Data Communication in the Automobile – Part 1: Architecture, Tasks, and Advantages of Serial Bus Systems

History
The first vehicles powered by gasoline engines already had electrical components, such as ignition coils, contact breakers, and spark plugs. These were quickly followed by other electrical devices such as headlights, brake lights, direction signals, interior lights, and windshield wipers. The introduction of components for entertainment and information, such as radios and record players and more recently cassette and CD players, meant that automobiles soon contained electronic components as well. Since the 1970s, electronics came along that improved or enhanced the functions of the vehicle itself or made driving easier. The next stage in the 1980s was directed at driving safety, including ABS and airbags, and expansion of comfort features such as air conditioning, mirror dimming and adjusting, power windows, and cruise control.

Growing competition brought about by increasing globalization ensured more and more innovation. Automobile manufacturers met this multifaceted challenge with electronics. The development of electronic components grew so rapidly in the 1990s that it is beyond the scope of this article to list all of the individual stages. Nowadays, the focus is on environmental friendliness and on the ability of automobiles to participate safely in road traffic as in self-driving cars, also known as autonomous. Vehicles are also starting to be connected with one another and with road infrastructure devices as well as with the Internet via WLAN. However, none of these functions would be possible in the automobile without data exchange between electronic components. And it is exactly this need for data exchange that has proven to be and still remains the real challenge.

Initially, a dedicated cable was used for the transmission of each separate information message (Figure 1). As the amount of information grew, however, the cable harnesses became so
large that their weight and multitude of connectors was problematic, not to mention the logistic challenges that arose during manufacturing, maintenance, and further development.

What is a “Bus”?
A pioneering solution for all of these problems was the introduction of the so-called bus. The word “bus” comes from the Latin word “omnibus”, which simply means “for all”. The many individual cables were thus replaced by a single cable that is shared by all information of all electronic control units (ECUs) (Figure 2).

However, it was then necessary to find ways to organize the timely transmission of this multitude of information over common wires. Different technologies arose, which we refer to as serial bus systems.

Before we delve into the specific characteristics of the individual bus systems in the subsequent articles in this series, we will start by explaining the technical fundamentals of the serial bus systems that are used in modern motor vehicles and then compare their underlying concepts.

A common characteristic of all buses is that each connected ECU shares a single input and output and, unlike in networks, information does not have to pass through the ECUs. Thus, when one ECU sends information on a bus, all other ECUs receive the information at almost the same time. The condition “almost” is necessary solely on account of the signal transit time on copper, which is approximately 5 ns/m.

Addresses Just Like the Post Office
For smooth information exchange, the data to be sent must be clearly allocated to its senders and receivers. We call this addressing. A general distinction is made between sender-selective and receiver-selective addressing. We are familiar with both of these types of addressing from our mailbox. With sender-selective addressing, the sender defines the desired receiver using a unique destination address. This corresponds to a standard letter with a destination address and return address.

In contrast to this, receiver-selective addressing identifies the information to be sent and not the ECUs. For this reason we talk about identifiers here and not addresses. We recognize deliveries like this in our mailboxes from the name (identifier) of the sending supermarket or home improvement store chain. Each receiver decides whether or not he uses the received information.

The sender pairs information and address respectively identifier thus allowing the receiver to recognize the unit. In transmission technology we refer to this unit as a “frame” because the addresses and information are framed by a start identification and end identification, which mainly serve to ensure error-free transmission between the sender and receiver. It’s also referred to as a message or packet.

The Protection of Data
The most important tasks of a serial bus system include timely data transmission at fast enough rates for the particular technical application and, above all, the protection of data during transmission. The use of a serial bus system in the automobile depends in large part on the degree of protection needed and the amount of data per unit of time that must be transmitted.

The protection of a serial bus system depends on how well the system prevents errors during data transmission and how well it detects remaining data corruption. The residual error probability is one measure for this protection. This is the product of probability A that the data to be transmitted are corrupted and probability B that corrupted data remain undetected. The lower the residual error probability, the more the data transmission is protected.

There are many different causes of data corruption. The most powerful are sparks from ignition plugs and electric motors. Other galvanic, capacitive, or inductive interactions and electromagnetic fields also play a role. Even reflections at the end of the bus cable are an internal cause of data errors on a serial bus. The more effective a bus is at eliminating or preventing these causes, the better the data transmission is protected.

