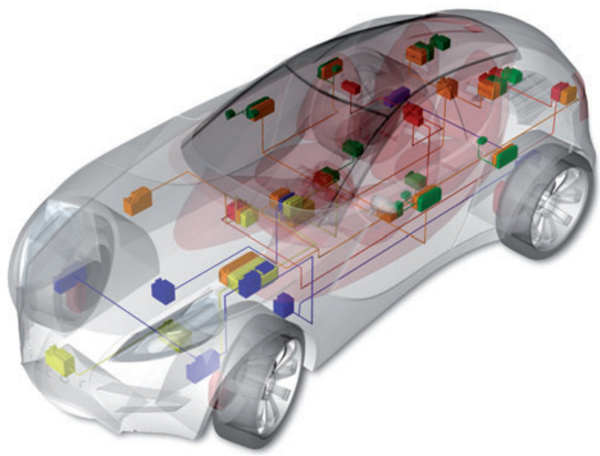
Part 4:

FlexRay for data exchange in highly critical safety applications

FlexRay is going into production for the first time. It will appear on the BMW X5, which was presented to the public at the Paris Auto Salon in August 2006, and it can be purchased in Germany beginning in March of this year. Within its active chassis system, FlexRay provides for secure and reliable data transmission between the central control module and the four satellite ECUs, one located at each shock absorber. This article traces FlexRay's path into the automobile and explains the key principles of FlexRay bus technology.



According to the German Federal Statistics Office [1] driving on Germany's roads was never so safe as in the year 2005. Although vehicle registrations grew considerably, there was a nearly one percent reduction in accidents involving personal injury (336619) compared to the prior year. There were also significant reductions in the number of traffic deaths (5361, -8.2%), serious injuries (76952, -4.6%) and minor injuries (356491, -1 %). This trend was continued in 2006: Between January and August, 3260 traffic participants were killed, and this represents a 7.8 percent reduction compared to the prior year. The number of injured dropped by 5.8 percent over the same time period.

Decisive in lowering the number of accidents and reducing the severity of accident outcomes are active safety systems and assistance systems that support drivers in their task of driving the vehicle. One study by a number of well-known automotive OEMs showed, for example, that ESP reduced the number of skidding accidents by up to 80 % [2]. Making just as important a contribution to reducing the severity of accident outcomes are increasingly safer passenger cells and optimized restraint systems.

In light of the goal of halving traffic fatalities by the year 2010, the automotive industry is focusing on further developing existing active safety systems and driver assistance systems and developing new innovative systems. Since these systems not only provide information and instructions, but often also make corrective interventions and assume driving tasks, it is no longer possible to do without electronic interfaces to the chassis and drivetrain. The combination of brake-by-wire and steer-by-wire systems is thought to have great potential.

Requirements of future data transmission in the automobile

Implementations of ever more challenging safety and driver-assistance functions go hand in hand with the increasingly more intensive integration of electronic ECUs in the automobile. These implementations require very high data rates to transmit the increasing number of control and status signals. They are signals that not only need to be transmitted extremely quickly; their transmission also needs to be absolutely deterministic.

That is the reason for the growing importance of communication systems that guarantee fast and deterministic data transmission in the automobile. Potential use of by-wire systems further requires the design of fault-tolerant structures and mechanisms. Although by-wire systems may offer wide-ranging capabilities and the benefits of increased design freedom, simplified assembly, personalization of the vehicle, etc., data transmission requirements in the automobile are elevated considerably, because these systems belong to the class of fail-operational systems. They must continue to operate acceptably even when an error occurs.

CAN cannot satisfy these requires due to its event-driven and priority-driven bus access, its limited bandwidth of 500 KBit/sec based on physical constraints in the automobile, and lack of fault-tolerant structures and mechanisms [3].

FlexRay – The answer to heightened data transmission requirements in the automobile

The certainty that CAN could hardly be expected to satisfy growing data transmission requirements in the automobile over the mid-term, led to the development of a number of deterministic and fault-tolerant serial bus systems with far greater data rates than CAN. Examples include: TTP (Time Triggered Protocol) [4], Byteflight [5] and TTCAN (Time Triggered CAN) [6]. Although a development partnership was created as early as 2001 between Audi and the TTP promoting company TTTech, and although Byteflight was successfully applied to BMW 7-series cars in 2001, it is FlexRay that has prevailed in the automotive industry.

