TECHNISCHE UNIVERSITÄT
CHEMNITZ

Realization of communication with SOME/IP stack over Ethernet

Research Internship

Submitted in Fulfilment of the
Requirements for the Academic Degree
M.Sc.

Dept. of Computer Science
Chair of Computer Engineering

Submitted by: Sreedhar Hegde
Student ID: 671453
Submitted On: 12.04.2022

Supervising tutor: Prof. Dr. W. Hardt

Company Supervising tutor: Mr. Rick Podszuweit

Abstract

Contents

List of Figures

List of Tables

List of Abbreviations

ADAS Advanced Driver Assistance Systems

ARP Address Resolution Protocol

ARXML AUTOSAR XML

API Application Program Interface

AUTOSAR Automotive Open System Architecture

AVB Audio Video Bridging

BSW Basic Software

CAN Controller Area Network

CRC Cyclic Redundancy Check

DoIP Diognastic over IP

ECU Electronic Control Unit

Eth Ethernet

EthIf Ethernet Interface

EthTrcv Ethernet Transceiver

IEEE Institute of Electrical and Electronics Engineers

IP Internet Protocol

ISO International Organization for Standardization

IVN In-vehicle Networking

MAC Media Access Control

MCAL Microcontroller Abstraction Layer

MDIO Management Data Input/Output

MII Media Independent Interface

MOST Media Oriented Systems Transport

OABR Open Alliance BroadR-Reach

Open Alliance SIG OPEN (One-Pair Ethernet) Alliance Special Interest Group

OEM Original Equipment Manufacturer

OS Operating Systems

OSI Open System Interconnection

PDU Protocol Data Unit

PHY Physical Layer

POSIX Portable Operating System Interface

RTE Runtime Environment

SDU Service Data Unit

SoAd Socket Adaptor

SOME/IP Scalable service-Oriented MiddlewarE over IP

SWC Software Component

TCP Transmission Control Protocol

UDP User Datagram Protocol

V2X Vehicle-to-Anything communication

1 Introduction

Communication is essential in modern vehicles to establish a link between the ECUs in the network. In addition, as the number of ECUs and high-performance controllers grows, so does the need for more bandwidth than traditional in-vehicle networks such as CAN, Flexray, and MOST can provide. With the introduction of Ethernet into the automotive domain, bandwidths of up to 1 Gb/s can now be achieved within the vehicle network. The use of Ethernet benefits systems such as ADAS and infotainment significantly. However, in order to transmit and receive data at a significantly high data rate, a robust communication control mechanism is required. The use of Ethernet benefits systems such as ADAS and infotainment substantially. However, in order to transmit and receive data at a remarkably high data rate, a robust communication control mechanism is required. With the growing interest in POSIX-based systems in the automotive domain, service oriented architecture (SOA) plays an important role in meeting the needs of technology-driven applications. The core of SOA is remote procedure calling (RPC) and the Client-Server mechanism. To realize these concepts, there is a need for a middleware that is specifically designed to run automotive applications smoothly. To accomplish this, SOME/IP middleware was introduced in the automotive context. As more applications migrate to Adaptive AUTOSAR, SOME/IP is well suited to serve as a communication control protocol alongside existing communication technologies.

In this report, a detailed study of the SOME/IP technology is conducted. In order to understand the working of SOME/IP technology, the open source library vsomeip offered by GENIVI is used. A demonstrator consisting of target hardware running with different underlying architectures such as x64, armv7 and armv8 is setup. The devices are connected with each other on a network using Ethernet. The working is realized by running applications based on vsomeip stack on theses hardware devices. Also, a troubleshooting guide consisting of the commonly faced issues and faults while using the SOME/IP technology have been documented as a reference document.

2 Literature Survey

2.1 Communication technologies in Automotive Domain

It is important to note that the incorporation of Ethernet as an in-vehicle networking system does not imply that traditional communication networks such as CAN, LIN, and MOST are rendered obsolete. Because these networks are robust, inexpensive, time-tested, and provide necessary performance for many applications, Automotive Ethernet will not completely replace them, but will supplement them to provide even more cost, performance, and feature benefits. Table 2.1 shows the important characteristics of automotive networks in comparison with the Ethernet.

