DEGREE PROJECT IN ELECTRICAL ENGINEERING, SECOND CYCLE
STOCKHOLM, SWEDEN 2017

# Modern Electrical/Electronic Infrastructure for Commercial Trucks

## *Generic Input/Output nodes for sensors and actuators in Commercial Trucks*

ABHINEET SINGH TOMAR

**KTH ROYAL INSTITUTE OF TECHNOLOGY**
***INFORMATION AND COMMUNICATION TECHNOLOGY***

# Modern Electrical/Electronic Infrastructure for Commercial Trucks

## *Generic Input/Output nodes for sensors and actuators in Commercial Trucks*

Abhineet Singh Tomar

2017-12-17

Master's Thesis

Examiner
Gerald Q. Maguire Jr.

Academic adviser
Anders Västberg

Industrial Supervisor
Roy Johansson

KTH Royal Institute of Technology
School of Information and Communication Technology (ICT)
Department of Communication Systems
SE-100 44 Stockholm, Sweden

# Abstract

The presence of electrical and electronic circuits in commercial trucks has increased at a very fast rate during recent decades. With advancements in embedded systems and the introduction of electric controls in the automotive industry, the design of complex electric systems for the vehicles has become one of the major design challenges. In the commercial truck industry, the development cycles are almost a decade long. Therefore, it is a big challenge to introduce a new architecture to accommodate the modern automotive technologies in the upcoming generation of trucks.

Currently, the commercial truck industry relies highly on a federated electrical/electronic (E/E) architecture. In this architecture, Electronic Control Units (ECU) are responsible for computation and Input/Output operations. These ECUs are clustered into different domains based on their respective functions. However, these domains are not isolated from each other. These modules communicate with each other using a vehicular network, which is typically a controller area network in the current trucks.

In the automotive industry, automation is increasing at a fast pace. As the level of automation increases, the need for high computation also increases, which increases the overall costs. This study aims to address this problem by introducing an integrated E/E architecture where all the computational power is concentrated at one place (or perhaps two or three places to allow for redundancy). This study proposes to introduce a low-cost replacement for the current ECUs with more limited computational power but with generic input/output interfaces.

This thesis provides the reader with some background of the current E/E architecture of commercial trucks and introduces the reader to ECUs. Additionally, the relevant network architectures and protocols are explained. A potential solution, based upon the centralized computation based E/E architecture and its implementation are discussed followed by a detailed analysis of the replacements for ECUs. The result of this analysis, if adopted, should result in a reduction of manufacturing and design costs, as well as make the production and maintenance process easier. Moreover, this should also have environmental benefits by reducing fuel consumption.

**Keywords:** ECU, E/E Architecture, CAN, LIN, Automotive Ethernet, Integrated Architecture, Commercial Trucks, Autonomous Driving

# Sammanfattning

Förekomsten av elektronik och elektriska kretsar I kommersiella lastbilar has ökat i en väldigt snabb takt under de senaste decennierna. Med framsteg inom inbyggda system och introduktionen av elektroniska styrsystem i fordonsindustrin så har komplexa elektroniska system blivit en av de största designutmaningarna. I den kommersiella lastbilsindustrin där utvecklingscyklerna är nästan ett decennium, är det en stor utmaning att introducera ny arkitektur som tillgodoser all den nya teknologin som införlivas i fordonet.

För närvarande så förlitar sig den kommersiella lastbilsindustrin mycket på en federated elektrisk/elektronisk (E/E) arkitektur. I denna arkitektur är elektroniska styrenheter (ECU) ansvariga för beräkningar och I/O (Input/Output) operationer. Dessa ECU:er är samlade i olika domäner baserade på dess funktioner. Domänerna är dock inte isolerade från varandra. De här modulerna kommunicerar därför med varandra med hjälp av ett fordonsnätverk, typiskt en CAN (Controller Area Network) i nuvarande lastbilar. I fordonsindustrin ökar automatiseringen i en snabb fart. I takt med att automatiseringen ökar så ökar även behovet av snabba och energiintensiva beräkningar, vilket i sin tur ökar den totala kostnaden. Denna studie har som mål att adressera det här problemet genom att introducera en integrated E/E arkitektur där all beräkningskraft är koncentrerad till en plats (eller två eller tre platser för att tillåta överskott). Den här studien föreslår att introducera en ersättning av nuvarande ECU:er till en låg kostnad, med lägre beräkningskraft och generiska I/O gränssnitt. Studien föreslår också ersättningar av nuvarande fordonsnätverk.

Den här uppsatsen förser läsaren med viss bakgrund till den nuvarande E/E arkitekturen för kommersiella lastbilar och introducerar läsaren till ECU:er. Dessutom förklaras de relevanta nätverksarkitekturerna och protokollen. En potentiell lösning som baseras på den integrated E/E arkitekturen och dess implementering diskuteras med fokus på en detaljerad analys av ersättningarna till ECU:er. Resultatet av den här analysen skulle, om den adopteras, medföra minskning av tillverknings- och designkostnader samt leda till en förenkling av produktion och underhåll. Utöver det så bör det även ha miljöfördelar genom minskad bränsleförbrukning.

**Nyckelord:** ECU, E/E Arkitektur, CAN, LIN, Automotive Ethernet, Integrerad arkitektur, Kommersiella lastbilar, Autonom körningxt

# Acknowledgements

After working in the hardware aspect of almost all the projects during my masters, I eventually found a rare thesis project which gave me a chance to work in the same field. I would begin by thanking Roy Johansson, my industrial supervisor and mentor, who gave me this opportunity to work. It was great to be involved in all the discussions and decisions related to specifications and scope of the project. Mikaela Öhman was always there with her expertise despite her busy schedule. Discussions with Dhas and Girish gave me constant insights about where my work fits in the bigger picture. My thesis partner, Johan was always there for a company during lunch and all the discussions, he also helped in translating the abstract in Swedish. Special thanks to my manager, Alejandro Cortes, who was appreciative of all the work.

I would like to thank my examiner Prof. Gerald Q. Maguire Jr. for the constant feedback and patience; his guidance right from the time of Research Methodology course helped in improving the way I approach problems. Prof. Mark Smith was always there as a mentor during my time at KTH, giving opportunity and space to work on numerous projects, which helped me in developing skills needed for this thesis work.

Among my friends, special thanks to Abu Darda, who always ended up suggesting me the right things at right time to work with, this thesis project included. Hassan Mahmood, who spared time to review almost everything I wrote. Aditya Gahlaut, who has been helpful throughout my Masters. Prashant Sharma was the person who did the most difficult task of finding a place for me to live in a city like Gothenburg, otherwise, I would have been a homeless thesis worker. Akshay and Krishna helped me in settling in the new city and finding things to do in leisure time.

Big thanks to my family and a couple of my close friends for their unconditional support and motivation. Special thanks to my parents who directly or indirectly taught me to be optimistic and stay focused on right things.

Gothenburg, December 2017
Abhineet Singh Tomar

# Contents

# List of Figures

# List of Acronyms and Abbreviations

This document requires readers to be familiar with terms and concepts related to Electrical/Electronic Architecture of commercial vehicles. For clarity they are summarized with a short description of them before presenting them in next sections.

| | |
|---|---|
| **A** | Ampere |
| **ACK** | Acknowledge |
| **ADAS** | Advanced Driver Assistance System |
| **Amp** | Ampere |
| **ARM** | Advanced RISC Machine |
| **ASCII** | American Standard Code for Information Interchange |
| **B8ZS** | Bipolar With Eight-zero Substitution |
| **CAN** | Controller Area Network |
| **CAN-FD** | Controller Area Network Flexible Data-rate |
| **CRC** | Cyclic Redundancy Check |
| **CSMA/CD** | Carrier Sense Multiple Access/Collision Detection |
| **CSMA/CR** | Carrier Sense Multiple Access/Collision Resolution |
| **DDP** | Distributed Data Protocol |
| **DLC** | Data Length Code |
| **EBCDIC** | Extended Binary Coded Decimal Interchange Code |
| **ECM** | Electronic Control Modules |
| **ECU** | Electronic Control Unit |

| | |
|---|---|
| **EMI** | Electromagnetic Interference |
| **FTP** | File Transfer Protocol |
| **GIO** | Generic Input/Output |
| **GTM** | Generic Timer Module |
| **GPS** | Global Positioning System |
| **HDLC** | High-level Data Link Control |
| **HTTP** | Hyper-text Transfer Protocol |
| **I2C** | Inter-Integrated Circuit |
| **I/O** | Input/Output |
| **IBS** | Inter Byte Space |
| **IFS** | Inter Frame Space |
| **IP** | Internet Protocol |
| **IPX** | Internetwork Packet Exchange |
| **ISO** | International Standards Organization |
| **LIN** | Local Interconnect Network |
| **LLC** | Logical Link Control |
| **MAC** | Media Access Control |
| **MIDI** | Musical Instrument Digital Interface |
| **MOSFET** | Metal–Oxide–Semiconductor Field-Effect Transistor |
| **MPEG** | Moving Picture Experts Group |
| **NetBIOS** | Network Basic Output System |
| **NRZ** | Non Return to Zero |
| **OEM** | Original Equipment Manufacturers |
| **OPEN** | One-Pair Ethernet |

| | |
|---|---|
| **OSI** | Open Systems Interconnect |
| **PAM** | Pulse Amplitude Modulation |
| **PHY** | Physical Layer |
| **PPP** | Point-to-point Protocol |
| **RF** | Radio Frequency |
| **RFI** | Radio Frequency Interference |
| **RPC** | Remote Procedure Call |
| **SAE** | Society of Automotive Engineers |
| **SMTP** | Simple Mail Transfer Protocol |
| **SOF** | Start of Frame |
| **SPI** | Serial Peripheral Interface |
| **SPX** | Sequenced Packet Exchange |
| **SSH** | Secure Shell |
| **TCP** | Transmission Control Protocol |
| **TIFF** | Tagged Image File Format |
| **UART** | Universal Asynchronous Receiver/Transmitter |
| **UDP** | User Datagram Protocol |
| **USB** | Universal Serial Bus |
| **V** | Volts |

# Chapter 1

# Introduction

This chapter presents the context for this thesis by introducing the problem together with a brief background description. It explains the purpose and the goal of this thesis project and the methods used to achieve them. At the end of this chapter, a short overview of the structure of this thesis is given.

## 1.1 Background

Over the last few years, the number of electrical circuits in a commercial truck has grown from roughly hundreds to thousands [1]. Traditionally, the battery of a vehicle was used to power only the lighting, electric starting, and a few low power accessories, such as radios, fans, horn, and wipers. Apart from these circuits and the battery charging circuit, there were not many electrical or electronic systems in a commercial truck. Today, a commercial truck is full of electric circuits, almost all of the truck's functions are electrically controlled, for example Engine Control, Brake Control, Transmission control, Climate control, After-treatment control, and many more. Modern trucks are also equipped with advanced telematics and infotainment systems as advanced as those found in cars. This trend is progressing towards internet and Global Positioning System (GPS) based vehicle-to-vehicle communication.

The commercial truck industry has tried various ways to handle this sophistication in electrical and electronic systems by trying different types of Electrical/Electronic (E/E) architectures. In today's commercial truck industry, it is common to find a decentralized computation and control based E/E architecture, also known as Federated E/E architecture. This architecture breaks from the tradition of using isolated electrical systems. Instead, this architecture is based on Electronic Control Units (ECU) which have computational capabilities and an ability to control devices, such as relays, motors, valves, and lights which are connected to them. ECUs are also often called as Electronic Control Modules (ECM). These ECUs are constantly talking to each other via a vehicle network, which forms the backbone network for all the communication among the ECUs.

## 1.2 Problem description

In the recent years, the car industry has been moving at a fast pace towards automation; the new players in the automotive industry, such as Tesla [2] and Google [3], have shown partial success in self-driving vehicles. This has propelled the commercial truck industry to plan their next generation E/E architecture to support automation.

In the current E/E architecture, ECUs are responsible for both computation and control of Input/Output (I/O). However, because of increasing automation, the requirement for computational power is rapidly increasing. Moreover, for different applications (for example, Engine Control, Climate control, Brake control) there are numerous ECUs involved, each with their own I/O interfaces. Given the uniqueness of the algorithms and the desired functionality of each application, each ECU is currently tailor-made, thus increasing the design and manufacturing costs. Additionally, the number of sensors in a vehicle is increasing rapidly. The data from these sensors might be needed in multiple applications, hence needed by multiple ECUs. Therefore, the need for a higher bandwidth vehicle network is imminent.

The need for increased computational power is difficult to address with the current computational capabilities of ECUs. To address this, one approach would be to increase the computational power of the individual ECUs. Another approach could be to introduce an improved E/E architecture and vehicle network which can support the impending modifications in the vehicle's embedded systems with increasing automation in a better and more efficient way. Following the latter approach, a potential solution is to replace the current E/E architecture with a centralized computation based architecture. This allows the computational power to be concentrated at one (or a small number of) sites while decreasing design and manufacturing costs by introducing generic I/O nodes to replace the ECUs for the control of the I/Os.

## 1.3 Purpose

This thesis project examines how a centralized computation based architecture, known as *Integrated E/E architecture* could not only cater to demands for high computation performance, but also reduce the cost of ECUs. This thesis project aims to introduce an Integrated E/E architecture followed by analysis and design of low computational power based control units that can replace the current ECUs and could be utilized in an integrated E/E architecture scheme. This alternative should be cost efficient and generic so that these new units could be used to replace many different types of ECUs (further lowering design, manufacturing, and service costs).

## 1.4 Goals

The goals of this thesis project are to propose a centralized computation based E/E architecture and to design potential replacements for the ECUs that would be used in this new architecture.

## 1.5 Delimitation

The current study for replacement of ECUs is done with focus on three ECUs of the chassis domain *. Other ECUs are not taken into consideration.

