains fixed, whereas the dynamic part can be changed, depending on the application. The static part includes API function calls, constants, global variables that shall not be changed once designed. The dynamic part includes configuration parameters for the required module. There are standard parameters and vendor specific parameters, where the former one is used by AUTOSAR. It is generic and standardized whereas the latter one mentions the specifications given by hardware. In this work, an MCAL (Microcontroller Abstraction Layer) generator tool, which is a PC tool, is used. The MCAL tool used in this work is the eZyConfig. It consists of two parts. One is Driver configuration part and other is Code generation part. The input to Driver configuration part is the definition file. The output is the Description file. This description file is one of the inputs to the dynamic code generation part. The second input is the Template file. The final output is the configuration files such a **.c** and **.h** files. The figure 2 shows the different steps for generating dynamic part of the driver.

a) ***Definition file***

The definition file contains the definition for each of the configuration parameters of the ECU. The definition file is unique and contains all the definitions of all the modules in the AUTOSAR. It is required to segregate the parameters for our module, here Ethernet and configure only those parameters required for our module. The definition file is generally an **.arxml** file (autosar xml file). It is an xml file as per AUTOSAR standards. It contains definitions for module containers and parameters. It gives standardized definition format per module with type, allowed range such as minimum and maximum values, multiplicity, default values etc. In the standard **.arxml** file user can add additional parameters depends on the microcontroller specification. Depends upon the microcontroller we can add controller specific parameters.

b) ***Description File***

The ECU Configuration Description file is also an **.arxml** file obtained as output from the ECU Spectrum tool. It consists of precise description about number of instances, parameter value definitions etc. This file is used for generating **.c** and **.h** configuration files.

c) ***Template file***

Template file is used to extract the information and logic from the description file for generating configuration files. This file gives a detailed description of module description in a structure format. It can be written using any scripting language.

d) ***Configuration file generation***

The definition file is given as input to the ECU Spectrum tool, which generates another file named Description file according to our requirement. ECU Spectrum is a user friendly configuration editor GUI based tool to configure AUTOSAR BSW & RTE Modules. Module Generator is used to generate configuration files (.c, .h) from the ECU Configuration Description file using template file. In our project the tool used was the eZyConfig.

A. Source code Preparation

For the development of static part of the driver, the configuration files such as **.c** and **.h** developed as part of dynamic code generation are used. The configuration file contains dynamic variable declaration, its initial values etc. These values are used by the static part for the initialization of hardware. The values in the configuration file may change depending on the application, but the static part remains unchanged. The static part implemented is depends on the specification mentioned in concerned AUTOSAR modules. The static part contains API function calls, both global and local. Local API calls are used internal to module.

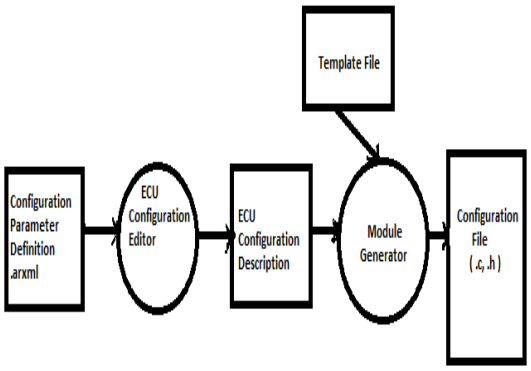

Fig 2: The Different Steps for Generating the Dynamic Configuration of the AUTOSAR Driver

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 3, Issue 1, July 2013

V. RESULTS

The embedded drivers and its corresponding Interface are developed using AUTOSAR specification, then integrated with existing lightweight embedded TCP/IP stack. An application program for transmitting web data is developed and run above the TCP/IP stack. The different modules are build and debug using Code Composer Studio. After successful building and debugging the program is flashed into the microcontroller. Then the microcontroller act as a standalone device and the web application running inside the microcontroller is accessed through the browser running in any node system connected to the corresponding internet through the Ethernet port of microcontroller. Thus communication over IP is provided.