An important reason for the success of FlexRay was the founding of the FlexRay Consortium [7], under whose auspices the two motor vehicle manufacturers DaimlerChrysler and BMW and the two chip producers Motorola and Philips joined forces in the year 2000. Based on Byteflight bus technology originally developed by BMW, the FlexRay Consortium created the cross-OEM, deterministic and fault-tolerant FlexRay communication standard with a data rate of 10 MBit/sec for extremely safety- and time-critical applications in the automobile.

Today the FlexRay Consortium is made up of seven “core partners”: BMW, Bosch, DaimlerChrysler, Freescale, General Motors, Philips and Volkswagen. Gradually, a number of Premium Associate Members (including Vector Informatik [8]) and Associate Members joined the organization.

Making a significant contribution to the success of FlexRay was the detailed documentation of the FlexRay specification. The two most important specifications, the communication protocol and the physical layer, are currently in Version 2.1. These and other FlexRay bus technology specifications can be downloaded from the homepage of the FlexRay Consortium [7].

FlexRay communication architecture – Time-triggered, fault tolerant and flexible

Just as in the case of data communication in a CAN cluster, data communication in a FlexRay cluster is also based on a multi-master communication structure. However, the FlexRay nodes are not allowed uncontrolled bus access in response to application-related events, as is the case in CAN. Rather they must conform to a precisely defined communication cycle that allocates a specific time slot to each FlexRay message (Time Division Multiple Access - TDMA) and thereby prescribes the send times of all FlexRay messages (Figure 1).

Time-triggered communication not only ensures deterministic data communication; it also ensures that all nodes of a FlexRay cluster can be developed and tested independent of one another. In addition, removal or addition of FlexRay nodes in an existing cluster must not impact the communication process; this is consistent with the goal of re-use that is often pursued in automotive development.

Following the paradigms of time-triggered communication architectures, the underlying logic of FlexRay communication consists of triggering all system activities when specific points are reached in the time cycle. The network-wide synchronism of FlexRay nodes that is necessary here, is assured by a distributed, fault-tolerant clock synchronization mechanism: All FlexRay nodes not only continuously correct for the beginning times (offset correction) of regularly transmitted synchronization messages; they also correct for the duration (slope correction) of the communication cycles (Figure 2).

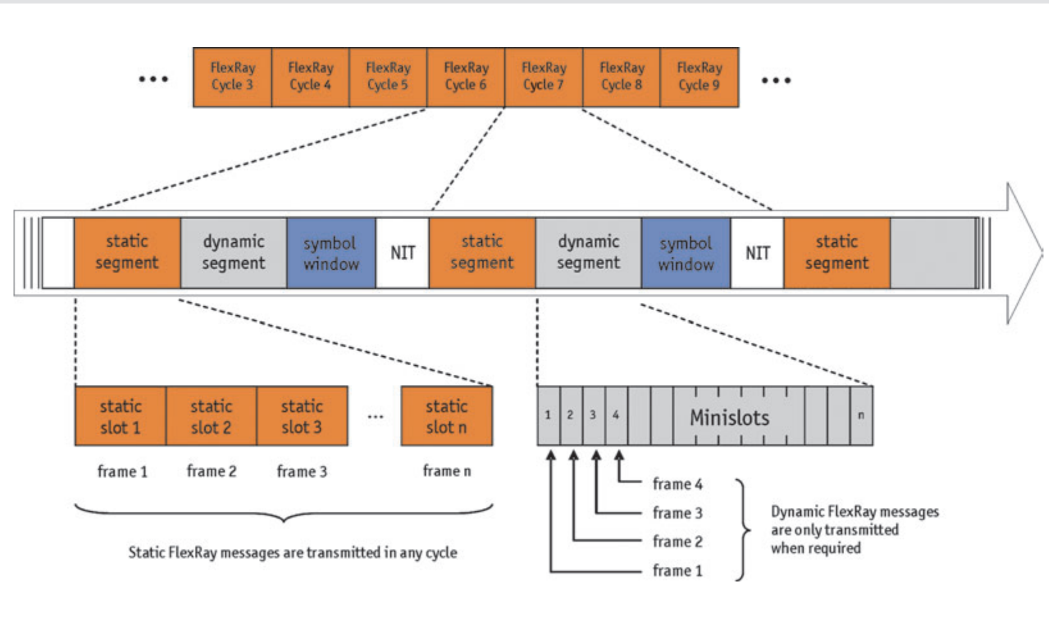


Figure 1: Principle of FlexRay communication.