## 1.6 Methodology

This thesis project started with a literature study of the various types of E/E systems involved in commercial trucks. This was followed by a study of various E/E domains and their communication protocols. A qualitative study of the existing E/E architecture was done in order to devise a new E/E architecture. Various new network and communication standards were also studied for their potential use in vehicle networks.

This was followed by a quantitative study of chassis ECUs in the current E/E architecture. These ECUs were analyzed based on their I/O capabilities in order to replace them with a generic low computation variant together with a centralized computation based architecture.

## 1.7 Structure of this thesis

Chapter 2 provides the background necessary to understand the problem and the specific knowledge that the reader will need to understand the rest of this thesis, it also talks about related work. Following this, Chapter 3 describes the E/E architecture solution proposed in this thesis project. Chapter 4 discusses the chassis ECUs and their potential replacements, which were prototyped during this thesis project. Finally, Chapter 5 offers some conclusions and suggests future work.

* Different ECU domains are detailed in Section 2.1

# Chapter 2

# Background

This chapter contains the fundamental knowledge that a reader will require to understand the remainder of the thesis. It will review the current technology that this thesis proposes to upgrade (or replace). Finally, this chapter will discuss a few solutions which others have proposed in the past to address the same problem.

## 2.1 Role of electrical systems in Commercial Trucks

The electrical systems in commercial trucks are becoming more and more prominent with time. In modern vehicles, the electrical systems account for a large part of development costs. These systems are responsible for a variety of functional and logical capabilities in the vehicle, for example, safety, communication, chassis, powertrain, human-machine interface, and various functions inside the cab [*]. Traditionally, the electrical systems were not so complex or dependent on each other. However, the current electrical systems are highly interdependent and the various systems on the vehicle are constantly talking to each other. Many different systems have access to the same sensors and actuators. One of the examples of this would be the indicator lights, in addition to being used when turning a corner, they are also used by the central locking system, emergency braking system, and are used as parking lights. In the current electrical systems, where the computation power is distributed across various physical modules on the vehicle, there is a high probability that all of these operations are performed at different locations and hence there is constant communication between different electrical systems of the vehicle.

With time, the vehicle's interior is becoming highly digital with a myriad of automated systems and gradually the automotive industry is trying to replicate the latest features of consumer electronics into vehicles, such as internet connectivity, cloud computing, swarm intelligence, and over-the-air feature updates. As these features

[*] The enclosed space of truck where the driver sits is known as cab or the cabin space

are being implemented in the vehicles, the load of computation and communication is increasing on the ECUs and the (a rather slow) vehicle network. The various details of the current E/E architecture and ECUs are discussed in the following sections in this chapter.

## 2.2 Federated E/E Architecture in commercial trucks

Today, the majority of commercial trucks use a decentralized computation based complex E/E architecture known as *Federated E/E architecture*. Different original equipment manufacturers (OEM) use slightly different versions of this architecture, but on an abstract level, the architecture is divided into four major domains: Powertrain, Chassis & Safety, Cabin & Comfort, and Infotainment & Telematics. These domains are connected to each other through different vehicular networks and each has their own subnet. Typically, the inter-domain backbone network is a high-speed Controller Area Network (CAN), while the subnet for some of the domains could be low speed CAN, for some it could be high speed CAN depending upon the size and complexity of the vehicle. An example of the functions in the different domains of a federated E/E architecture can be seen in Figure 2.1.

The powertrain domain typically consists of engine management modules, an emission after-treatment module, a body builder module, a braking module, a retarder module, and the gearbox or transmission control module. These are connected by a high-speed CAN subnet.

The chassis & safety domain typically consists of a level & roll control module, a tire pressure monitoring module, a body builder module, a battery management module, and a driver assistance module. The last of these modules is often called advanced driver assistance system (ADAS). All of these modules communicate over a high-speed CAN subnet.

The cabin & comfort domain typically consists of door modules, climate control modules, I/O modules for cabin and sleeper, access control, and airbag control. These communicate with each other via a low-speed CAN subnet as the bandwidth requirements are not very high.

The infotainment and telematics domain typically consists of the multimedia interface module, instrument cluster module, a secondary display control module, general purpose I/Os, and the Radio & Navigation module. This domain uses a high-speed CAN subnet and many modules are universal serial bus (USB) enabled to give access to the user.

Powertrain domain and chassis & safety domain are safety critical, hence there is a redundant CAN network between their modules.

All of these modules are realized through ECUs. While some of the features are realized in their own ECUs, some of them are spread over multiple ECUs. Additionally, there are cases when multiple modules are combined into the same ECUs (especially, when targeting low budget markets). ECUs are discussed in detail in Section 2.3.

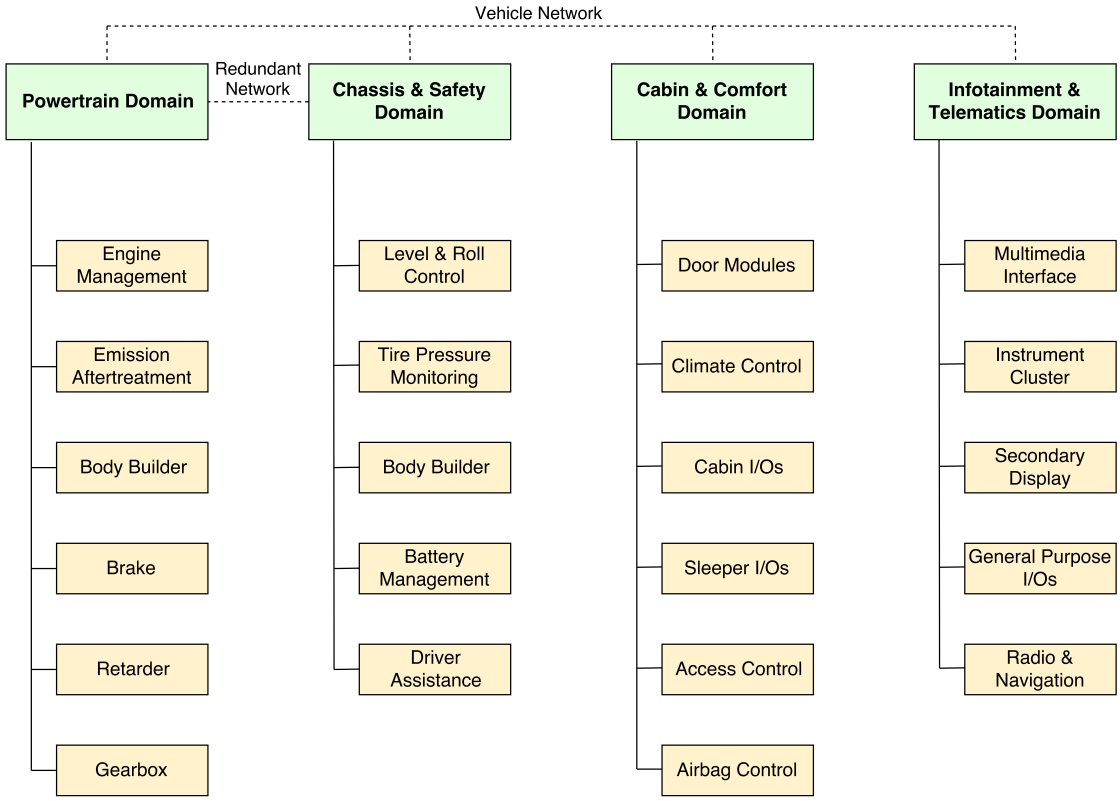


Figure 2.1: Example of a federated E/E architecture in commercial trucks

## 2.3 Electronic Control Units

In current E/E architecture, diverse topologies exist for implementing various domains and modules discussed in Section 2.2 using ECUs. The functions to be implemented in an ECU are based on its mounting location on the vehicle and the module to be realized using it. These ECUs perform computation for their local functions as well as control the I/O from/to various sensors and actuators on the vehicle. Moreover, these ECUs communicate with the other ECUs through the vehicle network. This communication is vital due to the inter-dependency of many functions within or between different domains. For example, a low fuel warning from the fuel sensor should automatically trigger the GPS and navigation system to find the closest fuel stations on the route; and turning on the wipers should also turn on the mini-head lamps; and unlocking the doors should automatically turn on the interior lights. There are many such cases where a constant collaboration between ECUs is essential for intelligent functioning of the vehicle.

The ECUs gather information from the sensors and then process the information according to the algorithms stored in them. As noted earlier, each algorithm is specialized for the intended application; for example, each of the Anti-Lock Braking System, Electronically Controlled Suspension, or After treatment module have their own algorithms which are used to achieve the required functionality. Based on the results of processing the various sensor inputs, the ECUs produce output signals which can operate various motors, valves, lights, etc. connected to them. In most cases, the ECUs communicate via the vehicle network. Each ECU has built-in self-diagnostic capabilities that enable them to collect and report their performance data and also raise alarm in case of malfunctioning.

The ECUs operate on voltages ranging from 12 volts to 48 volts depending upon the market * and the model of the ECU. These ECUs are electrically connected to the vehicle's electronics using wiring harnesses that run through out the vehicle.

## 2.4 Communication Protocols and Architecture

The two main technologies for communication within commercial trucks are Local Interconnect Network (LIN) and Controller Area Network (CAN). For future generations of vehicles, there is an ongoing development of CAN called the Flexible Data-rate (CAN-FD). In parallel to CAN-FD, the possibility of using of Automotive Ethernet is also being widely analyzed. These communication technologies are discussed in the following sections. However, before this, it is important to discuss the fundamentals of a network architecture.

### 2.4.1 Network Architecture

A network architecture defines the overall properties of the network. This includes the hardware, software, connectivity, communication protocols, and modes of transmission. The network architecture classifies the network into logical layers each with different purposes and intended uses. From a hardware perspective, it is necessary to know the hardware components used for communication, cabling, device types, network layout, and topologies, in order to realize physical and/or wireless connectivity. From a software perspective, it is essential to know the interfaces between the layers, and the protocols realized between entities at a given layer. Finally, the end user is concerned about what they see on their web browser or other web applications. To ensure that the details are well defined and to isolate the design decisions from each other, network

* The voltages used in E/E systems of commercial trucks are dependent on geographically defined markets with different voltages being used in different regions.

functions are divided into layers. The layers are stacked and structured so that each layer provides services to the layer above it.

The International Standards Organization (ISO) develops and publishes international standards. ISO introduced the Open Systems Interconnect (OSI) model of network layering in 1970s [4]. This standardized network layering into seven layers as shown in Figure 2.2.

OSI acts as a reference model for a network and facilitates understanding the relationships between various components of a network and it conceptualizes how applications can communicate over a network. The basic concept of OSI is to define the process of communication between two or more endpoints on a network. Each network user, in this case, is a computer which can realize these seven layers. In order for one endpoint to send a message, there is a flow of data down the layers in the sender's computer, across the network, and then up through the layers on the receiver's computer. This is illustrated in Figure 2.2. These seven layers are not always realized by distinct components and these functions are implemented by a combination of applications, operating systems, network hardware, and device drivers. Some details of these layers are given in following subsections.

#### 2.4.1.1 Layer 1 – Physical Layer

The physical layer provides the hardware with a means of sending and receiving data. This layer conveys the bit stream as electrical impulses, light, or radio signal through the network at the electrical and mechanical level. This layer conveys symbols which encode the frames provided by the data link layer which is discussed afterwards.

Examples of the physical layer include the media specific part of Ethernet, CAN, and V.24, along with connectors such as RJ45.

#### 2.4.1.2 Layer 2 – Data Link Layer

The data link layer works with frames, can provide switching of frames and handles the errors at the physical layer. The data link layer is divided into two sublayers: The Media Access Control (MAC) layer and the Logical Link Control (LLC) layer. The MAC sublayer controls how a node on the network gains access to the network and defines when it is permitted to transmit. The LLC layer controls frame synchronization, flow control, and error checking. The LLC may together with the MAC sublayer provide functions for addressing using MAC layer addresses.

Examples of this layer include Point-to-point Protocol (PPP), Fiber Distributed Data Interface, IEEE 802.5/ 802.2, IEEE 802.3/802.2, High-level Data Link Control (HDLC), and Frame Relay. *

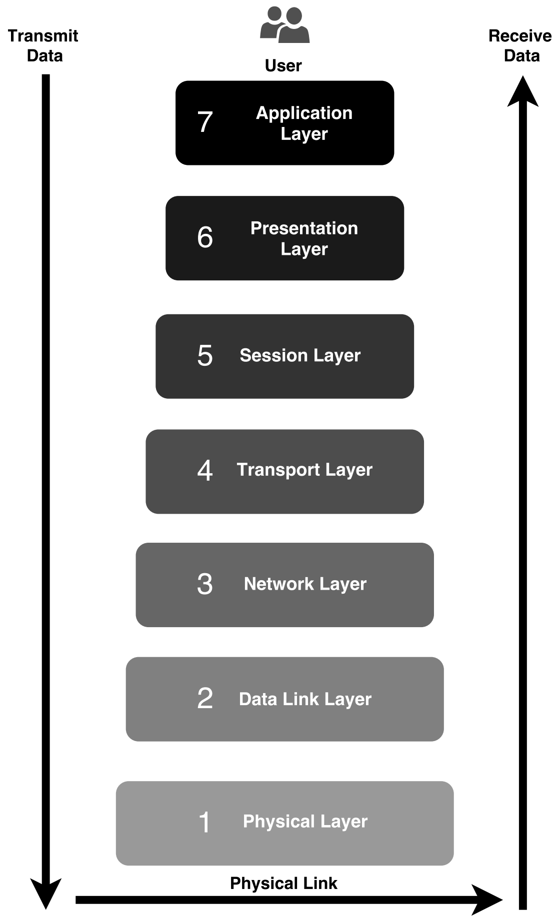


Figure 2.2: Open Systems Interconnect model of network layering

#### 2.4.1.3 Layer 3 – Network Layer

The network layer provides the routing for networks. The network layer is responsible for transmitting data from a source node to one or more destination nodes. In addition to routing and forwarding, this layer is responsible for network addressing, internetworking, and network layer error handling.

