Alexandru N. Onea

Alexandru.onea@toporcomputing.com

Abstract

[Draw your reader in with an engaging abstract. It is typically a short summary of the document.   
When you’re ready to add your content, just click here and start typing.]

Modern Vehicle Architectures

Advanced Embedded Systems

# Introduction

Since the first modern automobiles produced by Karl Benz and Henry Ford, the automotive industry underwent continuous growth and development, becoming one of the largest industries worldwide. One of the most important drivers of the automotive industry is the constantly increasing demand for functionality, comfort, safety and performance inside the vehicle, matched to the rapid development of electronics and software technologies [1]. During the last decades more and more functions of the vehicle moved away from mechanical and hydraulically technologies and adopted electronic counterparts. Examples of such functions are the steering, transmission and wipers. Moreover, new functions were added to the vehicle once certain embedded technologies became available, such as the electronic break system and the electronic stability control. In the same time, governmental regulations reading transportation safety and pollution led to the advancement of specific technologies aimed towards low power consumption and environmentally friendly vehicles. All these factors contributed to the development of embedded systems and software technologies for the automotive industry.

## Early stages

Early iterations of vehicle architectures comprised a relatively small number of electronic control units (ECUs). Each of the ECUs implemented a function of the vehicle completely. Although limited, the communication between ECUs was implemented using point-to-point connections [1]. In later years, the number of ECUs steadily increased as the technology became available, thus the number of connections grew proportionally. The development of hardware components made possible the integration of new functions and even domains into the vehicle, such as radio equipment for the purpose of in-vehicle entertainment. Adding new features and functions had a direct impact on the hardware requirements of the embedded systems regarding processing speed and memory size. The software size within the vehicles rapidly increased from around 1KB to almost over 2MB in just 20 years [1]. Other features required completely new hardware to be integrated, such an example being the introduction of GPS in vehicles for safety and security, and much later for navigation purposes. Even though such hardware was available, it soon became obvious that it was not suitable for the road conditions in which the embedded systems within the vehicle should operate. Soon, silicon providers started producing automotive graded integrated circuits (ICs) which met the requirements of temperature, vibration, shock, electromagnetic compatibility and electrostatic discharge of the automotive industry.

## Current development

Vehicle became complex distributed embedded systems consisting in a large numbers of sensors, actuators and processing units. Each functions of the vehicle moved from being stand-alone implemented in a single ECU to being distributed across several electronic control units (ECUs). The number of ECUs in a vehicle was also subject to growth, averaging in current vehicle architectures to around sixty [1][2]. Moreover, the data exchange demand increased as well since more and more functions required inputs from different parts of the vehicle. For example, modern vehicles employ dynamic volume control of the media player with respect to the vehicle speed. The volume control and the vehicle speed reading are implemented in different ECUs, thus data exchange is required. The previous approach of point-to-point connections between ECUs proved inefficient and the need for large bandwidth communication busses emerged. The advantages that communication busses have over the traditional interconnections are numerous, including reduced weight of cables and, implicitly, reduced cost. In fact the wiring and the harnesses required to connect various ECUs proved to be the most expensive component before the year 2000, wiring sometimes peaking at a few kilometres in length [3].

The early vehicle communication networks were not standardized and were based on proprietary circuitry and mostly Universal Asynchronous Receiver Transmitter (UART) interfaces. With the increasing vehicle functionality and competition on the market, the automakers started employing external suppliers to develop more complex and standardized communication interfaces [4]. The standardization of protocols and in-vehicle communication allowed for more complex systems to be integrated and led to the so-called open architecture of the modern vehicles. Bosch developed one of the first vehicle networks – the CAN (Controller Area Network) around the year 1980, which was first integrated into production vehicles by Mercedes-Benz in 1992.

## Future trends

Latest trends in the automotive industry support even more demanding requirements such as vehicle communication with the external world. Most notable examples are internet connectivity, vehicle-to-infrastructure and vehicle-to-vehicle capabilities. Such requirements add constraints to the vehicle architecture and communication systems and often require a change in paradigm in order to be feasible.

# Vehicle functional domains

Vehicle functions and their implicit hardware, software and mechanical components are usually split into domains by Original Equipment Manufacturers (OEMs). Each domain has its own requirements in terms of safety and performance and communication. The four domains that are commonly referred by OEMs and suppliers in the industry are *powertrain*, *chassis*, *body* and *infotainment* [1]. Each domain comprises several ECUs which co-operate to achieve its specific requirements. Due to the common communication requirements shared by all the ECUs belonging to a certain domain, a single communication network is sufficient to implement the intra-domain communication. This grouping of ECUs with shared communication requirements into domains increases the robustness of the overall system by separating the communication as well into critical and non-critical communication.