This increases both the bandwidth efficiency and robustness of the synchronization.

FlexRay communication can be based on either an electrical or optical physical layer. Speaking in favor of electrical signal transmission is its simplicity, which brings cost advantages. The comparatively cost-intensive optical signal transmission is characterized by substantially better electromagnetic compatibility (EMC) compared to electrical signal transmission.

FlexRay communication is not bound by a specific topology. A simple, passive bus structure is just as feasible as an active star topology or a combination of the two (Figure 3). The primary advantages of the active star topology lie in possibility of disconnecting faulty communication branches or FlexRay nodes and - in designing larger clusters - the ability to terminate with ideal bus terminations when physical signal transmission is electrical.

To minimize failure risk, FlexRay offers redundant layout of the communication channel (Figure 4). This redundant communication channel could, on the other hand, be used to increase the data rate to 20 Mbit/sec. The choice between fault tolerance and additional bandwidth can be made individually for each FlexRay message.

Finally, an independent control mechanism (Bus Guardian) ensures that a FlexRay node only gets access to the bus during its turn in the communication cycle. This prevents bus monopolization by a defective FlexRay node (babbling idiot).

FlexRay communication: Deterministic and dynamic

Each communication cycle is equal in length and is essentially organized into a static time segment and a dynamic time segment (Figure 1). Of central importance here is the static segment that begins each communication cycle. It is subdivided into a user-definable number (maximum 1023) of equally long static slots.

Each static slot is assigned to a FlexRay message to be sent by a FlexRay node. Assignments of static slots, FlexRay messages and FlexRay nodes are made by slot number, message identifier (ID), and the value of the slot counter implemented on each FlexRay node. To ensure that all FlexRay messages are transmitted at the right time and in the correct sequence in each cycle, the slot counters on all FlexRay nodes are incremented synchronously at the beginning of each static slot. Because of its guaranteed equidistant and therefore deterministic data transmission, the static segment is predestined for the transmission of real-time relevant messages.

Following the static segment is an optional dynamic segment that has the same length in every communication cycle. This segment is also organized into slots, but not static slots, rather so-called minislots (Figure 1). Communication in the dynamic segment (minislotting) is also based on allocations and synchronous incrementing of the slot counters on the FlexRay nodes.

However, it is not mandatory to transmit the FlexRay messages associated to the minislots with each communication cycle, rather they are only sent as needed. If messages are not needed, the slot

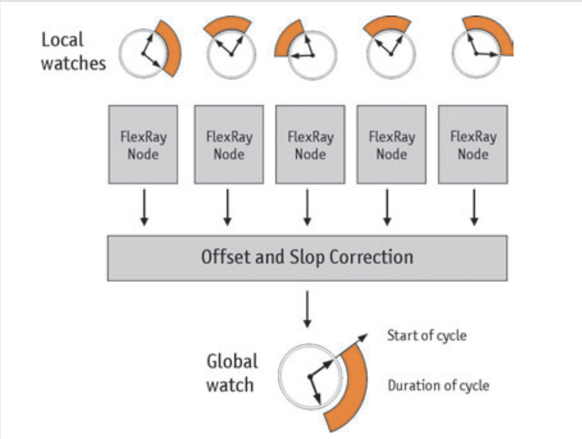


Figure 2:
Clock synchronization.

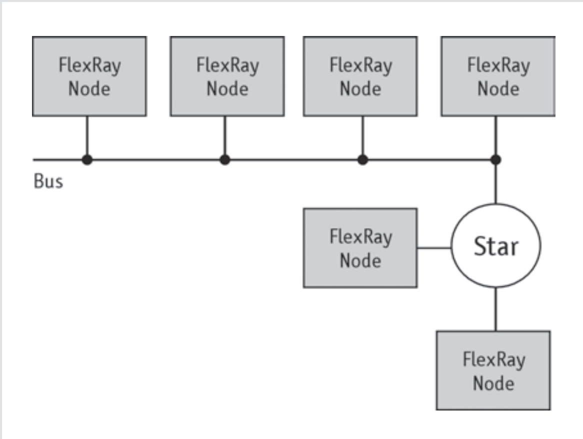


Figure 3:
Combined topology of passive bus and active star.

counter of a minislot is incremented after the defined time period. While a (dynamic) FlexRay message is being transmitted, incrementing of the slot counter is delayed by the message transmission time (Figure 5).