* In these examples, IEEE 802.3 (a MAC layer specification) / 802.2 (a logical link layer specification) are mentioned - this combination is what is typically thought of as ”Ethernet” - but is actually an evolution of the original Ethernet standard. This combination will be described further in Section 2.4. Further note that with a common LLC specification (in this case IEEE 802.2) it is possible to utilize a bridge to interconnect a IEEE 802.3 and a IEEE 802.5 network segment. Similarly, switches can be used, together with spanning tree protocols to interconnect segments of IEEE 802.3 networks to form a larger network.

Examples of this layer include Internet Protocol (IP) and Internetwork Packet Exchange (IPX).

#### 2.4.1.4 Layer 4 – Transport Layer

The transport layer is responsible for end-to-end data transfer between systems or hosts. It is responsible for error recovery and flow control. The transport protocol can optionally provide reliable data transmission, and thus ensure complete transfer of a set of user data (such as a file).

Examples of this layer include Sequenced Packet Exchange (SPX), Transmission Control Protocol (TCP), and User Datagram Protocol (UDP) . The latter two protocols are part of the Internet protocol stack defined by the Internet Engineering Taskforce and standardized through a series of Requests for Comments (RFC) - in this case, RFCs 793 [5] and 768 [6].

#### 2.4.1.5 Layer 5 – Session Layer

The session layer is responsible for session coordination. This layer establishes, manages, and terminates sessions between applications.

Examples of this layer include Network Basic Output System (NetBIOS), Remote Procedure Call (RPC), and Secure Shell (SSH).

#### 2.4.1.6 Layer 6 – Presentation Layer

The presentation layer is responsible for transforming data into a form acceptable and understandable by the application layer. It translates the application data to a network format and vice versa. It formats and encrypts data to be sent via the networks and deals with the compatibility problems.

Examples of this layer include encryption, American Standard Code for Information Interchange (ASCII), Unicode Transformation Format-8 (UTF), Tagged Image File Format (TIFF), Moving Picture Experts Group (MPEG), and Musical Instrument Digital Interface (MIDI).

#### 2.4.1.7 Layer 7 – Application Layer

The application layer is responsible for all the application-specific functions for the end-user layer processes. It provides user authentication, privacy, etc. This layer provides application services for file transfers, e-mail, and other high-level network services.

Examples of this layer include Telnet, Simple Mail Transfer Protocol (SMTP), Hyper-text Transfer Protocol (HTTP), and File Transfer Protocol (FTP).

### 2.4.2 Controller Area Networks

A Controller Area Network (CAN) is a serial bus communications protocol developed at Robert Bosch GmbH in the early 1980s [7]. It defines a standard for efficient and reliable communication between sensors, actuators, controllers, and other nodes in support of real-time applications. CAN is a standardized method of communication in a large variety of networked embedded control systems. CAN was initially developed for automotive systems and commercial trucks and was supported by the vehicle industry. Presently CAN can be found in passenger cars, trucks, boats, avionics, spacecraft, and other kinds of vehicles. This protocol is also widely used in industrial automation and other areas of networked embedded control, with applications in diverse products including production machinery, medical equipment, building automation, weaving machines, and wheelchairs. Within the automotive industry, embedded control has shown a trend towards networking of electro-mechanical subsystems in contrast to traditional stand-alone systems. This has resulted in highly integrated systems with modularized functionalities and hardware. This facilitates reuse of components and leads to cost savings.

#### 2.4.2.1 Hardware

CAN is a multi-master system in which multiple nodes may initiate transmissions. It supports deterministic bus access timing and automatic handling of node and hardware failures, which makes it a reliable and predictable communication technology. These properties are especially important in automotive industry applications. CAN is a low-cost network which supports both broadcast and multicast communication.

CAN bus uses a simple, differential copper wire connection. It uses two or three copper wires. CAN High (CAN_H) and CAN Low (CAN_L) transmit different voltage levels, while there is an optional CAN Ground (CAN_GND). The cables are twisted to reduce the probability of transmission errors. CAN_GND is optional because the low-speed CAN bus in cars usually uses the vehicle's body as ground. The bits sent during CAN communication are represented by differential voltage levels and use non-return to zero (NRZ) coding.

CAN uses a linear topology and the physical medium is shared by all of the nodes. Figure 2.3 shows a typical CAN bus topology.

CAN bus uses Carrier Sense Multiple Access/Collision Resolution (CSMA/CR) protocol which defines the behaviour to resolve concurrent transmissions. When a node is transmitting its frame, it listens to the bus to confirm if the ongoing frame is its own frame. If it turns out to be its own frame, then it keeps possession of the bus and continues to send its frames. If the frame is different, then the node releases the bus. CAN bus supports transmission of two-bit levels, logical 0 bits are dominant bits and Logical 1 bits are recessive bits. When two-bit levels are transmitted

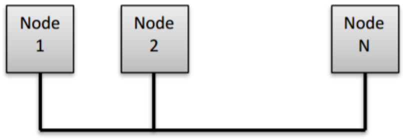


Figure 2.3: CAN topology

concurrently, dominant bits overwrite recessive bits. CAN bus implements a logical AND for concurrently transmitted bits. It implements CSMA/Bit Arbitration - which is used as part of CSMA/CR as illustrated in Figure 2.4.

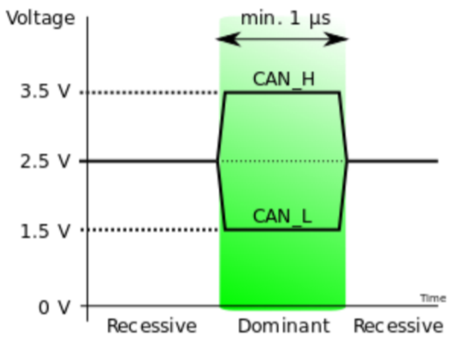


Figure 2.4: A CAN differential bus

CAN bus collision resolution adds bit timing requirements, so the bit transmission over CAN requires twice the maximum propagation delay of the bus segment. As a result, CAN's minimum bit time and the resulting maximum transmission rates depend on bus length. The different CAN bus lengths and resulting maximum bit rates are shown in Table 2.1.

Table 2.1: CAN bus length at several bit rates [8]

| **Bit Rate (kbit/s)** | **Bus Length (meters)** |
|---|---|
| 1000 | 25 |
| 800 | 50 |
| 500 | 100 |
| 250 | 250 |
| 125 | 500 |
| 50 | 1000 |
| 20 | 2500 |
| 10 | 5000 |

#### 2.4.2.2 Protocol

The CAN protocol standardizes the physical and data link layers of the ISO-OSI model as illustrated in Figure 2.5.

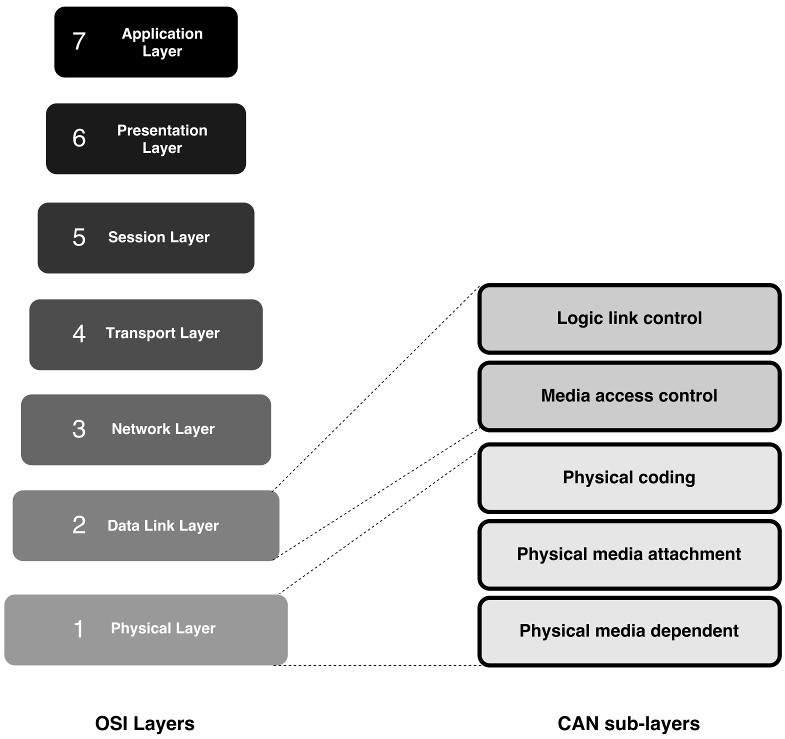


Figure 2.5: CAN sublayers with reference to ISO OSI model

The CAN physical layer is divided into three parts: physical coding (implemented by the CAN controller chips), the physical media attachment specifies the transceiver characteristics, and the physical media-dependent sub-layers, which are application-specific.

The interface between physical coding and physical media attachment is called the attachment unit interface and this is the interface between the CAN controller and CAN transceiver chip. The interface between physical media attachment and physical media-dependent sub-layers is called the medium-dependent interface and is the interface to the physical bus-lines [8].

The CAN Message frame format is shown in Figure 2.6. In a typical case of ECU communication the frame has the following bits [7]:

- The Start of Frame (SOF) is a dominant 0 to tell all the other ECUs that a message is on the way.

- The Identifier field usually contains a functional address or the source and destination addresses. This field also determines the message's priority. In standard CAN the identifier value is 11 bits, while in Extended CAN it is 29 bits.

- Using the remote transmission bit, an ECU can request the identifier from another ECU.

- The Data Length Code (DLC) field specifies the length of the data field in bytes and can be between 0 to 8.

- The data field has the information which is to be transmitted.

- The Cyclic Redundancy Check (CRC) is a 15-bit sequence used for checking the integrity of the data. In order to carry out the CRC calculation CAN uses the generator polynomial: $X^{15}+X^{14}+X^{10}+X^{8}+X^{7}+X^{4}+X^{3}+X^{0}$.

- The Acknowledge (ACK) slot is a single bit generated by all receiving CAN nodes to indicate that their CRC processes are OK. Because of the way the bits are coded any node transmitting a zero bit (indicating that they are not acknowledging the frame) would cause the transmitter to see a negative acknowledgment.

- End-of-Frame (EOF) is a seven-bit recessive sequence at the end of the CAN frame. This, along with the minimum of 3 bits of Interframe Spacing (IFS), give sufficient time delay to ensure that all nodes can distinguish between different frames and hence between CAN messages.

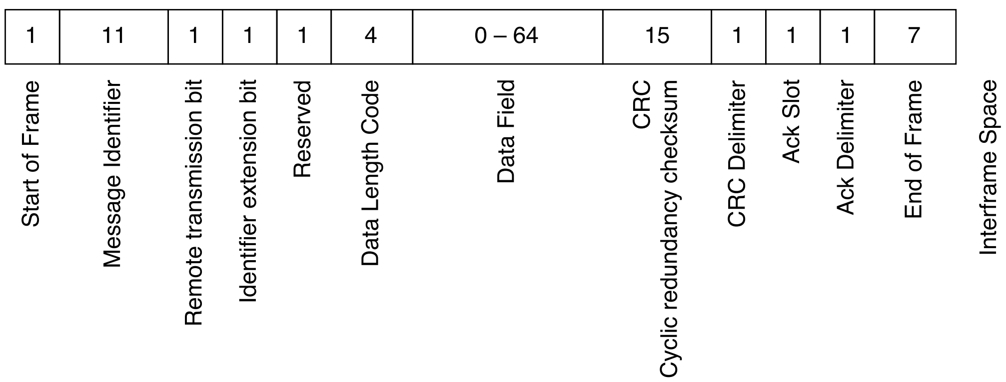


Figure 2.6: CAN data frame

Since CRC and ACK fields have a one-bit delimiter to allow a space for the next process. The recessive delimiter bits ensure that there are bit transitions in the fields that do not have bit-stuffing applied. The bit transitions are necessary to recover timing synchronization that might not be otherwise available due to NRZ encoding.

For efficient development and operation in most systems, higher-layer protocols are needed. In order to achieve this and facilitate the adoption of CAN in many other application fields, ISO and Society of Automotive Engineers (SAE) defined various protocols based on CAN. These higher-layer protocols include CANopen [9], CAN Kingdom [10], DeviceNet [11], and J1939 [12] targeted for truck and bus applications

Introducing networks into vehicles also makes it possible to more efficiently carry out diagnostics and to coordinate the operation of the separate subsystems. For example, various diagnostic systems such as Onboard Diagnostics (OBD) and European Onboard Diagnostics (EOBD) enable monitoring of emissions, mileage, speed, and other useful data. This system is connected to the Check Engine light, which illuminates when the system detects a problem. This has been facilitated by the presence of a vehicle network. There are various advanced diagnostic systems available in current vehicles such as Enhanced On-Board Diagnostics, Second Generation (EOBD2), and OBD2/OBD-II.

#### 2.4.2.3 Drawbacks

It can be seen that everything on a CAN bus network is dependant on one cable. This dependency on a single cable has its own problems, and in many cases, a redundant CAN network has to be installed, hence adding to the cost and the vehicle weight. CAN also presents installation problems, as the length of cable could not be arbitrarily set and needs to be coordinated with the required bit rates required by the system.

### 2.4.3 Local Interconnect Network

The Local Interconnect Network (LIN) bus was developed to create a standard for low-cost, low-end, multiplexed communication in automotive networks during the 1990's by an industry consortium [13]. Although the CAN bus addresses the need for high-bandwidth, advanced error-handling networks, the hardware and software costs of a CAN implementation were considered prohibitive for lower performance devices such as power windows, seat controllers, doors, mirrors, light, etc. In contrast, LIN provides cost-efficient communication in applications where the bandwidth and versatility of CAN are not required. LIN can be implemented relatively inexpensively using the standard serial Universal Asynchronous Receiver/Transmitter (UART) embedded into most modern low-cost 8-bit microcontrollers.