VI. CONCLUSION

The communication over IP communication over IP using existing TCP/IP embedded stack is achieved with AUTOSAR Ethernet specification. In order to provide the communication over IP a data link layer service is mandatory. The Ethernet interface and Ethernet driver comes under the data link layer services. Instead of using TCP structure, a socket adapter run over the Transport layer may be used to transmit complex data. This may reduce the overhead associated with TCP transmission. Ethernet usage in car is expected to spread in several domains such as diagnostics, multimedia and infotainment, Advanced Driver Assistance System (ADAS), charging of electric cars etc. A shift from proprietary solutions to novel network architecture, based on an established standard technology, would allow for faster design and analysis of the network transmission schedule, better quality and performance assessments. The adoption of a common communication network architecture (example: Ethernet) would simplify the task of network suppliers allowing for component reusability across different car manufacturers shortening the time to market of their products.

VII. ACKNOWLEDGEMENT

This work has been supported by TataElxsi Pvt Ltd, Trivndrum. The author is grateful to Tata Elxsi Pvt Ltd, Trivadrum and the management of Tata Elxsi for their support and interesting discussions.

REFERENCES

[1] AUTOSAR [have page on the internet], Munich: AUTOSAR; (2003-2011), Available from http://www.autosar.org.

[2] YanHongwei, Pan Hongxia, " The design and Implementation of Network Data link Layer based on Embedded TCP/IP protocol stack", in Proceedings of International Conference on Networking and Information Technology, 2010.

[3] Ri-Kun Liao, Yue-Feng Ji, Hui Li, "Optimized Design and Implementation of TCP/IP Software Architecture Based on Embedded System", in Proceedings of the Fifth International Conference on Machine Learning and Cybernetics, Dalian, 13-16 August 2006, pp. 590-594.

[4] Yun-Chen Li, Mei-Ling Chiang, "LyraNET: a zero-copy TCP/IP protocol stack for embedded operating systems", in Proceedings of the 11th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications, 17-19 Aug. 2005, pp. 123-128.

[5] ISO/CD 13400, "Road vehicles — Diagnostic communication between test equipment and vehicles over Internet Protocol (DoIP)," 2009.

[6] Mathias Johanson, Pål Dahle and Andreas Söderberg," Remote Vehicle Diagnostics over the Internet using the DoIP Protocol", Proceedings of the Sixth International Conference on Systems and Networks Communications, 2011.

[7] Nöbauer, J., Continental AG: Migration from MOST and FlexRay Based Networks to Ethernet by Maintaining QoS. 1st Ethernet & IP @ Automotive Technology Day, BMW, Munich, 2011.

[8] Bogenberger, R., BMW AG: IP & Ethernet as potential mainstream automotive technologies. Product Day Hanser Automotive. Fellbach, 2011.

[9] Hans-Werner Schaal, Vector, "IP and Ethernet in Motor Vehicles" technical article, April 2012.

[10] LIN Consortium, Lin Specification Package Revision 2.0, Motorola GmbH, Schatzbogen 7, 81829 Munchen, Germany, Sep. 2003.

[11] CAN Specification Version 2.0, Robert Bosch GmbH, Postfach 50, 70049, Stuttgart, Germany, Sept. 1991.

[12] FlexRay Consortium, FlexRay Communications System Protocol Specification Version 2.1 Revision A, Altran GmbH and Co. KG, Schillerstr. 20, 60313 Frankfurt am Main, Germany, Dec 2005.

[13] MOST, Media Oriented System Transport, Multimedia and Control Networking Technology, Rev 2.4, OASIS Sylicon systems 2005.

[14] Bruckmeier, R.2010. Ethernet for Automotive applications. Free scale Technology Forum (Orlando, FL, US, June 23, 2010).

[15] Hammerschmidt, C.2011. Ethernet to gain ground in automotive applications. Bosch predicts. EETimes Europe. (4 FEB 2011).

[16] Lim, H.T., Volker L, Herrscher.D, 2011. Challenges in a future IP/Ethernet based in-car network for real-time applications. Proceedings of the 48th ACM/EDAC/IEEE Design Automation Conference, 2011.

[17] TMS570LS12x/11x 16/32-Bit RISC Flash. Microcontroller, www.ti.com/general/docs/lit/getliterature.tsp?literatureNumber=spnu515...