Although each domain is responsible for driving certain functions of the vehicle, it is almost always the case that inter-domain communication is required as well.

# Architectures

## Gateway architecture

## Domain controlled architecture

# Communication systems

In-vehicle data communication has been a key component to the modern vehicles. Whether it is inter-ECU or intra-ECU communication, the parameters of data transfer play a critical role in the performance of each individual function of the vehicle and the overall system. The communication technologies integrated into modern vehicles are standardized and described in detail at electrical, physical, software and mechanical level. The characteristics of the physical layer impact the throughput and the maximum data rates. The mechanical aspects of the connection contribute to the cost and weight reduction of the vehicle while also affecting the behaviour of the connection in the presence of shock, wide temperature ranges, vibrations and other environmental constraints. The electrical aspects influence both the performance of the data transfer and the power consumption while the software design of the connection transmission and reception sides impact the resources consumption for the processing units (CPU load, memory, etc.).

# Domain controlled architectures of the future

This chapter describes the state-of-the-art next generation in-vehicle networking standards proposed and described in [5] and presents the potential use of automotive Ethernet as backbone for the domain controlled architecture as described in [6].

## Limitations of current technologies

The increase in vehicle functions both in the infotainment and the driver assistance systems domains exploits to the limit the currently available in-vehicle communication technologies in terms of bandwidth and performance. Following the development of advanced driver assistance systems (ADAS) and other technologies aimed towards autonomous driving, it became obvious that as the driver interaction with the vehicle will be reduced, the potential and opportunities for in-vehicle entertainment and even workspace functions will rapidly grow. Thus, the vehicle will need to support many more functions such as high speed internet connectivity, high quality media streaming and playback, workspace applications and fully featured connectivity with the external environment. These features require more bandwidth than it is currently available using traditional communication and in-vehicle networking technologies.

In the traditional gateway vehicle architecture, the communication between different domains in routed through special ECU called the *gateway*. One of the purposes of the gateway ECU is to make sure that there is information flow across domains and across their different networks. In addition, the gateway performs conversions of messages between different protocols (e.g. CAN to LIN, MOST to CAN, etc.) when necessary. Furthermore, the gateway is responsible for delivering diagnostic requests and responses between domains and the On-Board Diagnostic (OBD) clients.

Historically, the advantages of such a gateway architecture are both in terms of lifetime and in terms of flexibility, enabling easy replacement of damaged parts as well as enhanced robustness for adjusting the system throughout the production stages [5]. However, the advantages described before add overhead to the communication between different parts of the vehicle and lead to overloading and interferences.

In an attempt to solve the overloading problem and, possibly, to address the timing constraints imposed by the processing of various critical sensor information such as video camera inputs, a possible solution is to cluster vehicle functions into single ECUs [5]. With this approach, and also depending on the various functions of the vehicle, the signal acquisition from sensors, the processing and the decision are performed within the same local system, thus reducing the amount of time required to perform actions based on sensor input. Moreover, this approach reduces the number of ECUs in the vehicle and also the number of inter-connections, potentially saving production costs.

Such an architecture imposes two main disadvantages. Foremost, centralizing functions demands increased processing power in order to meet the functional requirements of the applications. Such processing power is not available at low cost and more expensive high-performance hardware platforms are necessary. Secondly, by grouping all the functionality related to a particular feature of the vehicle into a single place, some redundancy is lost and must be dealt with while also complying with the automotive standards [5].

## Domain controlled architecture structure

Following the functional classification of domains within the vehicle, a new domain controlled architecture is described in [5] and [6], which attempts to address the limitations of the traditional gateway architecture while preserving the flexibility, scalability and cost efficiency. The domain controlled architecture partitions functions and ECUs into domains which are further abstracted by a so called *domain controller*. Domain controllers are ECUs acting as gateways for their associated domains by managing the intra-domain communication networks and routing higher-level messages within and outside of the domains. Furthermore, by acting as integration platforms for the intra-domain ECUs, domain controllers save integration and development time [5]. Functionality is also grouped within a domain depending on the required flexibility. Critical software is integrated into the domain controllers and is always available while other lightweight functions are integrated into slave ECUs and can be turned on and off at any time for both flexibility and reduced power consumption.

The domain controllers further communicate and form a network together with the central gateway ECU which is commonly referred to as the *body controller*. The communication between the body controller and the domain controllers is realized using point-to-point connections for high speed and reliability as well as meeting the time constraints imposed by critical message paths between domains. The communication within a domain is still realised using traditional communication systems such as CAN, LIN, FlexRay and MOST. Each domain implements intra-domain communication accustomed to its performance needs.

## Mapping of vehicle functions to domains