Modern automotive networks use a combination of LIN for low-cost applications (primarily in body electronics), and CAN for mainstream powertrain and vehicle body communication. The LIN bus uses a master/slave approach with a LIN master and one or more LIN slaves.

#### 2.4.3.1 Hardware

LIN operates on a nested bus which is used to connect devices to a single CAN bus master. LIN is a one wire bus which uses NRZ coding. It uses byte-wise transmission using start and stop bits and supports transmission rates up to 19.2 kbps while providing a usable data rate up to 800 byte/s. LIN supports a bus length up to 40m. A typical LIN network is shown in Figure 2.7

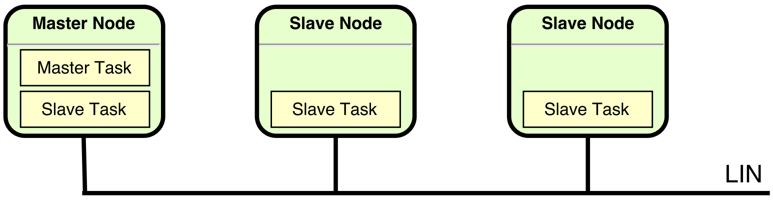


Figure 2.7: LIN topology

#### 2.4.3.2 Protocol

LIN works on single-master/multiple-slave network. Medium access is controlled by the master node. The nodes may implement master and slave software tasks. The master node can also realize a gateway to other networks, such as CAN. A LIN network can have 1 Master node and up to 16 slave nodes. The master periodically polls the slave nodes to check for messages. Each of these messages is identified by a unique identifier that is assigned to every node. Similar to CAN, LIN does not use node addressing, hence LIN nodes decide whether to receive a message or not based on the ID of message.

In LIN transmission, the master node divides the communication medium into time slots of fixed, predefined length. The LIN master transmits a request frame with a defined frame ID called the polling frame in a predefined transmission slot. The LIN slave node responsible for that particular transmitted frame ID responds in that transmission slot.

A typical LIN frame is shown in Figure 2.8. It begins with a sequence of at least 13 dominant bits for signalling bit arbitration by the master node, which is followed by at least one recessive bit. This unique bit sequence is called *Sync Break* and is recognized by all slaves, hence all nodes are informed about the frame start. The bit sequence of 01010101 is transmitted for synchronizing the bit clock of slaves. Following the identifier from the LIN master, the slaves can transmit a message of up to 8 bytes length in a single frame. There is a fixed space between the frames known as Inter-byte space (IBS). The lengths of frames are statically configured for each message type. The frame ends with a checksum, the space between two frames is known as the Inter-Frame Space (IFS).

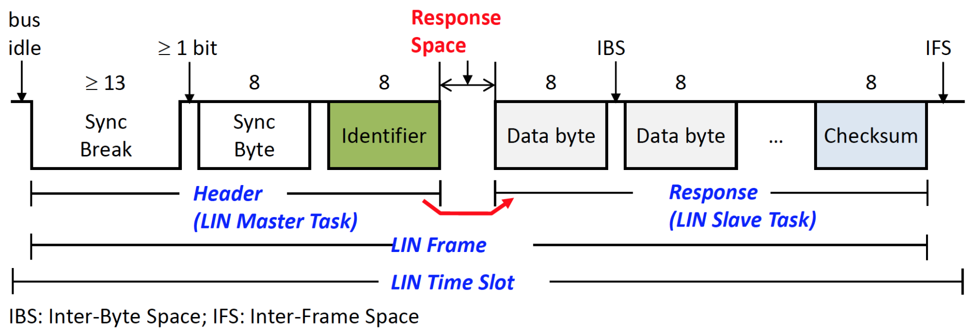


Figure 2.8: LIN frames transmission

#### 2.4.3.3 Drawbacks

The main drawbacks of LIN that make it a less effective bus access scheme are the master-slave configuration and low bandwidth.

## 2.5 Ethernet

In the late 1970s, Ethernet was formalized by Xerox Corporation, Digital Equipment Corporation, and Intel Corporation in a document titled *The Ethernet, a Local Area Network: Data Link Layer and Physical Layer Specifications* [14] [15]. Subsequently, IEEE released the IEEE standard 802.3 [16]. and this lead to a family of IEEE standards referred to as IEEE 803. Some members of this family of standards are:

| | |
|---|---|
| **IEEE 802.1** | Overview, Architecture, Interworking, and Management [17] |
| **IEEE 802.2** | Logical Link Control (upper part of data link layer) [18] |
| **IEEE 802.3** | Ethernet CSMA/CD; MAC and PHY |
| **IEEE 802.4** | Token Bus; MAC and PHY [19] |
| **IEEE 802.5** | Token Ring; MAC and PHY [20] |

In IEEE 802.3, to arbitrate access to the shared channel a method called carrier sense multiple access with collision detection (CSMA/CD) was employed. IEEE 802.3 defines the basic medium access control and frame formats and laid the foundation for modern Ethernet networks.

### 2.5.1 Hardware

Over the years, various types of physical media have been used for Ethernet. With the increasing bandwidths, the types of media used have evolved.

Initially, the standard 10Base5 (also known as *Thick Ethernet*) was used. This standard specified a 9.5 mm coaxial cable. In 1985 a new standard called 10Base2 (also known as *Thin Ethernet*) was developed using a thinner coaxial cable. After 1990, this further developed into various forms of twisted pair cables, which are still in use. In 1990, 10BASE-T was developed, which used two twisted pairs and supported up to 100 MHz symbols using Manchester encoding. This was followed by 100BASE-TX in 1995, which used 4B5B encoding [21], thus it mapped 4 data bits into 5 code bits. In conjunction with the introduction of twisted pair physical media, the standard also enabled full duplex communication. This was upgraded to 1000BASE-T in 1999, which was a standard for Gigabit Ethernet [22]. Gigabit Ethernet supports frame transmission of up to one gigabit per second for distances of up to 100 meters. This was upgraded to 10GBASE-T which supports data rates of up to ten gigabit per second.

### 2.5.2 Protocol

Ethernet is based on CSMA/CD, when using switched Ethernet, the medium is not shared, hence the two endpoints can send data over the cable freely if the two endpoints are operating in full-duplex mode. This also eliminates the need for an IFS. A generic Ethernet frame is shown in Figure 2.9

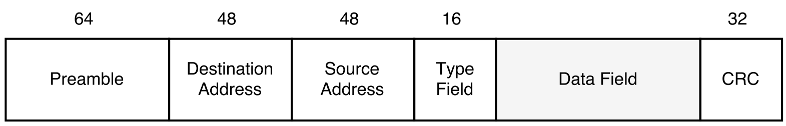


Figure 2.9: Generic Ethernet Frame

The fields are:

| | |
|---|---|
| **Preamble (64 bits)** | This allows the destination interface to synchronize its clock with source interface |
| **Destination Address (48 bits)** | This contains the MAC address of the destination |
| **Source Address (48 bits)** | This contains the MAC address of the source |
| **Type Field (16 bits)** | This contains the type of frame |
| **Data Field (368-40000 bits)** | This contains the packets to be transmitted |
| **CRC (32 bits)** | A CRC value for error checking |

### 2.5.3 Automotive Ethernet

Although Ethernet was a successful protocol and is widely used in many different types of applications, it has not been used for automobiles for several reasons [23]:

- Ethernet wiring has double the number of pins and wires compared to CAN, hence doubling the wiring costs.
- Ethernet emits too much Radio Frequency (RF) noise and is susceptible to noise from other devices in the vehicle. Hence it does not meet the Electromagnetic Interference (EMI) and Radio Frequency Interference (RFI) requirements for the automotive market.
- For the fast reaction time of some sensors and controllers, latency should be in the low microsecond range, which Ethernet could not guarantee.
- There was no way to control the bandwidth allocation to different streams. This was a hurdle to transfer data from multiple types of sources.
- Ethernet uses special connectors resulting in additional cost.

These limitations of Ethernet lead the automotive industry to work with different standards, the introduction of many networks for different devices in a vehicle added to overall wire harness costs, vehicle weight, and complexity. Today, Automotive Ethernet, which uses 100Base-Tx is used to communicate from cameras for driver assist and for infotainment video displays [24]. The current technology, Broadcom's BroadR-Reach PHY [25] meets the EMI requirements. Today this PHY technology is being adapted for various new communication needs in the car industry; however, in the commercial truck industry; this technology is yet to be adapted in most of the E/E domains.

### 2.5.4 One-Pair Ethernet (OPEN)

As discussed in Section 2.5.3, Broadcom developed BroadR-Reach, a standard to address the EMI requirements of Ethernet. This technology fulfills the EMI requirements by using a lower bandwidth of around 27MHz. It uses Pulse Amplitude Modulation-3 (PAM-3) signalling with better encoding and echo cancellers to enable transmission of bidirectional data on a single pair of copper wire. It is currently licensed as OPEN and compliant components are available from multiple vendors. OPEN can allow support data transmission at 100Mbits/s, which is sufficient for video transmission; however, to act as the backbone of a vehicle network, higher data rates are required. To address these requirements a new upcoming standard 1000Base-T1 [26] has been introduced.

## 2.6 Automation in commercial trucks

The commercial truck industry, mainly heavy goods vehicles, are simultaneously facing a number of challenges, for example safety, hours of service, driver shortage, and fuel costs. Introducing automation in trucks can address many of these issues and is discussed in this section.

### 2.6.1 Industry issues addressed by automation in commercial trucks

Currently, the commercial truck industry is facing a lot of challenges including: [27]

**Hours-of-service** Automation can lead to optimization in the resting times for the drivers, and for trailing vehicles in case of platooning *.

**Driver Distraction** Autonomous technology can contribute to compensate for the driver's lack of attention.

**Driver Shortage** Driving in platoons can decrease the need for drivers; moreover, a change in the role of driver can potentially attract younger drivers. This can also lead to driver retention by lowering driving stress because of monotonous time periods.

**Fuel Costs** Mileage can be improved through better control, and in the case of platoons through better aerodynamics.

In addition to these issues, automation can address many social issues as well, for example:

**Emission Reduction** Autonomous and efficient control of trucks will lead to reduced emissions.

**Congestion** Shorter distances between trucks will help to reduce congestion.

**Safety** Automation can increase the safety by a huge margin as it will reduce the human contributions to driving which is one of the biggest contributors to accidents [29].

* Platooning refers to grouping vehicles into platoons in order to increase the capacity of roads. Platoons decrease the distances between vehicles using electronic, and possibly mechanical coupling. [28]

### 2.6.2 Different levels of vehicle automation

The Society of Automotive Engineers (SAE), a standards developing organization, developed a SAE standard J3016 [30], which classifies vehicle automation into 6 levels or stages:

**Level 0 – No Automation:** The driver is fully engaged and in complete control. The driver is responsible for braking, steering, throttle, and motive power. Some passive warning signals, such as lane-departure warning or blind spot monitoring system may be present.

**Level 1 – Driver Assistance:** The vehicles in level 1 can have individual functions which are automated; however, these automated functions are not connected to each other. Adaptive cruise control, front-collision avoidance, and lane keeping assist are some of the examples of such independent systems. The driver can be either "feet off" or "hands off" at once.

**Level 2 – Partial Automation:** The vehicles in level 2 can have multiple automated primary-control functions active at the same time. An example could be the combination of adaptive cruise control and lane keeping assist simultaneously. The driver can be "feet off" and "hands off" at the same time, but eyes must stay on the road.

**Level 3 – Conditional Automation:** The vehicles in level 3 exhibit automation of multiple functions and the driver has to respond to a request to intervene. This is applicable under some particular traffic conditions, where the vehicle makes the decision to transfer control back to the driver based on the surroundings. The driver can be "feet off", "hands off", and "eyes off".

**Level 4 – High Automation**: The vehicles in level 4 would be totally automated in certain conditions, such as driving on highways, when the driver is not expected to monitor the road. The driver has no responsibility during the automated mode.

**Level 5 – Full Automation:** The vehicles in level 5 are expected to have situation independent automated driving, the driver's sole responsibility is to enter the destination.

The present technology has reached level 2 and is on market by some car vendors, such as Tesla. There is a high possibility that Level 3 vehicles might not enter the market because of safety concerns in transferring control and there could be a direct jump to Level 4 and 5. Hence the E/E architecture of next-generation must be ready to facilitate level 4 automation.

## 2.7 What has already been done?

This section discusses previous works which have suggested solutions similar to that proposed in this thesis project.

### 2.7.1 Work done on E/E Architecture

In an article by Paul Hansen [31], detailing the plans for changing the electrical architectures by BMW and Audi, it is discussed that the car makers are developing end-to-end architectures with hardware-software decoupling. The proposed new architecture is called *Central computing platform* by BMW and *Central computing cluster* by Audi. They aim to have one or more central computers that do all the computation, which make up the backbone of a scalable architecture. This is done with an aim to have a single software platform by 2025, which is capable of vehicle automation and is common to all the central computers.

Daimler announced their plans in 2016 to standardize the E/E architecture across all the truck and bus platforms across its multiple brands [32].

In their 2015 master's thesis project [33], Jonas Hemlin and Dan Larsson proposed to analyze the Generic Timer Module (GTM) from Robert Bosch to perform all timer-module tasks in Volvo Group Truck Technology powertrain ECUs. They used Volvo's existing ECUs as the reference system. These ECUSs currently which utilize another timer module, the enhanced Time Processing Unit. Their results show that GTM can perform the required tasks.