A few important data protection measures are sufficient. These include shielding the transmission medium (cable or wire) and all electrical and electronic components. Alternatively the principle of differential signal transmission (Figure 3)
Such a system hardly emits any noise. The voltage difference fields around the wires cancel each other almost completely. Decr cables are exactly opposed and the resulting magnetic fields around the wires cancel each other almost completely. Such a system hardly emits any noise. The voltage difference between the two wires represents the signal from which external noise is subtracted. Sufficient clearance between the data transmission and power supply cables and between electrical and electronic components is helpful. It is also essential to limit the data transmission speed as well as the number and steepness of the data signal edges and to terminate the two bus ends with the characteristic impedance of the transmission medium to prevent reflections.

Nevertheless, transmission errors can never be fully eliminated, which is why error detection measures are needed. The checksum calculation method is most commonly used. With this method, the sender uses a defined algorithm to calculate a checksum from the data to be transmitted and includes it at the end of the message. A receiver can compare this checksum with the one that it has calculated from the received data. If the two checksums do not match, there is an error. The more sophisticated the algorithm and the longer the checksum, the better the data error detection capability. A checksum must not be too long, however, because every message in turn becomes longer and less data can be transmitted on the bus.

If an error is detected, the question arises as to how to correct it. Errors can be corrected based on the checksum contained in a message. However, this requires longer checksums and, in particular, immense computing capacity in the receiving ECUs. This correction method is not used in automobiles. Instead, the faulty message is discarded and a new transmission is requested.

**Is Information Transmitted Fast Enough?**

A bus system is regarded as capable of transmission in real-time [1], or real-time capable, if it can guarantee sufficiently fast transmission of all data that accumulates for an application. The essential factors for this are the number and size of messages, the available transmission speed (also referred to as bandwidth), and the bus access method of the ECUs. For the latter, a distinction is made between controlled bus access and random bus access (Figure 4). With controlled bus access, the bus access right of an ECU is already defined before its bus access. Such systems are called deterministic systems because it can be exactly determined or calculated when a particular ECU transmits which data. Deterministic behavior is an important precondition for achievement of real-time. Because the entire communication sequence runs according to schedule and can hardly be influenced, bus systems with controlled bus access are characterized by a poor dynamic response. Bus systems with uncontrolled bus access avoid this disadvantage. Each ECU has the right to transmit data at any time. This results in a very fast bus access but also poses the following risk: Depending on the density and length of the messages and the available bandwidth, there is a greater or lesser acute risk of collision. This is not an ideal basis for real-time.

If all ECUs monitor the bus continuously and send information only when the bus is available, the risk of collision is significantly reduced, but not fully reduced. This risk is eliminated altogether by introducing priorities for information, which can be recognized on the CAN bus with help of the identifier. However, even these bus access methods cannot guarantee timeliness (real-time). The reason is that, as a result of prioritization, there is a risk that messages with lower priority will be subject to long delays and no longer be received in a timely manner.

---

3

---

**Figure 4:** Controlled and random bus access

**Figure 5:** Architecture of serial bus systems
Architecture of Serial Bus Systems

Based on the reference model for data communication specified by the ISO (International Standardization Organization), the serial bus interface of an ECU in the automobile is generally distributed among two (communication) layers. The physical layer implements the physical bus connection including the physical signal transmission. Above that is the data link layer with its tasks including addressing, framing, bus access, synchronization, and error detection and correction (Figure 5).

The physical bus connection is made with help of a transceiver. A communication controller takes over the tasks of the data link layer. If all ECUs on the bus use the same type of transceivers and communication controllers, the basic preconditions exist for smooth data exchange.

During serial communication, the sender’s application delivers the data block to be sent together with addresses or identifiers to the communication controller. Check and synchronization information as well as start and end identification are added, so that a frame results. The transceiver now sends the frame on the bus.

Most buses in automobiles are in the form of a cable to which the ECUs are connected via short spur lines. This is called a line topology or bus topology. On the receiver side, the transceiver passes the frame to the communication controller, which checks the information, and if data is received correctly, forwards the data block to the application.

For some tasks such as bus management, that is, the concerted putting to sleep and waking up of all ECUs, and the diagnosing and configuring of ECUs, the functions of the physical layer and the data link layer are not sufficient. The higher layers of the ISO reference model for higher communication protocols are then used in order to achieve the required communication functionality.