Property	**Ethernet**	**CAN**	**FlexRay**	**MOST**	**LIN**
Bandwidth(Mb/s)	>100	1	20	150	0.02
Nodes	Scalable	30	22	64	16
Network Length	15m per link	40m	24m	1280m	40m
Topologies	Star, Tree	Bus	Bus,start	Ring,Star	Bus
Cost	High	Low	Low	High	Very low
Cabling	UTP	UTP	UTP	Optical,UTP	1-wire

Table 2.1: Comparison of important characteristics of automotive networks [1, p.202]

2.2 Role of Ethernet in the Automotive Domain

Insert text here.

2.3 Ethernet as backbone in vehicles

Insert text here.

2.4 Service Oriented Architecture

Insert text here.

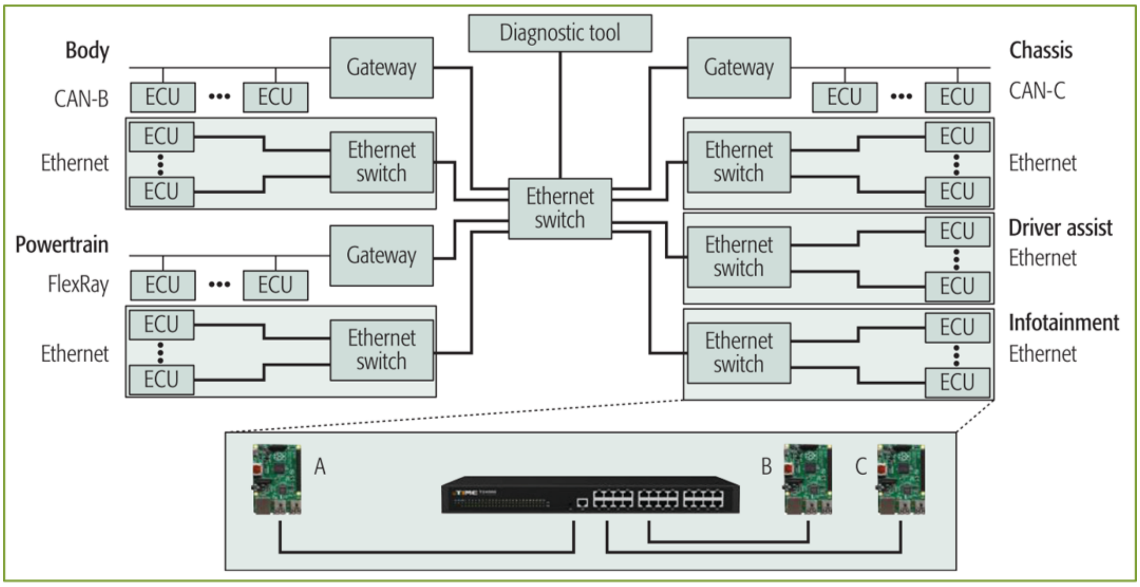

Figure 2.1: Ethernet as a backbone [12]

2.5 Middleware technologies in automotive domain

Briefly describe about the need and existing technologies. (SOME/IP, DDS)

2.5.1 SOME/IP

"'Scalable service Oriented MiddlewarE over IP"' abbreviated SOME/IP represents a middleware that was created for automotive use cases [2]. The compatibility with AUTOSAR was a necessity regarding SOME/IP at least on wire-format level [2]. SOME/IP communication is an exchange of messages between different devices like ECUs over IP [2]. There are more patterns available for basic SOME/IP communication [2].

Communication methods

Brief description of the communication methods. Explain in brief about each of the types.