The allocation of a dynamic FlexRay message to a minislot implicitly defines the priority of the FlexRay message: The lower the number of the minislot, the higher the priority of the dynamic FlexRay message, the earlier it will be transmitted, and the higher the probability of transmission given a limited dynamic time segment length. The dynamic FlexRay message assigned to the first minislot is always transmitted as necessary, provided that there is a sufficiently long dynamic time segment.

In the communication design it must be ensured that the lowest priority dynamic FlexRay message can be transmitted too – at least provided that there are no other, higher priority needs. The designer of a FlexRay cluster must also ensure that transmission of the longest dynamic FlexRay message is even possible. Otherwise, the communication design would not make any sense.

The communication cycle is completed by two additional time segments (Figure 1). The "Symbol Window" segment serves to check the functionality of the Bus Guardian, and the "Network Idle Time – NIT" time segment closes the communication cycle. During the NIT the FlexRay nodes calculate the correction factors needed to synchronize their local clocks. At the end of the NIT, an offset correction is made if necessary (the slope correction is always distributed over the entire communication cycle). There is no data transmission during the NIT.

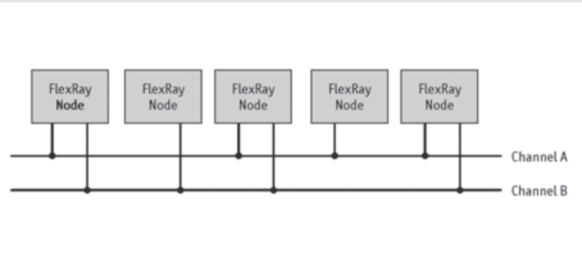


Figure 4:
Passive bus structure with two communication channels minimizes failure risk.

CRC-protected data transmission

The signals in a FlexRay cluster are transmitted by the well-defined FlexRay message, wherein there is essentially no difference in the formats of the FlexRay messages transmitted in the static segment and those transmitted in the dynamic segment. They are each composed of a header, payload and trailer (Figure 6).

The header comprises the five-bit wide status field, ID, payload length and cycle counter. The header-CRC (11 bits) protects parts of the status field, ID and payload length with a Hamming distance of 6. The ID identifies the FlexRay message and represents a slot in the static or dynamic segment. In the dynamic segment the ID corresponds to the priority of the FlexRay message. The individual bits of the status field specify the FlexRay message more precisely. For example, the "sync frame indicator bit" indicates whether the FlexRay message may be used for clock synchronization.

After the header comes the so-called payload. A total of up to 254 useful bytes may be transported by one FlexRay message. The trailer encompasses the header and payload-protecting CRC (24 bit). Given a payload of up to 248 useful bytes, the CRC guarantees a Hamming distance of 6. For a larger payload the Hamming distance is 4 [8].

In networking issues: Achieving objectives rapidly with external expertise

In the year 2001, Vector Informatik was already offering the first product solution for the development of FlexRay systems. In the meantime, developers can now obtain a comprehensive portfolio of products [9]. These include tools for designing, developing, simulating, analyzing, testing and calibrating ECUs and distributed networks. DaVinci Network Designer FlexRay gives the developer an environment for efficiently designing network architecture and communication relationships. Simulation, analysis and testing of FlexRay systems are performed with CANoe.FlexRay, whose multibus concept enables simultaneous operation of FlexRay, CAN, LIN and MOST bus systems. For precise study of the FlexRay network's system behavior in response to errors and disturbances, FRstress generates them on a channel in the FlexRay cluster. For direct access to internal ECU variables, the developer needs a special measurement and calibration protocol: XCP on FlexRay. In the context of the development of the active chassis system on the new BMW X5, BMW engineers implemented Vector's measurement, calibration and diagnostic tool CANape. As the XCP-on-FlexRay Master, CANape measures and calibrates individual ECU parameters directly via FlexRay. Besides software, Vector also develops stacks for ECUs.