In their paper published in 2013 [34], Hauke Stähle and his co-authors proposed a centralized ICT architecture for future vehicles by the analysis of a potential system, hardware and software architecture, and their properties. The talked about technologies including drive-by-wire systems or advanced driver assistance systems. They tried to bring the previously described ICT architecture one step closer to a large-scale implementation in the automotive domain. Demonstrators for proof-of-concept and evaluation are also discussed.

### 2.7.2 Work done on vehicle network upgradation

In his 2011 master's thesis project [15], Muhammad Ibrahim studied whether Ethernet could be a solution to complement or replace CAN, thus overcoming CAN's limitations. He performed this project for CPAC Systems, a company in the Volvo group which develops and manufactures steer-by-wire systems based on CAN technology. The results of his work showed that Ethernet in combination with Time Critical Network's modeling tool, (for time-determinism) can be a complement and/or an alternative to the CAN bus.

In his 2014 bachelor's thesis project [35], Rasmus Ekman studied replacing CAN networks with Ethernet in vehicles, Additionally, he studied the possibility of using power over Ethernet to reduce the overall weight and components of a vehicle. His thesis shows that an Ethernet-based system can serve as a possible replacement candidate for the CAN system due to Ethernet's low latencies and high bandwidth.

In his 2017 master's thesis project [36], Ammar Talic analyzed the security of automotive Ethernet, in order to be considered as the backbone network for the vehicle. He predicts that despite of needs for some improvements, a the rate at which automotive Ethernet related protocols are undergoing standardization and development, it can soon be a good protocol to be used by the automotive industry.

# Chapter 3

# Integrated E/E Architecture

This chapter discusses a centralized computation based E/E architecture for the commercial trucks, known as *Integrated E/E architecture*. This architecture aims to achieve a hardware-software decoupling in the vehicle's electrical systems. The aim is to consolidate the majority of application software of the vehicle into one (or few) place(s). In order to exploit modern technological advancements, it is proposed to replace the current ECUs with different types of nodes, these nodes could be computation nodes, I/O nodes, or custom nodes. This architecture aims to move the computation function to one or more centralized node(s) connected to configurable I/O nodes. This creates an opportunity for specialized and generic I/O (GIO) nodes, which can be configured for various functions without having each of them tailor-made for different applications.

In the current electrical systems, which are realized using ECUs, the ECUs have the software components embedded into them. Which gives rise to many technical, financial, and/or logistical problems. Usually, the ECUs are developed by Tier-1 suppliers*, who design them based on the requirements of the vehicle maker. When the vehicle maker has to change supplier, they not only get a new hardware platform, but also a different software platform, which needs to be re-validated and tested. This also hinders the possibilities of reusing the application software in different functions.

By separating the computational and I/O functions into different nodes, in addition to hardware-software decoupling, several other important and positive effects can be realized:

- The necessary computational power can be concentrated in a small number of ECUs (henceforth referred as *computation nodes*) ,
- The GIO nodes will be generic from a computational node's point of view, hence the algorithms for performing I/O operations can be reused,

* Tier-1 suppliers are the companies who supply parts or components to the OEMs. Some examples are Bosch, Continental, and Delphi.

- The software in the GIO node can be generic since it does not need to be different for each node (although configuration of "parameters" for a given node may be necessary), and
- Each GIO node will be tested once and then can be used to realize different types of nodes, in contrast to the tailor-made ECUs, each of which needs to be tested separately on a per ECU type basis.

The Integrated E/E architecture may not necessarily be 'centralized', the integrated platform would still be distributed, especially when controlling the sensors and actuators. These sensors and actuators will be multiplexed using less intelligent GIO nodes. The ECUs attached to the sensors and actuators in earlier architecture were responsible for both 'computation' and 'control'. In contrast, in the Integrated E/E architecture, the GIO nodes, which replace the current ECUs at their respective physical locations, would only be responsible for 'control' based on the 'computation' done at the computation node(s).

There can be numerous advantages of using an integrated E/E architecture along with GIO nodes which will be discussed in next chapter.

## 3.1 Nodes in a Integrated E/E architecture

In order to simplify the architecture and to integrate the modern features of automation, there has been an initiative to move towards a more centralized E/E architecture with regards to computation. The aim is to have either one computation unit in every electrical domain or one large computation unit for the entire vehicle. This architecture will result in a cost reduction and enable better integration of functions across domains.

Using the integrated E/E architecture for commercial trucks aims to utilize high-performance computational nodes in contrast to the distributed ECU based computational approach of federated E/E architecture. This will not completely replace the current setup of multiple ECUs across the vehicle, but lead to many changes in the hardware configuration. The introduction of a high cost, high computational power based computational node enables ECUs to be replaced by low power, low-cost GIO nodes.

A possible realization of the proposed integrated E/E architecture scheme is shown in Figure 3.1. This architecture proposes to use two computation nodes interconnected by a high-speed network. The second computation node is dedicated to performing redundant computation for safety-critical operations.

In this scheme, the ECUs are replaced by GIO Nodes in as many modules as possible, while some ECUs may still be used. The different components of this architecture: computation node, GIO nodes, custom nodes, and network switch are explained in the following sections.

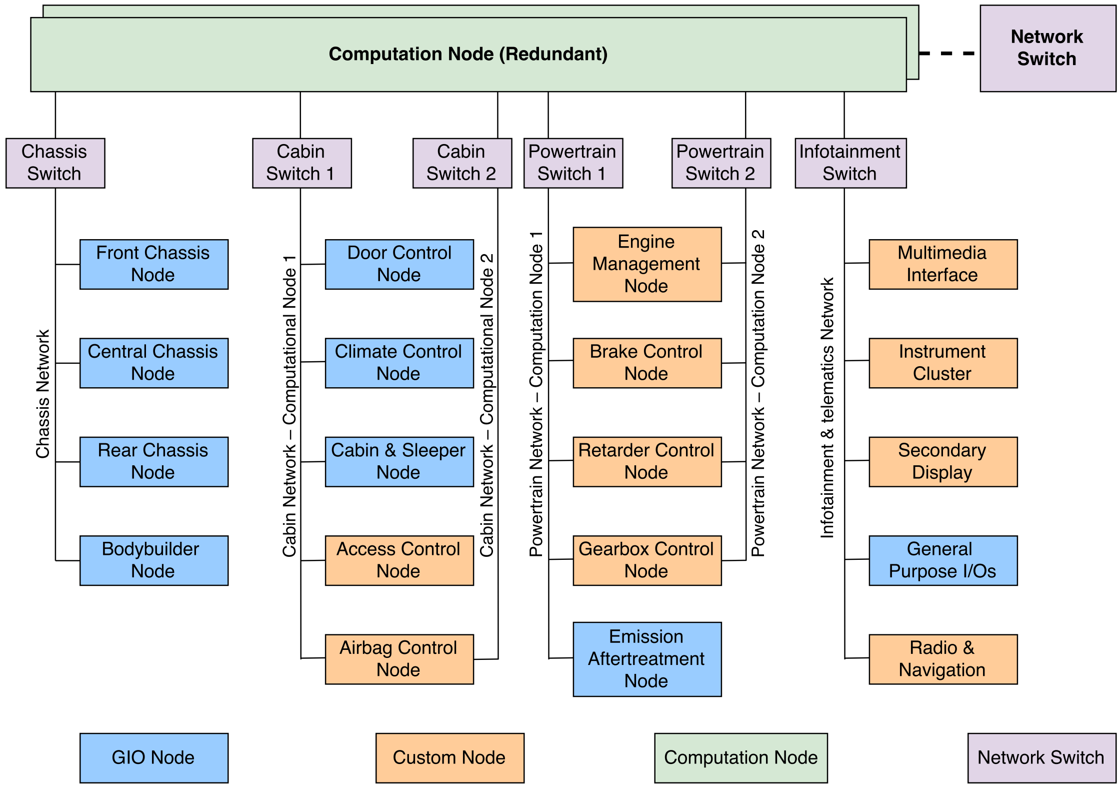


Figure 3.1: Example of the proposed partially redundant integrated E/E architecture for commercial trucks

## 3.1.1 Computational node

The computational node is a centralized high computation capacity node. It is capable of supporting all of the computational needs of a vehicle, at the same time thus enabling various types of vehicle automation. This role requires some level of redundancy; for example, having another computation node which redundantly performs all safety critical tasks to ensure the fail-safe operation of the vehicle's electronics. This redundancy also requires the computational nodes to be physically distributed about the vehicle so that even in the event of an accident, there is a low probability that all of these computational nodes would fail at the same time. Note that these computational nodes are connected to each other via a high speed network.

## 3.1.2 Custom node

Custom nodes are inherently similar to the present ECUs and they are expected to be used for functions which are safety critical; for example braking, steering, engine control, etc. These nodes will maintain their independent computational capabilities;

however, their communication capabilities will be enhanced to enable them to frequently communicate with the computational node at a higher bandwidth.

### 3.1.3 GIO Node

In the proposed E/E Architecture, nearly all computation is done at the computational node(s). However, to interact with sensors and actuators, there is a need to place the I/O controller close to the physical location of the sensors and actuators. This minimizes the length of the wiring harness, while reducing the overall weight of the vehicle. This I/O control is performed by the GIO nodes, which undertake all the I/O operations previously performed by the ECUs at that position.

Currently, all the ECUs are tailor made for their specific functionality, thus adding to the overall manufacturing costs. This also increases service costs as the vehicle is inoperable until a malfunctioning ECU is replaced. GIO nodes address these problems by shifting the high computation load away from all of the I/O nodes to the computational node(s). Additionally, these I/O nodes can be designed to be interchangeable, hence the same type of GIO node can be used in multiple locations and for multiple purposes. In this way, OEMs can replace ECUs with GIO nodes connected to the computational node. The GIO nodes will be configured according to the desired functions to perform them in *cooperation* with the computation node(s).

Next chapter focuses on defining the specifications and producing a design of the GIO node.

### 3.1.4 Network Switch

The integrated E/E architecture aims to utilize one or more high speed vehicle networks. In order to utilize a star topology for the various high speed networks, a network switch should be placed in every domain of the vehicle. Moreover, to support redundant computation nodes, these network switches should be connected to to the computational node(s). In several cases it is desirable to also have a redundant physical network and switch for a given domain. This switch will also have connections to each of the computation nodes.

## 3.2 Network infrastructure for next generation E/E architecture

The E/E architecture aims to support complete vehicle automation of Level 5, hence the communication backbone needs to be more capable then it presently is. In trucks driven by human drivers, the majority of sensors require a very low network bandwidth. But when realizing the more advanced automation features of level 4 and 5, the network backbone needs to be of a much higher bandwidth. To accommodate high bandwidth data from various cameras, LiDAR, radar sensors, etc., it is necessary to upgrade the current CAN based vehicle network to the more advanced and faster Automotive Ethernet. Currently, some high bandwidth features use automotive Ethernet in the cab of the trucks for infotainment and communication purposes. However, the architecture for complete automation needs to support the following features: [37]

**Vehicle-to-X communication** This functionality is aimed at having an intelligent transport systems, with vehicles communicating to all of the various infrastructure systems and vice versa. The aim is to provide a better and more accurate awareness of the traffic on the entire road network and individual highways.

**Fully Automated driving** In this scenario with situation independent autonomous driving, the driver's role is near negligible.

**Road trains platooning and cross docking** Platooning aims to decrease the distance between vehicles by the means of electrical coupling resulting in decreased congestion. Cross docking means that trucks could exchange trailers while moving on the highways to reduce the storage or transfer time.

**Green corridors and electro-mobility** This aims at having the option of powering trucks with electricity supplied directly from the lines placed above or beneath them on motorways - similar to trains or trams.

Inspired by other industries, a strong architectural trend that the automotive industry seems to be adopting is integration. An integrated E/E architecture intends to consolidate as much application software as possible in an ECU. In an ideal case, there would be one vehicle computer that executes all vehicle functionality. The transition from 'control' ECUs to 'compute' ECUs for autonomous driving also fits well with the integrated pattern where a powerful vehicle computer executes all application logic. As noted previously, an integrated E/E architecture may not necessarily be 'centralized'. The integrated E/E platform could still be distributed, especially when sensing and actuating are well distributed throughout the vehicle's geometry and needs to be multiplexed using less intelligent ECUs.

This section discusses an advanced integrated E/E architecture which can be deployed in commercial trucks by 2030. This architecture is highly scalable and has the ability to use the same architecture in trucks of all sizes and to reuse the same system software. It is similar to the modern day computers and smart-phones in which devices have different configurations but the overall architecture is coherent for each brand. Moreover, all of the brand's devices use the same operating system, for example MacOS, Windows, or Android. An abstract form of this architecture is shown in Figure 3.2. In the present scenario, it is not easy to upgrade the vehicle's backbone network due to the cost factors and the fact that this would mean a hardware upgrade for all of the ECUs currently communicating over CAN. However, once the majority of functions are moved towards the central computation unit(s), it would be necessary to replace all the ECUs with new hardware; hence there is an opportunity to use GIO nodes for the majority of functions. In this situation the backbone network of the vehicle can be upgraded to Automotive Ethernet which would increase the network's bandwidth to the Gigabit per second range. Currently, the use of One Pair Ethernet by Broadcom looks attractive.

This architecture aims to have two fully redundant computation nodes in order to ensure an error-free and fail-safe functioning of the automated functions of the vehicle. A high speed gigabit Ethernet network backbone facilitates performing even closed loop operations at the computation node. However, the GIO nodes performing critical operations will have the capability to perform these operations in an event of communication loss with the computation node to ensure a fail-safe-operational functioning for such operations. The decoupling of hardware and software can ensure a single scalable software that reuses the tested functions from one truck model to another without needing rework, which will reduce the development costs and increase the reliability of software (a scenario similar to commercial computer software).