- Request-Response

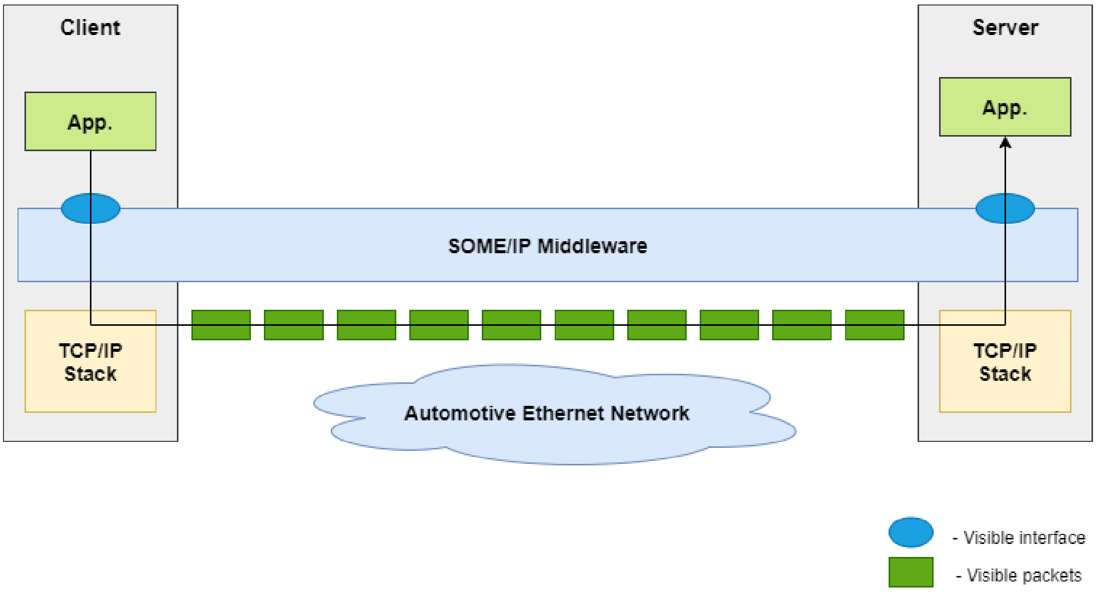

Figure 2.2: SOA representation with SOME/IP middleware

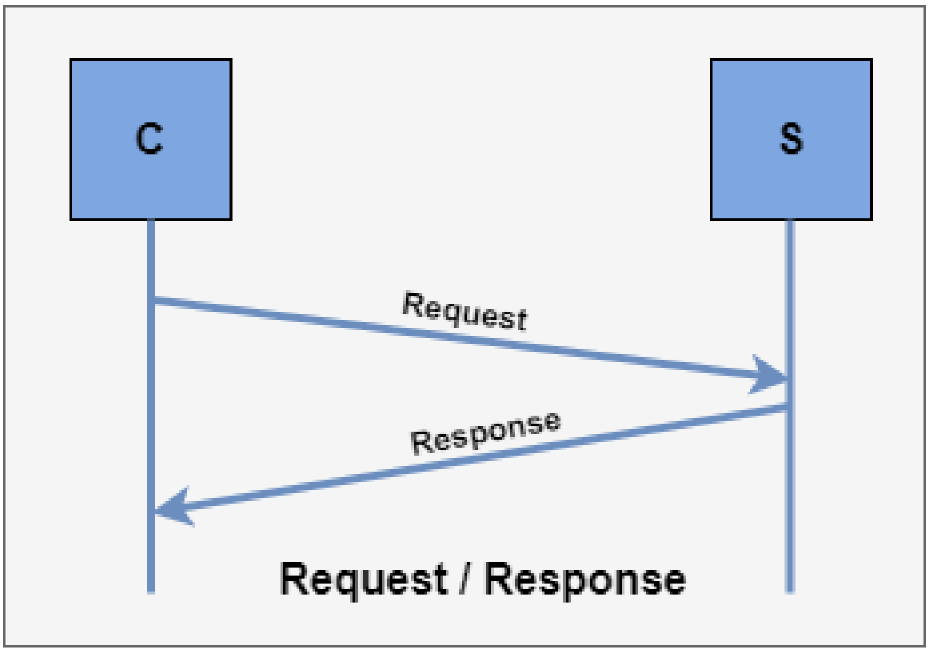

Figure 2.3: Request-Response communication type

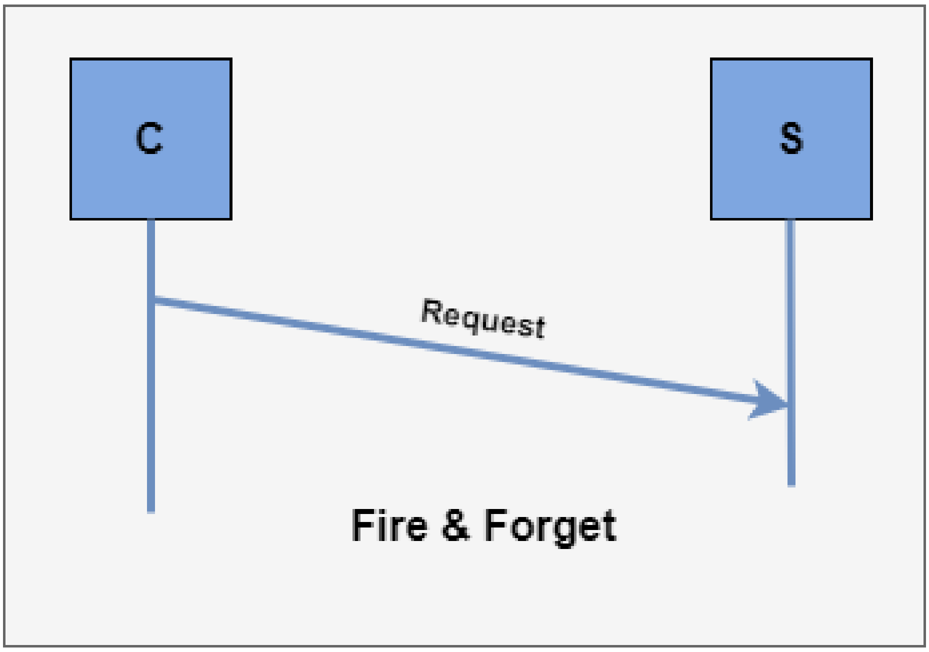

Figure 2.4: Fire & Forget communication type

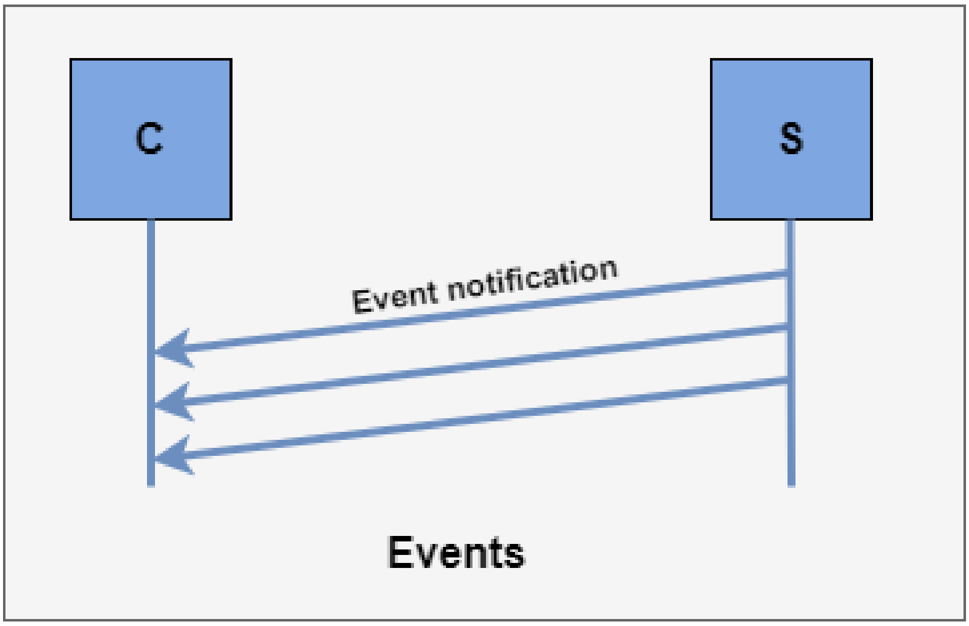

Figure 2.5: Events communication type

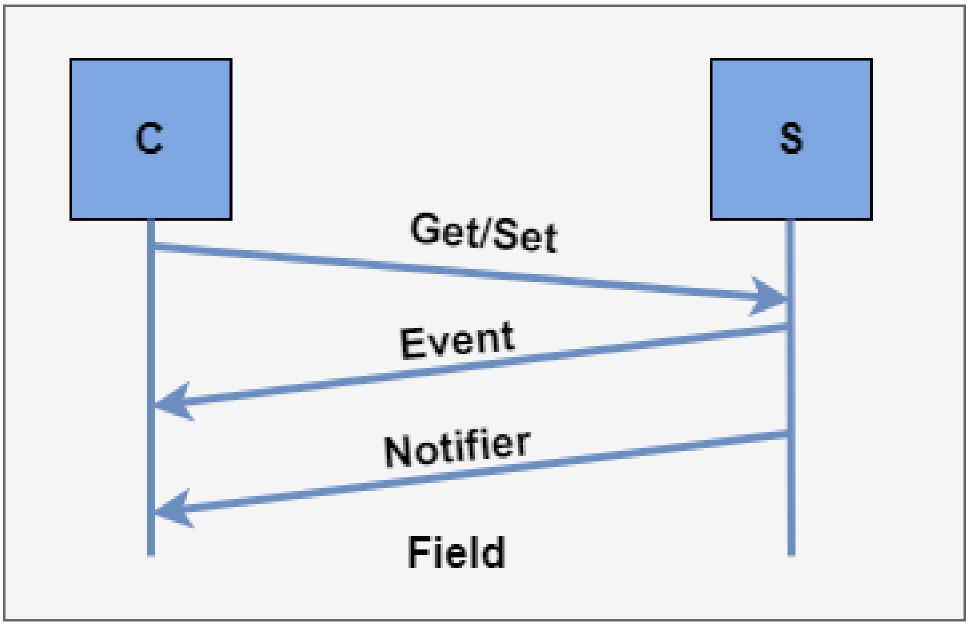

Figure 2.6: Fields communication type

3 Implementation

To demonstrate the use of the SOME/IP technology, several devices (target controllers) are connected within a network and communication is established between them using Ethernet. In this chapter, the requirements to visualize the technology are explained and also the procedure to setup the demonstrator is discussed in detail.