FlexRay software components make it possible to interconnect applications with different bus or operating systems in an uncomplicated way. For hardware access to FlexRay buses, suitable bus interfaces connect to the USB, PCI and PCMCIA ports of a PC or notebook computer.
The Vector Academy [10] can teach the basic knowledge needed to quickly become familiar with the diverse development activities related to ECU communication in the automobile. This knowledge is shared in the context of seminars on CAN, LIN, FlexRay and MOST.

Literature and links:
[1] www.destatis.de [2] www.bosch.com
[3] Mayer, E.: Datenkommunikation im Automobil – Teil 2: Sicherer Datenaustausch mit CAN im Automobil ["Data communication in the automobile – Part 2: Reliable data exchange with CAN in the automobile"]. In: Elektronik Automotive 8/2006, pp. 34-37
[4] www.tttech.com [5] www.byteflight.com
[6] www.can-cia.org/can/ttcan [7] www.vector-informatik.com
[8] www.flexray.com [9] www.flexray-solutions.com
[10] www.vector-academy.com

Eugen Mayer
(Graduate Engineer with Technical Teaching Certificate), after completing his vocational training to become a communications technician, studied electronics at the Technical College in Ravensburg/Weingarten, Germany, and studied electrical engineering and vocational teaching at the University of Stuttgart. Since 1999 he has been working at Vector Informatik where he is employed as a Senior Trainer.
Tel. +49-711/80670-574,
Fax +49-711/80670-111,
E-Mail: eugen.mayer@vector-informatik.de

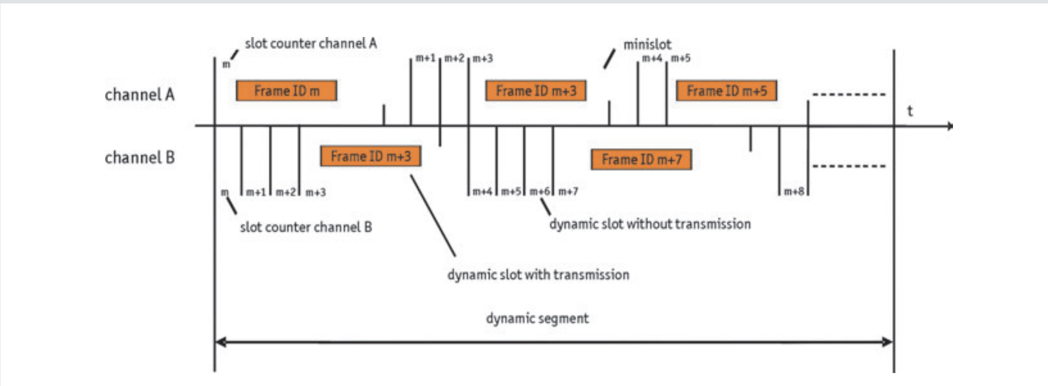


Figure 5: Example of communication flow in the dynamic time segment.

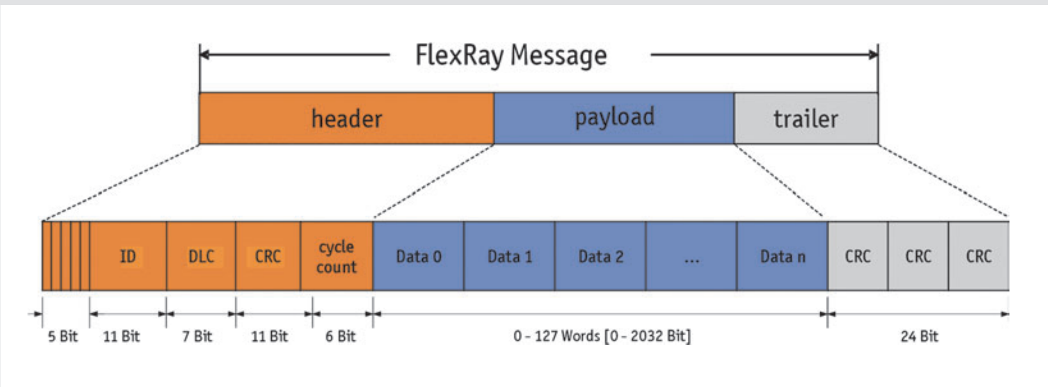


Figure 6: Structure of the FlexRay message with header, payload and trailer.