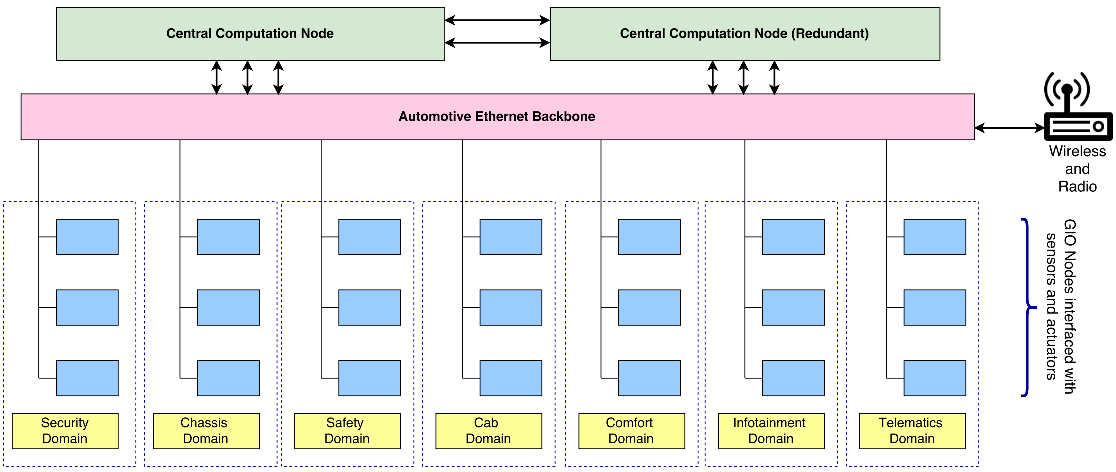


Figure 3.2: Future concept for next generation Integrated E/E Architecture

# Chapter 4

# Generic I/O Nodes: Analysis and Design

This chapter discusses the analysis and design of the federated E/E architecture with a focus on the ECUs in the chassis domain. The configurations of three ECUs used in this domain in the existing generation of trucks are analyzed in order to replace them with GIO Nodes. Based on the result of this analysis, the configuration and technical specifications for GIO nodes are proposed. A prototype of GIO node was implemented as a part of this thesis project.

Finally, an advanced version of a GIO node is proposed which would be suitable to be used in the next generation E/E architecture discussed in Section 3.2.

## 4.1 Architecture of GIO Nodes

For the purpose of this thesis project, consider that the ECUs in the chassis domain of a truck are to be replaced with GIO nodes. Presently, most commercial trucks have either two or three chassis ECUs; specifically, front chassis, central chassis, and rear chassis modules named according to their mounting position on the chassis. In some trucks, the central chassis ECU also performs the functions of the rear chassis module.

### 4.1.1 Multi-module design of GIO Nodes

All of these modules are analyzed with regards to their I/O capabilities in order to decide upon the optimal number of I/O interfaces for the GIO node. However, no redistribution of functions is done from the current chassis ECUs, hence the new GIO node could replace a given ECU on a one-for-one basis. These GIO nodes are expected to be configured by the computational node either at the time of installation or dynamically at run-time. This thesis project proposes a split-design architecture for GIO Nodes. This is done to address hardware scalability issues. The cost driving factors of the design process are circuit design, housing design, and connector design. The manufacturing costs are highly affected by the number of units to be manufactured, hence a reusable generic design can considerably reduce these costs. For every ECU replaced by a GIO node on a truck, the number of units of GIO node increases while the number of unique ECUs decreases, thus as noted earlier reducing design and manufacturing costs. It is proposed to use a modular two board configuration as illustrated in Figure 4.1, this maintains genericness and facilitates scalability.

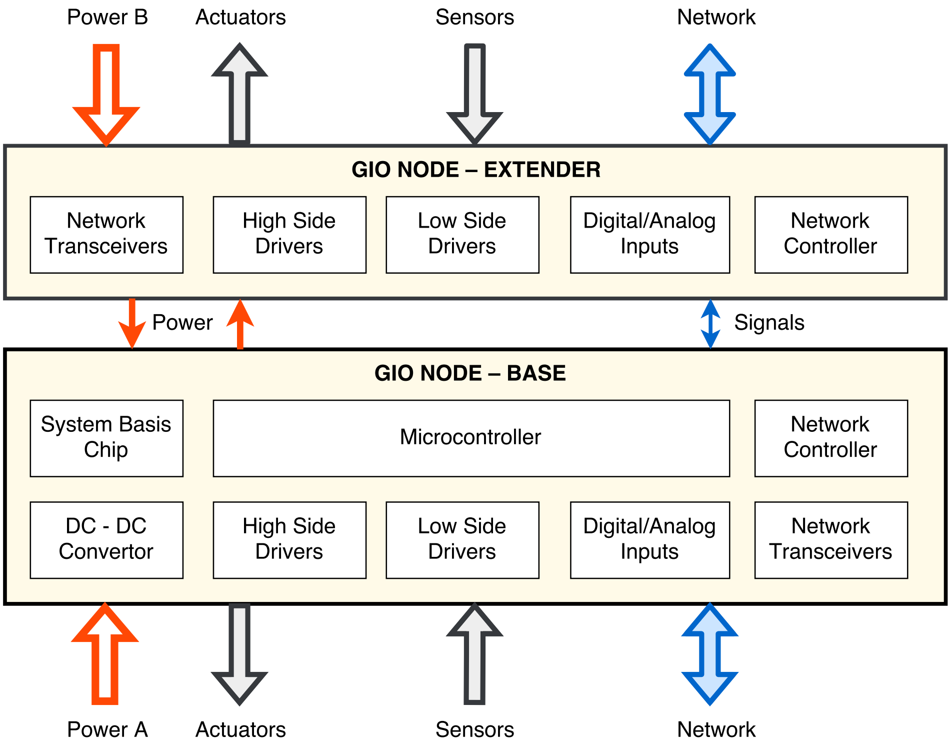


Figure 4.1: GIO node architecture

### 4.1.2 GIO Base board and Extender board

To realize the GIO Nodes as two modules; two electronic circuit boards are stacked on top of each other. The main circuit board is called the "GIO Node – Base", referred to henceforth as the *Base board*. On the base board, all the major components required for computation are mounted; specifically, the DC-DC converter to step down the supply voltage of the truck to provide the desired voltages for the GIO node, the microcontroller, a network controller for communicating with another circuit board, the network transceivers for communicating with other GIO nodes and computation node(s). The digital and analog inputs along with high side * and low side † drivers (outputs to actuators) can vary for various mounting positions of the GIO nodes, hence there is an optimal number of inputs and outputs to place on the base board. The DC-DC converter steps down the high DC voltage to the desired lower voltages used by the electronics in the node using various voltage regulators, thus also serving as a power supply for various sensors that operate at lower voltages.

The second module of the GIO Node is called the "GIO Node – Extender", referred to henceforth as the *Extender board*. This circuit board is mounted on top of the base board. The extender board is designed to be used when a GIO Node requires more inputs and outputs than the base board provides. The extender board provides additional I/Os without needing to change either the housing or the base board's circuit. As high side drivers are among the most expensive electronic components on the circuit board, hence it is unwise to mount more high side drivers on the base board of the GIO node unless they are likely to be used. The extender board has a network controller which enables it to communicate with the base board, thus utilizing a point-to-point network connection to reduce the number of pins needed to interconnect the two circuit boards. There is an additional set of network transceivers on the extender board, enabling both the extender board and base board to connect via separate external connectors to potentially different switches in the domain. Additionally, having two network connectors enables the GIO node to be connected to different wiring harnesses - facilitating network redundancy. Having two transceivers enables the GIO node to utilize either vehicle network and enables tolerance to damage of one of the wiring harnesses.

When many more I/Os are required the two board setup can be used and when fewer I/Os are required, then a extension board that is not fully populated can be deployed while still maintaining redundant connectivity.

The analysis of number of high side & low side drivers and digital & analog inputs is detailed in the following sections.

## 4.2 Chassis ECUs and their configuration

In the present generation of commercial trucks, two or three chassis ECUs are used for controlling and communicating with various sensors and actuators on the vehicle, such as lights, DC motors, solenoid valves, electronic suspension, and various sensors in the chassis domain.

On a heavy duty truck, the chassis domain ECUs are mounted at three locations on the truck chassis. These ECUs control the sensors and actuators which are in the vicinity of their mounting location. The three ECUs are named according to their position on the chassis: front chassis module, central chassis module, and rear chassis module.

* In a high-side switch/driver, the load is between ground and the switching device. † In a low-side switch/driver, the load is between the power rail and the switching device.

In the current generation of ECUs, there are different configurations of modules used for different variants of medium duty and heavy duty trucks. However, these configurations, utilize the same circuit and casing design. By re-utilizing the same circuit design for all variants, the circuit is designed with a focus on the highest capability configuration variant. Given on this circuit, some of the components are not mounted to produce lower capability configuration variants. However, this results in different part numbers for all the variants. The following analysis will take into account the need to replace all the variants of ECUs with GIO nodes. The functions controlled by these modules are discussed in the following subsections.

### 4.2.1 Front Chassis Module

The front chassis module is responsible for lights in the front region of the truck, for example, the position lights, fog lights, daytime running lights, direction indicators, main beam light, extra main beam light, and the low beam light. It is also responsible for controlling the windscreen washer pump, air horn, city horn, and the headlight levelling motors. This ECU is responsible for collecting sensor data from the washer fluid sensor, hazard signal, and ambient temperature sensor.

Details of number of I/O pins available on different the variants of the front chassis module are given in Table 4.1

Table 4.1: I/O summary of front chassis module variants

| Configuration | Digital Output (High Side) | Digital Output (Low Side) | Digital Input | Analog Input |
|---|---|---|---|---|
| High | 25 | 0 | 3 | 2 |
| Low | 20 | 0 | 3 | 2 |

### 4.2.2 Central Chassis Module

The central chassis ECU is mounted near the fifth wheel of the truck *. This ECU is responsible for controlling various valves of the electronically controlled suspension (ECS), the alternator, chassis lubrication motor, and power takeoff valves. Due to its vicinity to the fifth wheel, it controls the lights on the trailer, for example, the trailer reverse light, fog light, and warning light. This ECU collects a lot of sensor data due to its mounting position; for example, data from various ECS sensors, fuel level sensors, multiple fifth wheel sensors, gearbox oil sensors, and neutral & reverse feedback sensors.

* The fifth wheel is a “U” shaped coupling component found of the rear portion of the large vehicle, it allows the driver to connect a cargo attachment to the back of vehicle, for example, a trailer.

Details of number of I/O pins available on different the variants of the central chassis module are given in Table 4.2

Table 4.2: I/O summary of central chassis module variants

| Configuration | Digital Output (High Side) | Digital Output (Low Side) | Digital Input | Analog Input |
|---|---|---|---|---|
| High | 20 | 3 | 14 | 8 |
| Medium | 14 | 3 | 7 | 6 |
| Low | 9 | 0 | 7 | 3 |

### 4.2.3 Rear Chassis Module

This is the most complex chassis module with a lot of sensors and actuators connected to it. This ECU is responsible for controlling various lights in the rear half of the truck; for example, the fog lights, position lights, reverse lights, direction indicator lights, brake lights, and warning lights. This ECU is also responsible for controlling various solenoid valves for the rear ECS. This ECU collects the pressure sensor data from the rear ECS sensors and the brake pressure sensors. It also collects data from the dog clutch feedback sensors and axle switch from the rear wheel.

Details of number of I/O pins available on different the variants of the rear chassis module are given in Table 4.3

Table 4.3: I/O summary of rear chassis module variants

| Configuration | Digital Output (High Side) | Digital Output (Low Side) | Digital Input | Analog Input |
|---|---|---|---|---|
| High | 38 | 2 | 14 | 12 |
| Medium | 31 | 2 | 5 | 4 |
| Low | 28 | 0 | 6 | 2 |

## 4.3 Design of GIO Nodes

This thesis aimed to replace the existing chassis ECUs without redistributing any functionalities among them. This meant that, the GIO had to be designed with the highest configuration ECU in focus.

### 4.3.1 Configuration of GIO Node I/Os

The rear chassis ECU has almost double the number of outputs of the front chassis ECU. Therefore, when a GIO node is used as a replacement for the front chassis module, half of its hardware would not be utilized.

To deal with such scenarios and to make the GIO nodes applicable in a much wider perspective, it was decided to design the GIO node as a two board design. This design utilizes the concept of modularity, in which functionality is split into two modules as previously discussed in Section 4.1.2

The final number of I/Os and the distribution of inputs and outputs among the two modules of GIO node is detailed in Table 4.4

Table 4.4: I/O distribution between GIO node modules

| I/O Type | GIO Node | Base Board | Extender Board |
|---|---|---|---|
| Digital Output (High Side) | 38 | 20 | 18 |
| Digital Output (Low Side) | 4 | 2 | 2 |
| Digital Input | 12 | 6 | 6 |
| Analog Input | 14 | 7 | 7 |

This distribution of inputs and outputs between the base board and the extender board was done with an aim that the standalone base board can replace all variants of central chassis module and the lower variant of front chassis module. While the GIO node with both base and extender board can replace both the variants of rear chassis module and the higher capability variant of front chassis module.

### 4.3.2 Major design decisions

In order to implement the decided configuration of the GIO node as per Section 4.3.1, few design decisions were taken to make the prototype complaint with the geometrical, electrical, and design constraints of automotive electric systems. These decisions are detailed in this section.

#### 4.3.2.1 Two board design

The two board multi-module design is inspired by the concept of different shields used in modern day development boards, such as Raspberry Pi, Arduino, Beaglebone, etc. The current GIO node concept utilizes just one extender board. This is done to save resources when using the GIO node in an application with lower requirement of I/Os.

However, in future, this concept can be extended with multiple variants of extender boards with different capabilities to utilize the same base board to replace further ECUs in other truck domains.

This reduces the design costs by reusing the base board and housing.

#### 4.3.2.2 Choice of microcontroller

Since the concept of integrated architecture is focused on moving all the computation power to the central computation node, the GIO nodes need lower computing abilities compared to their predecessors. However, the microcontroller being used must be able to perform some of the closed loop functions which need to be done with low delay, so that the ECU functions correctly in case of momentary loss of communication with the computational node.