3.1 Concept

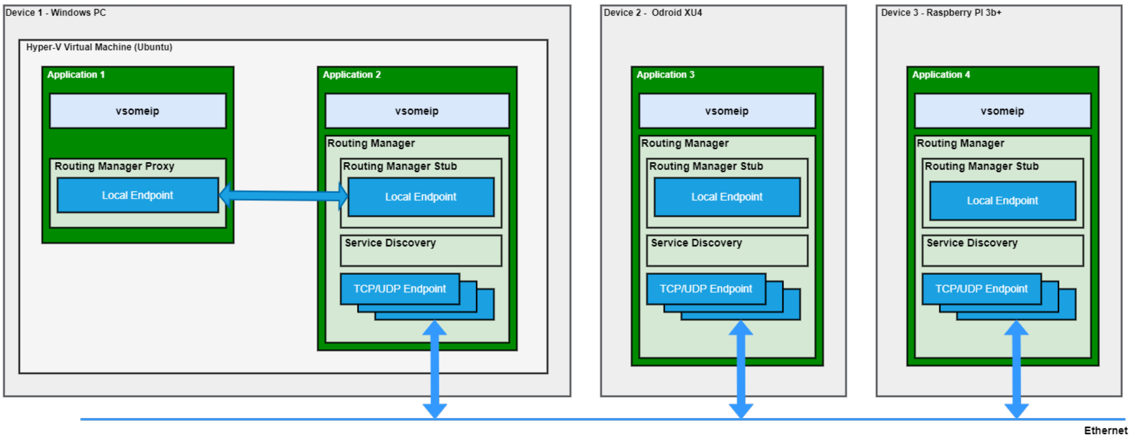

Figure 3.1: SOME/IP concept

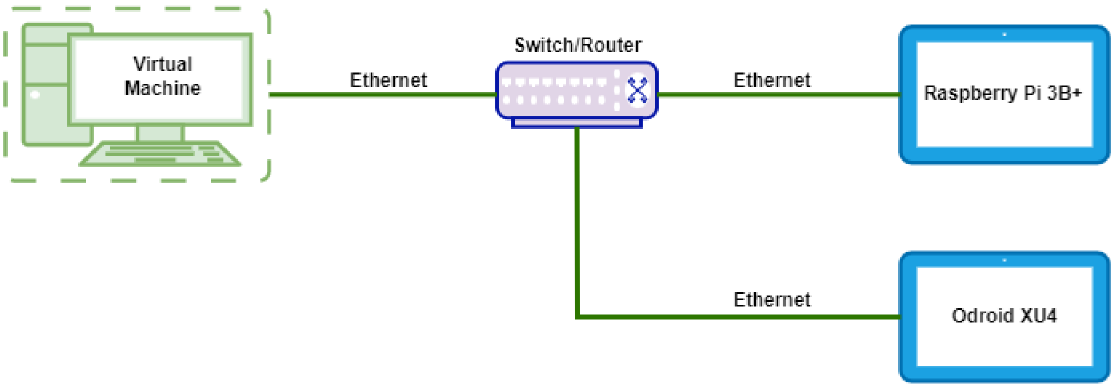

Figure 3.2: Visual representation of hardware setup

3.2 Environment

3.2.1 Operating system

Linux OS

3.3 Target Hardware

3.3.1 Virtual Machine

3.3.2 Raspiberry Pi 3b+

3.3.3 Odroid XU4

3.4 Software

3.4.1 Target Libraries

GENIVI vsomeip stack

Other dependencies

Boost

3.4.2 Tools

Qt Creator

Describe in brief about Qt Creator

CMake

Describe in brief about CMake

PuTTY

PuTTY is an SSH and telnet client, developed originally by Simon Tatham for the Windows platform[6]. PuTTY is open source software that is available with source code and is developed and supported by a group of volunteers[6]. This tool is required to run the terminals for Raspberry Pi3b+ and Odroid XU4 remotely on the host machine. This enables to smoothly switch between the target devices when running the applications on the target hardware respectively.

3.5 Cross-compilation

Every development board is embedded with a specific amount of RAM, storage capacity, input and output peripherals and other hardware components. Hosting the target environment on multiple boards can be complicated and time consuming. Furthermore, building target libraries on these boards can take a long time and may fail in some cases. To address these issues, it is worthwhile to setup a generic build environment on a single platform and build the projects for different targets accordingly. This process is called as cross-compilation. In this section, the process to setup a cross-compilation environment on the Linux platform is illustrated and along with it, the procedure to cross-compile boost libraries for Raspberry Pi and Odroid XU4 target platforms is demonstrated respectively.

3.5.1 Installing cross compilers on the host machine

In this section, the basic requirements to setup a cross-compilation tool-chain is described. Also, based on the requirements for the demonstration of the SOME/IP technology, build process for libraries such as Boost, CommonAPI, vsomeip and other relevant libraries are described in detail.