Cross-Vendor Bus Technologies

Intensified competition is ensuring more and more safety and comfort functions in automobiles. As a result, there is not only a continual increase in the number of electronic components in vehicles but also a significantly higher degree of networking and a rapid rise in data volumes since most new automobile functions rely on data exchange. The challenge arising from this is to keep the increasing complexity under control to ensure the continued quality and reliability of functions. For this purpose, the automobile industry has developed standards, such as “AUTOSAR” (AUTomotive Open System ARchitecture, Figure 6), which provide the necessary transparency on a system or function level. Cross-vendor communication standards such as the serial bus systems CAN [2], LIN [3], FlexRay [4], MOST [5], and Ethernet [6] ensure more transparency on the lower communication levels.

CAN (Controller Area Network) is used mainly in the drive and chassis areas and in the operation of the vehicle. LIN (Local Interconnect Network) is used to control simple comfort functions, such as the air conditioning, seats, mirrors, and windows. MOST (Media Oriented System Transport) has long served the infotainment area with transmission of video and audio signals that require large bandwidths. Also Ethernet is increasingly being used for this. Currently, Ethernet is mainly used for diagnostics including flashing. In the future, however, its main use will be in the driver assistance area, including the park assist and autonomous driving sub-areas. Finally, FlexRay should enable the most demanding communication in safety-critical applications such as electronically controlled steering and braking. However, lawmakers along with the automakers are finding it difficult to push forward here, by trying to avoid the risks in these areas that are highly critical to safety.

CAN was developed in the early 1980s by Robert Bosch GmbH and became an international standard (ISO 11898) in 1994. The founders of Vector played a central role in this development. LIN (ISO 17987), FlexRay (ISO 17458), and MOST came from cross-vendor organizations, namely the LIN Consortium, FlexRay Consortium, and MOST Cooperation. Vector [7] supports automobile manufacturers and suppliers in networking using CAN, LIN, FlexRay, MOST, and Ethernet with a consistent set of design and development tools as well as with software components and basic software for AUTOSAR ECUs. Advice, consulting services and tools for process management supplement the areas of application.

A comprehensive training program that includes basic seminars for CAN, LIN, FlexRay, and MOST rounds out the services. Parts 2 to 5 of this series cover the details of the serial bus systems CAN, LIN, FlexRay, and MOST.

Closing Remarks About the Term “Real-Time”

The term “real-time” is often used loosely or imprecisely, because it is not very easy to grasp. Because I had to use it at the outset, I would like to clarify a few facts about this subject. Whoever wants to know more can draw on other sources. [1]

Part 9 of DIN 44300 (Information Processing), which has been replaced by DIN ISO/IEC 2382, defines real-time as follows: “Real-time refers to the operation of a computing system on which data processing programs are always operable such that processing results are available within a specified time span. The data may occur randomly or at predefined times, depending on the application.”
For a real-time capable system, it is thus not enough to deliver a measurement or calculation result with the correct value. Rather, this also has to occur within a specified response time. If this is not the case, the system has failed. According to the theory of real-time systems, the required response time must be calculated for an application running on the system. People often speak carelessly about "real-time" if a program runs without a noticeable delay. However this definition is not very accurate. It is wrong to use "real-time" as a synonym for "very fast", because real-time systems even have to schedule no-load operations in order to also meet real-time requirements under high load. A distinction is made between hard real-time and soft real-time. The distinguishing criterion is the different consequences of a violation of the real-time requirements.

- **Hard real-time requirements:** If the system exceeds the required response time, it has failed. Real-time systems must always supply the correct result within the required response time, and the user of a hard real-time system must be able to rely on this. (For example, engine control: if the requirements are violated, the engine sputters and even damage may occur.)

- **Soft real-time requirements:** Such systems typically meet the required response time but not always. Thus, for example, the required response time reaches only an average or satisfies a different statistical criterion. The time requirements are not absolutely strict, but rather are viewed as guidelines. Exceeding the required response time is not regarded as a failure. It can be exceeded frequently as long as it still falls within a tolerance range. Or, the response time can far exceed the required response time occasionally. (For example, a video conference: When response time requirements are violated, the image “jerks” but work can continue.)

**Literature References:**

[3] de.wikipedia.org/wiki/Local_Interconnect_Network

Ernst Christmann, Physicist, Mathematician

is a Senior Technical Trainer in the area of software tools for ECU testing and development and has worked for Vector Informatik GmbH in Stuttgart since 2004.