#### 4.3.2.3 Choice of High side drivers

The high side drivers are needed to drive the high side digital outputs. In a GIO node, these are the major cost and design driving element.

In the application of high side driver in the chassis node digital outputs, the highest current requirement is 12 Ampere (A). The current design proposes to use dual output high side drivers, which can supply a maximum of 18A when used in dual mode. However, they can be used as single outputs as well. These high side drivers communicate with the microcontroller over Serial Peripheral Interface (SPI) bus.

The GIO node uses three variants of high side drivers with dual outputs of 3A, 6A, and 9A. These are distributed among the two boards depending upon the functional requirements. The distribution of these drivers among the two boards is shown in Table 4.5

Table 4.5: Distribution of different high side drivers among GIO Node modules

| **Output Current** | **Number of Outputs** | |
|---|---|---|
| | **Base Board** | **Extender Board** |
| 3A | 12 | 8 |
| 6A | 6 | 6 |
| 9A | 2 | 4 |

According to this distribution, the maximum current requirement on base board and extender board is 90A and 96A respectively.

#### 4.3.2.4 Configurable Inputs

The inputs on the GIO node are designed to be configurable in three different configurations: pull-up, pull-down, and disconnected. This means that the inputs, both digital and analog, can be either configured as pull-up inputs or pull-down inputs. This configuration can be done during the initial configuration or on the go if needed. This makes the GIO node highly flexible in terms of collecting data from various kinds sensors and other sources.

In order to reduce passive energy leakage, the pull-up and pull-down circuits can be disconnected if a particular input is not used in a certain application.

As an additional feature, these inputs can also act as low power digital outputs when used in the pull-up configuration. This can be utilized for the purpose of powering low power sensors.

#### 4.3.2.5 Communication

For communication with other nodes (or ECUs), this version of GIO node uses CAN-FD and LIN protocol. This allows the GIO node to communicate with the computation node(s) and to communicate with other nodes in the same domain. The GIO is also capable of controlling various devices communicating over the LIN protocol.

For communication between the two modules, i.e. the base board and extender board, multiple SPI buses on the microcontroller are utilized, the analog signals are communicated over Inter-Integrated Circuit (I2C) bus.

#### 4.3.2.6 Dual power supply

Commercial trucks utilize redundant 24 Volts (V) power supplies with two wiring harnesses running throughout the span of truck, which ensure a fail-safe operation. The GIO node aims to utilize both of them, especially for fail-safe operation of high-side drivers. Half of the high side drivers on each board are supplied by one supply and other half by the other power supply. The GIO node has three separate grounds, an analog ground for analog inputs, a power ground for all the 24V circuitry, and a signal ground for rest of the 5V circuitry. These all grounds meet at a common star point in order to acheive noise isolation.

The mapping of functions on the truck to the high side drivers is also done so that there is never a complete loss of any functionality. For example, in case of headlight lamps, the high beam lamps are mapped to one supply and low beam lamps are mapped to the other power supply. This ensures that at least one of the head lamps works in case of failure of one of the supply lines. In many other cases, mapping is done in a way that the left-side actuators are powered from one power supply and the right side actuators are powered by the other power supply to achieve similar safety features.

#### 4.3.2.7 Choice of connectors

For external connections, the GIO node design uses different connectors for signal and power due to the high power demand. Moreover, the mechanical design of the GIO node housing * allows the GIO node to either have straight or angled connectors to allow

* The mechanical design was carried out as another thesis project in conjunction with this thesis project by a product design master's student [38]

different mounting positions. This decision does not affect the manufacturing process of the housing and the type of connects can be selected at the time of assembling the GIO node. The connector used for this purpose was a customized version of TE Connectivity 154 Pin and Tab header [39], resulting in a 58 pin connector with 6 power and 52 signal pins.

A separate power connector pin housed in TE Connectivity DTHD Plug [40] allows up to 100A current. The two boards use one power and one signal connector each.

For connecting the two boards together, redundant inter-board connectors are used. For utilizing the dual power supply, the power travels between the boards using inter board power connectors. Additionally, there are various inter-board connections for communicating the I/Os from the external connector on the extender board to the microcontroller over SPI and I2C. The connectors used for these inter-board connections are custom headers from Molex*.

All the I/Os and communication interfaces like CAN, LIN, analog and digital inputs, high-side and low-side outputs, 5V power for some sensors, and identification pins are mapped with the 58 Pin connector. Power could be supplied through the power pins of this connector (up to 20A) or through the separate power connector (up to 100A).

### 4.3.3 Technical specifications of GIO Node

The final specifications of the GIO Node based on Section 4.3.2 are detailed in this section. These correspond to the configuration of the final prototype of the base board.

For the computation on GIO node, 32-bit CPU from NXP, S32K144 [41] was used on the GIO Node, It has an ARM$^{TM}$ Cortex-M4 based on ARMv7 architecture. Among the highlights of memory and memory interfaces, it houses 512 KB of program flash memory, 64 KB of FlexNVM, and 16 KB Boot ROM. The microcontroller has two 12-bit analog-to-digital converters with up to 16 channel analog inputs per module. It supports debugging via Serial Wire JTAG Debug Port. It also supports a CAN-FD module.It runs at a clock frequency of 40MHz using an external oscillator.

For power management, a system basis chip, NXP FS4500 [42] was used. This is a highly flexible switch mode power supply, with linear voltage regulator. It has configurable fail silent safety behavior and features. It houses built in CAN-FD and LIN interface for enabling external communication on the GIO node.

For controlling the outputs, Dual 24 V High side switches, NXP 20XS4200 [43], NXP 10X4200 [44], and NXP 06X4200 [45] (formerly Freescale Semiconductors) were respectively used for driving 3A, 6A, and 9A outputs. These are controlled over SPI, and have diagnostic features, with over-current, short-circuit, and over-temperature protection. These high-side drivers have the ability to control various load types

* These connectors were ordered according to the distance between the two boards using the Molex custom header configurator: http://www.molex.com/molex/family/customheader

including bulbs, solenoids, and DC motors. The current design utilizes a mix of these drivers. Out of the 38 outputs, 6 are designed for the high power drivers (9A dual supply drivers), while 12 of them are designed for moderate power drivers (6A dual supply drivers), and 20 of them are designed for low power drivers (3A dual supply drivers). There are 4 low side drivers, which are based on metal–oxide–semiconductor field-effect transistors (MOSFET).

The GIO Node houses software configurable pull-up and pull-down configurations for the analog and digital inputs. These are configuration of these pull-ups and pull-downs can be done when starting up the node.

### 4.3.4 Comparison of GIO node design with the design requirements

**Generic Operation** The GIO Nodes are not expected to be used for one specific application, but can be configured to be used in multiple use cases (in this case as any of the three chassis ECUs). This unique feature not only helps in bringing down the costs, but also helps in an incremental hardware development. It adds the flexibility for the OEMs. They have an option of adding features during the later part of development process, which was very difficult earlier.

**Computation capabilities** The GIO node needed a less powerful processor compared to the chassis ECUs, as it is aimed to be utilized in an integrated E/E architecture. The current CPU fulfills this requirement.

**Power Capabilities** The maximum current requirement on base board (90A) and extender board (96A) is lower than the current capacity of the power connector (100A). So, the power requirements are fulfilled.

**Number of unique part numbers** The chassis I/O ECUs have 3 different versions, which have a total of 8 variants. Which results in 3 unique PCB designs and 8 different part numbers. With GIO nodes, only 2 unique PCB designs are needed and 2 unique part numbers. This can result in a lot of cost savings. Moreover there is only one housing design needed in GIO node compared to three unique designs used by chassis ECUs.

### 4.3.5 GIO Node prototype

Figure 4.2 shows the manufactured prototype of base board. Figure 4.3 shows the base board connected with the wiring harness to a test rig.

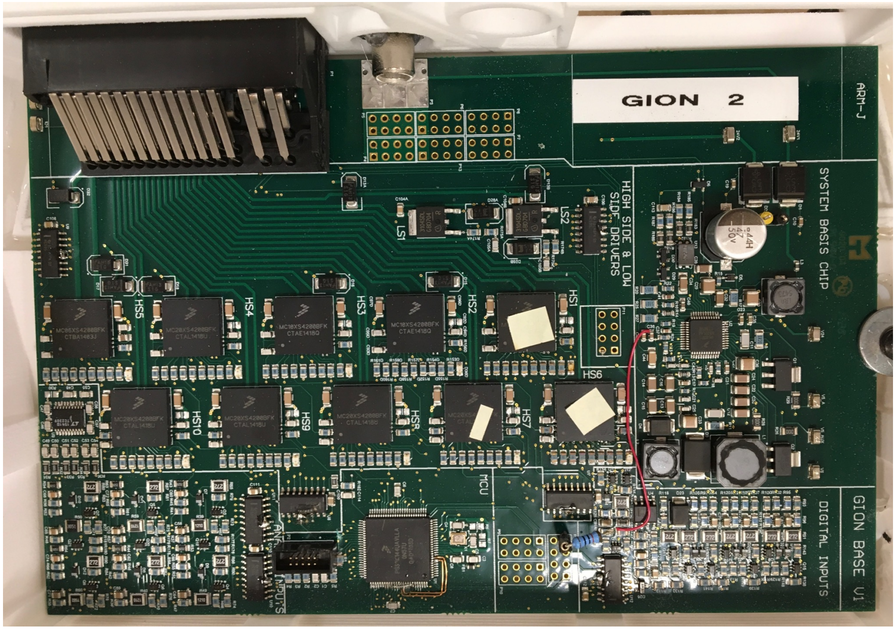


Figure 4.2: Generic I/O node prototype

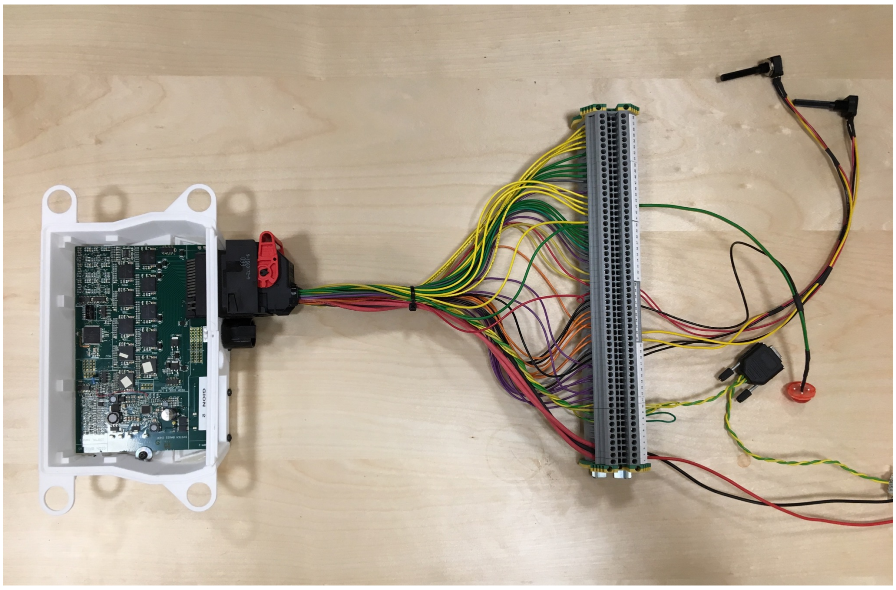

Figure 4.3: Generic I/O node prototype connected with the test rig

## 4.4 Next generation variant of GIO Node

The GIO node configuration detailed and designed in the previous sections of this chapter is a minimal configuration targeting the chassis domain of a commercial truck, it may be further expanded to include additional domains of the current generation of trucks. However, in a more advanced architecture, such as the one discussed in Section 3.2, a generic IO node would be ubiquitous in the vehicle hence it needs to have further advanced features that can enhance the capabilities and application of GIO node.

These advanced features would include using Automotive Ethernet the primary form of communication. A wireless radio could be used as a secondary mode of communication with the with the wireless network gateway at the computational node. There can be multiple variants of the extender boards depending upon the needs of the particular function. This will have multiple advantages, for example, when using a different extender board in order to cater to the requirements of a particular functionality, there would be no need to design a new casing, or a new base board and hence no need for a new software. This would mean, that a slight modification in the software configuration of the GIO node would be enough to configure a new extender board on the GIO node.

# Chapter 5

# Conclusions

This chapter explains the conclusions obtained throughout the design, development, and evaluation described in this thesis and proposes a number of improvements, extensions, or complements that may be of interest in order to continue this work.

## 5.1 Conclusion

With a swiftly changing automobile industry, automation has taken the center stage in recent years. With software being the factor in speeding and enabling the introduction of automation, the hardware also needs to evolve to adapt to these changes. The hardware needs to be ready for the future phases of vehicle automation and be more cost effective even as software continues to become more expensive. This thesis was aimed at a hardware upgrade by proposing a new E/E architecture for commercial trucks, i.e. the Integrated E/E architecture, and designing the potential replacements for the ECUs, i.e., the GIO nodes.

The proposed E/E architecture was discussed in Chapter 3 with details of different nodes in Section 3.1. The major focus of this thesis was to design a GIO node, the architecture and concept behind the GIO node was discussed in Section 4.1, while a tentative sketch of a future E/E architecture was discussed in Section 3.2.

A majority of time was devoted to the design of GIO node, the parameters and configuration of the design were discussed in Sections 4.2 and 4.3. The thesis work aimed at having a prototype by the end of thesis period, and a prototype of the GIO node base board was manufactured and is shown in Section 4.3.5.; it is currently being tested for various software capabilities. Initially, the thesis also aimed at designing a prototype of the extender board for the GIO node; however, only the base board could be designed and manufactured according to the configuration presented in Section 4.3. Expected features for a more capable variant version of the GIO node are presented in Section 4.4.