In order to cross-compile, appropriate tool-chain packages has to be first setup in the host environment. The commands from the following listings are required to be run in a terminal window in the Linux machine. Please note that an active internet connection is required to download the packages from the server.

Listing 3.1: Command to install packages for ARM 32-bit (armv7) tool-chain

```
user@machine:~$ sudo apt-get install gcc-arm-linux-gnueabihf
      g++-arm-linux-gnueabihf
```

Listing 3.2: Command to install packages for ARM 64-bit(armv8) tool-chain

```
user@machine:~$ sudo apt-get install gcc-aarch64-linux-gnu
      g++-aarch64-linux-gnu
```

Listing 3.3: Command to install other required packages

```
user@machine:~$ sudo apt-get install build-essential
      manpages-dev openjdk-8-jdk libssl-dev wireshark
      g++-aarch64-linux-gnu
```

Installing Boost libraries

3.6 Demo Application

3.6.1 Server Application

3.6.2 Client Application

3.6.3 Comunication establishment between devices

Configuration

4 Results

Enter text here

4.0.1 Server Application Output

4.0.2 Client Application Output

4.0.3 Troubleshooting guide

5 Conclusion

Conclusion

Bibliography

[1] Kozierok, C. M., Correa, C., Boatright, R. B., & Quesnelle, J. (2014). Automotive Ethernet: The Definitive Guide. Intrepid Control Systems.

[2] A. Ioana and A. Korodi, "VSOMEIP - OPC UA Gateway Solution for the Automotive Industry," 2019 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC), 2019, pp. 1-6, doi: 10.1109/ICE.2019.8792619.

[3] G. L. Gopu, K. V. Kavitha and J. Joy, "Service Oriented Architecture based connectivity of automotive ECUs," 2016 International Conference on Circuit, Power and Computing Technologies (ICCPCT), 2016, pp. 1-4, doi: 10.1109/ICCPCT.2016.7530358.

[4] "Example for a Serialization Protocol (SOME/IP)," AUTOSAR Release 4.3.1, 31-Mar-2014. [Online]. Available: https://www.autosar.org/fileadmin/user_upload/standards/classic/4-1/AUTOSAR_TR_SomeIpExample.pdf. [Accessed: 06-Nov-2022].

[5] Rumez, M., Grimm, D., Kriesten, R., & Sax, E. (2020). An overview of automotive service-oriented architectures and implications for security countermeasures. IEEE access, 8, 221852-221870.

[6] "Download Putty - a free SSH and telnet client for windows," Download PuTTY - a free SSH and telnet client for Windows. [Online]. Available: https://www.putty.org/. [Accessed: 17-Mar-2022].

[7] Steinbach, T., Korf, F., & Schmidt, T. C. (2011, September). Real-time Ethernet for automotive applications: A solution for future in-car networks. In 2011 IEEE International Conference on Consumer Electronics-Berlin (ICCE-Berlin) (pp. 216-220). IEEE.

[8] Han, S., & Kim, H. (2016). On AUTOSAR TCP/IP performance in in-vehicle network environments. IEEE Communications Magazine, 54(12), 168-173.

[9] Matheus, K., & Königseder, T. (2017). A Brief History of "Ethernet" (from a Car Manufacturer's Perspective). In Automotive Ethernet(pp. 1-29). Cambridge: Cambridge University Press. doi:10.1017/9781316869543.003

[10] Matheus, K., & Königseder, T. (2017). A Brief History of In-Vehicle Networking. In Automotive Ethernet(pp. 30-68). Cambridge: Cambridge University Press. doi:10.1017/9781316869543.004

[11] Matheus, K., & Königseder, T. (2017). The Physical Transmission of Automotive Ethernet. In Automotive Ethernet (pp. 102-188). Cambridge: Cambridge University Press. doi:10.1017/9781316869543.006

[12] Hank, P., Müller, S., Vermesan, O., & Van Den Keybus, J. (2013, March). Automotive ethernet: in-vehicle networking and smart mobility. In 2013 Design, Automation & Test in Europe Conference & Exhibition (DATE) (pp. 1735-1739). IEEE.

[13] Krishnadas, S. (2016). Concept and Implementation of AUTOSAR compliant Automotive Ethernet stack on Infineon Aurix Tricore board.

[14] Kim, Y. (2011). Very low latency packet delivery requirements and problem statements. In IEEE 802.1 AVB Task Group Interim Meeting.

[15] Bo, H., Hui, D., Dafang, W., & Guifan, Z. (2010, May). Basic concepts on AUTOSAR development. In 2010 International Conference on Intelligent Computation Technology and Automation (Vol. 1, pp. 871-873). IEEE.

[16] Wagner, M., Zobel, D., & Meroth, A. (2014, September). Towards runtime adaptation in AUTOSAR: Adding Service-orientation to automotive software architecture. In Proceedings of the 2014 IEEE Emerging Technology and Factory Automation (ETFA) (pp. 1-7). IEEE.

[17] AUTOSAR Release 4.3.1, "Layered Software Architecture", 08-Dec-2017. [Online]. Available: `https://www.autosar.org/fileadmin/user_upload/standards/classic/4-3/AUTOSAR_EXP_LayeredSoftwareArchitecture.pdf`, [Accessed: 05-May-2021].

[18] AUTOSAR Release 4.3.1, "Specification of Ethernet Switch Driver," 08-Dec-2017. [Online]. Available: `https://www.autosar.org/fileadmin/user_upload/standards/classic/4-3/AUTOSAR_SWS_EthernetSwitchDriver.pdf`. [Accessed: 11-May-2021].

[19] AUTOSAR Release 4.3.1, "Specification of TCP/IP Stack", 08-Dec-2017. [Online]. Available: `https://www.autosar.org/fileadmin/user_upload/standards/classic/4-3/AUTOSAR_SWS_TcpIp.pdf`. [Accessed: 13-May-2021].

[20] AUTOSAR Release 4.3.1. "Specification of Socket Adaptor", 08-Dec-2017. [Online]. Available: `https://www.autosar.org/fileadmin/user_upload/standards/classic/4-3/AUTOSAR_SWS_SocketAdaptor.pdf`. [Accessed: 14-May-2021].

[21] Abdelhamid, A. Y. A. (2018). Evaluation of a Control Protocol for Testing an Automotive Ethernet TCP/IP Stack.

[22] Alliance, O. (2017). OPEN Alliance Automotive Ethernet ECU Test Specification. TC8 ECU, August, 23.