This thesis has introduced a highly cost effective way of using generic hardware in commercial trucks, which, if pursued with further ECUs can become a revolutionary way of manufacturing electrical hardware for commercial trucks and can be extended into other heavy duty vehicles as well. The biggest advantage apart from cost savings and lower design costs would be that companies would maintain fewer modules (hence less part numbers) and the service points would have a greater possibility of having a replacement for a faulty ECU, thus allowing service points to stock a smaller inventory. This in turn reduces the time during which a vehicle is out of service while waiting for the replacement of a faulty ECU.

## 5.2 Future work

There are several possibilities for further work in this topic. From a networking point of view, a better automotive network using a single protocol is a topic of interest. From the perspective of the GIO nodes, trying to make the nodes more generic while replacing more and more ECUs would be desirable. The multi-module design can be exploited to use different types of extender boards in order to provide different capabilities for different applications. From the software perspective, it is also possible to have some deep learning algorithms running on the GIO node which observe the sensor decay over its lifetime and predict a possibility of sensor malfunction.

## 5.3 Required Reflections

This thesis project is important in many ways for the heavy duty commercial trucks and automotive industry as well as for the environment. Upgrading to the proposed E/E architecture will reduce the weight of the wiring harnesses of a truck, leading to the reduction of the emissions. Moreover, this architecture is the initial step for commercial trucks to move in the direction of vehicle automation. This vehicle automation will have multi fold benefits for the environment, due to a large reduction in fuel consumption resulting from efficient driving, platooning, and electromobility.

The GIO node is a new concept for the automotive industry which has heretofore used highly specialized ECUs designed by the Tier 1 suppliers for specific functions. This has led to little reuse of designs and software. Moreover, since these ECUs were tailored to specific need, multiple ECUs have been designed for the same function on different variants of the vehicles. The overall process lead to high number of modules and variants (and hence numerous part numbers), and a need to maintain them and have replacements ready regardless of the volume of them being purchased by customers. Automotive companies spend large amounts of money per unique module to keep replacement parts in stock, so that they can be ready in case of failures of ECUs. This large inventory of parts has its own environmental and financial impact. Replacing many different types of ECUs with GIO nodes looks like a promising concept to tackle this.

# Bibliography

[1] O. C. Duffy and G. Wright, *Fundamentals of medium/heavy duty commercial vehicle systems*. Jones & Bartlett Publishers. ISBN 1284041166

[2] C. J. Murray, "Tesla adds all-wheel drive, autopilot: CEO Elon Musk seeks to match the performance of the mclaren F1 supercar." *Design News*, vol. 69, no. 12, December 2014.

[3] S. Thrun, "What we're driving at," Oct., (Accessed on June 09, 2017). [Online]. Available: https://googleblog.blogspot.se/2010/10/what-were-driving-at.html

[4] J. F. Kurose and K. W. Ross, *Computer networking : A Top-Down Approach*, seventh edition ed. ISBN 9780133594140. [Online]. Available: https://www.pearson.com/us/higher-education/program/Kurose-Computer-Networking-A-Top-Down-Approach-7th-Edition/PGM1101673.html

[5] J. Postel, "Transmission Control Protocol, Internet Request for Comments, vol. RFC 793 (INTERNET STANDARD), Sep. 1981," (Accessed on June 14, 2017). [Online]. Available: http://www.rfc-editor.org/rfc/rfc793.txt

[6] ——, "User Datagram Protocol, Internet Request for Comments, vol. RFC 768 (INTERNET STANDARD), Aug. 1980," (Accessed on June 14, 2017). [Online]. Available: http://www.rfc-editor.org/rfc/rfc768.txt

[7] "BOSCH CAN Specification," *Robert Bosch GmbH*, 1991, (Accessed on June 06, 2017). [Online]. Available: http://www.bosch-semiconductors.de/media/ubk_semiconductors/pdf_1/canliteratur/can2spec.pdf

[8] S. Corrigan, "Controller Area Network Physical Layer Requirements," Application Report, SLLA270, Texas Instruments, Dallas, Texas, Jan 2008, (Accessed on June 09, 2017). [Online]. Available: http://www.ti.com/lit/an/slla270/slla270.pdf

[9] HMS Technology Center Ravensburg GmbH, "CANopen Basics," (Accessed on June 09, 2017). [Online]. Available: http://www.canopensolutions.com/english/about_canopen/about_canopen.shtml

[10] L.-B. Fredriksson, "CAN Kingdom Rev. 3.01 - Specification document," *Kavser AB*, 1995. [Online]. Available: https://www.kvaser.com/wp-content/uploads/2014/02/ck301p.pdf

[11] ODVA, Inc, "DeviceNet™ – CIP on CAN Technology," Technology Overview Series: DeviceNet, PUB00026R4, ODVA, Inc,. Ann Arbor, Michigan, USA, Mar 2016, (Accessed on June 09, 2017). [Online]. Available: https://www.odva.org/Portals/0/Library/Publications_Numbered/PUB00026R4_Tech-Adv-Series-DeviceNet.pdf

[12] Truck Bus Control and Communications Network Committee, "Serial Control and Communications Heavy Duty Vehicle Network - Top Level Document - J1939 Ground Vehicle Standard," SAE, Standard J1939_201104, 14 Apr 2011, (Accessed on June 09, 2017). [Online]. Available: https://saemobilus.sae.org/content/j1939_201104

[13] National Instruments, "Introduction to the Local Interconnect Network (LIN) Bus," White Paper, National Instruments, 24 Aug 2016, (Accessed on June 06, 2017). [Online]. Available: http://www.ni.com/white-paper/9733/en/

[14] "The ethernet: A local area network: Data link layer and physical layer specifications," *SIGCOMM Comput. Commun. Rev.*, vol. 11, no. 3, pp. 20–66, Jul. 1981. doi: 10.1145/1015591.1015594 (Accessed on June 06, 2017). [Online]. Available: http://doi.acm.org/10.1145/1015591.1015594

[15] M. Ibrahim, "Ethernet in steer-by-wire applications," *Master's thesis, KTH Royal Institute of Technology, Stockholm, Sweden*, TRITA-ICT-EX-2011:175, July 2011. [Online]. Available: http://kth.diva-portal.org/smash/record.jsf?pid=diva2%3A507875&dswid=2414

[16] IEEE 802.3 Ethernet Working Group, "IEEE 802.3 ETHERNET," 2012, (Accessed on June 11, 2017). [Online]. Available: http://www.ieee802.org/3/

[17] T. J. Glenn Parsons, John Messenger, "IEEE 802.1: IEEE Standard for Local and metropolitan area networks," 2009, (Accessed on June 11, 2017). [Online]. Available: http://www.ieee802.org/1/

[18] M. G. David E. Carlson, Om Agrawal, "IEEE-SA -IEEE Get 802 Program - 802.2: Logical Link Control," 1998, (Accessed on June 11, 2017). [Online]. Available: http://standards.ieee.org/about/get/802/802.2.html

[19] K. McCloghrie and R. Fox, "IEEE 802.4 Token Bus MIB," Internet Request for Comments, vol. RFC 1230 (Historic), May 1991, (Accessed on June 11, 2017). [Online]. Available: https://tools.ietf.org/html/rfc1230

[20] Cisco Systems Inc., "Token Ring/IEEE 802.5)," Cisco Press, 8 Dec 1999, (Accessed on June 11, 2017). [Online]. Available: https://www.cisco.com/cpress/cc/td/cpress/fund/ith2nd/it2409.htm

[21] G. Held, "Fundamentals of digital signaling, coding and clocking," in *High Speed Digital Transmission Networking*. Chichester, UK: John Wiley Sons, Ltd, April 2001, pp. 27–46. ISBN 9780471983583

[22] "IEEE 802.3-2008: Standard for Information technology-Specific requirements - Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications," (Accessed on June 11, 2017).

[23] Ixia Network, "Automotive Ethernet: An Overview," May 2014, (Accessed on June 06, 2017). [Online]. Available: www.ixiacom.com

[24] B. Kraemer, "Automotive ethernet," *IEEE Communications Magazine*, vol. 54, no. 12, pp. 4–4, December 2016. doi: 10.1109/M-COM.2016.7785879

[25] Broadcom Corporation, "BroadR-Reach Physical layers specifications for automotive applications," 2014, (Accessed on June 11, 2017). [Online]. Available: http://www.ieee802.org/3/1TPCESG/public/BroadR_Reach_Automotive_Spec_V3.0.pdf

[26] A. E. Tan, "Designing 1000BASE-T1 Into Automotive Architectures." Marvell, Oct 2014, (Accessed on June 06, 2017). [Online]. Available: https://standards.ieee.org/events/automotive/2014/02_Designing_1000BASE-T1_Into_Automotive_Architectures.pdf

[27] W. Walter Rentzsch, "Automated Trucks – The next big disruptor in the automotive industry?" April 2016, (Accessed on 2017-07-11). [Online]. Available: https://www.rolandberger.com/en/Publications/pub_automated_trucks_the_next_big_disrupter_in_the_automotive_industry.html

[28] D. Cottingham, "What is a vehicle platoon?" March 2014, (Accessed on 2017-11-07). [Online]. Available: https://www.driverknowledgetests.com/resources/what-is-a-vehicle-platoon/

[29] O. Olarte, "Human error accounts for 90% of road accidents," AlertDriving, Toronto, ON, Canada, April 2011, (Accessed on 2017-07-11). [Online]. Available: http://channel.staging.alertdriving.com/home/fleet-alert-magazine/international/human-error-accounts-90-road-accidents

[30] SAE On-road automated driving committee, "J3016A: Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles - SAE International," 30 September 2016, (Accessed on 2017-07-11). [Online]. Available: http://standards.sae.org/j3016{_}201609/

[31] P. Hansen, "BMW and Audi Want to Separate Vehicle Hardware from Software," Electronic Design, 28 April 2017, (Accessed on 2017-05-15). [Online]. Available: http://www.electronicdesign.com/automotive/bmw-and-audi-want-separate-vehicle-hardware-software

[32] C. Hammerschmidt, "Daimler standardizes electronics platform for trucks," eeNews, June 2016, (Accessed on June 14, 2017). [Online]. Available: http://www.eenewsautomotive.com/news/daimler-standardizes-electronics-platform-trucks

[33] J. Hemlin and D. Larsson, "Exploring the generic timer module's feasibility for truck powertrain control," *Master's thesis, Chalmers University of Technology, Gothenburg, Sweden*, 2014. [Online]. Available: http://publications.lib.chalmers.se/records/fulltext/219127/219127.pdf

[34] H. Stahle, L. Mercep, A. Knoll, and G. Spiegelberg, "Towards the deployment of a centralized ict architecture in the automotive domain," in 2013 2nd Mediterranean Conference on Embedded Computing (MECO), 2013, pp. 66–69. DOI: 10.1109/MECO.2013.6601320.

[35] R. Ekman, "Automobile control systems transition from controller area networks to ethernets," *Bachelors's thesis, KTH Royal Institute of Technology, Stockholm, Sweden*, ICT-EX-2014:108, July 2014. [Online]. Available: http://kth.diva-portal.org/smash/get/diva2:734675/FULLTEXT01.pdf

[36] A. Talic, "Security analysis of ethernet in cars," *Master's thesis, KTH Royal Institute of Technology, Stockholm, Sweden*, ICT-EX-2017:180, October 2017. [Online]. Available: http://kth.diva-portal.org/smash/get/diva2:1147703/FULLTEXT01.pdf

[37] P. Thorngren, "'The Epic Split´ of Software and Platform - The Integration Challenge," May 2014, (Accessed on 2017-06-28). [Online]. Available: http://www.electronicdesign.com/automotive/bmw-and-audi-want-separate-vehicle-hardware-software

[38] J. Ljunggren, "Development of a multipurpose input/output module - mechanical concept/design generation of ecu enclosure," *Master's thesis, Chalmers University of Technology, Gothenburg, Sweden*, 2017.

[39] TE Connectivity, "TE Connectivity 284617-1, 154 POS PIN TAB HEADER ASSY," 23 August 2016, (Accessed on 2017-06-28). [Online]. Available: http://www.te.com/usa-en/product-284617-1.html

[40] ——, "TE Connectivity DTHD06-1-4S, DTHD PLUG ASM," 23 August 2016, (Accessed on 2017-06-28). [Online]. Available: http://www.te.com/usa-en/product-DTHD06-1-4S.html

[41] NXP Semiconductors, "S32K: 32-bit Automotive General Purpose Microcontrollers," 30 September 2016, (Accessed on 2017-11-11). [Online]. Available: https://www.nxp.com/products/processors-and-microcontrollers/arm-based-processors-and-mcus/s32-automotive-platform/32-bit-automotive-general-purpose-microcontrollers:S32K

[42] ——, "FS4500: Fail Silent System Basis Chip," 30 September 2016, (Accessed on 2017-07-11). [Online]. Available: https://www.nxp.com/products/power-management/system-basis-chip-sbc/functional-safety-sbcs/fail-silent-system-basis-chip-with-ldo-up-to-500ma-on-vcore:FS4500

[43] ——, "Dual 24 V, 20 mOhm High Side Switch," 30 September 2016, (Accessed on 2017-07-11). [Online]. Available: http://cache.freescale.com/files/analog/doc/data_sheet/MC20XS4200.pdf

[44] ——, "Dual 24 V, 10 mOhm High Side Switch," 30 September 2016, (Accessed on 2017-07-11). [Online]. Available: http://cache.freescale.com/files/analog/doc/data_sheet/MC10XS4200.pdf

[45] ——, "Dual 24 V, 6 mOhm High Side Switch," 30 September 2016, (Accessed on 2017-07-11). [Online]. Available: http://cache.freescale.com/files/analog/doc/data_sheet/MC06XS4200.pdf

TRITA-ICT-EX-2017:199