Seoul 2000 FISITA World Automotive Congress
June 12-15, 2000, Seoul, Korea

F2000G316

byteflight - A New Protocol for Safety Critical Applications

Josef Berwanger*, Martin Peller, Robert Griessbach

BMW AG, EE-211 Development Safety Systems Electronics, Knorrstrasse 147, 80788 Munich, Germany

There is a clear trend in automobile development towards using electronic systems instead of mechanical components and towards realizing an increasing number of convenience and safety functions while at the same time taking into consideration the environment. Consequently, vehicle electronics are becoming increasing complex and the number of functions, sensors, and actuators is growing fast. These requirements cannot be met with a central control unit. The solution is to network the various electronic modules using a high-performance data bus. This makes it possible to reduce the complexity of the wiring and to make multiple use of sensor data in order to integrate the individual functions in such a way that they form an intelligent overall vehicle. The high-performance bus required to make this possible is now available: Together with the companies Motorola, ELMOS, Infineon and Tyco EC, BMW AG has developed the *byteflight* high-performance data bus for safety-related applications. Fully integrated, inexpensive electronic components are already available for the construction of a complete bus system based on fiber optics technology. The first application in high-volume automobile production will start within the next year and will include passive-safety and body functions.

Keywords: High Speed Databus Protocol

1 INTRODUCTION

A data bus system that is to be used in safety-related areas must satisfy the highest requirements concerning transmission rate, reliability and interference immunity. To make it possible to integrate safety-related functions, the data protocol must be fault-tolerant and deterministic, and must feature excellent data integrity. Another fundamental prerequisite for such a system is for it to be compatible with the current background issues of automobile development (such as ever-shortening development periods, fast reaction times for new market requirements, consideration of many different vehicle and equipment variations, modular construction optimized for installation space, and high quality standards with low system costs). Good diagnosis capabilities for effective service demand flexible utilization of the bandwidth. Easy expansion capability is required to accommodate the rising number of functions. Since none of the previously known data protocols were able to meet all of these requirements, BMW has developed the *byteflight* high-performance data bus system (1)**.

2 DATA COMMUNICATION

2.1 DATA TRANSMISSION METHOD

Data-transmission methods that have been available until now have been either event-controlled or time-controlled.

Event-controlled (or asynchronous) protocols (e.g. CAN) only transmit data when a transmission request is present. Bus access is usually controlled either by priority or by chance. Nevertheless, latency times can only be determined statistically, and are therefore not suitable for hard real-time requirements. The advantage of this new protocol is the flexible use of the bandwidth by different nodes and the ease of system expansion.

Data protocols that are purely time-controlled (synchronous, e.g. TTP) grant transmission time to each node in accordance with a pre-defined sequence. With this protocol, the latency times for the messages are deterministic. The number of messages is prescribed by the number of time slots, and therefore cannot generally be changed during operation. Assignment of the bandwidth to different nodes is permanent, so that later expansion to include new nodes or new functions require reconfiguration of the entire system.

The *byteflight* protocol unites the advantages of the synchronous and asynchronous methods. It guarantees deterministic latencies for a specific number of high-priority messages and flexible use of transmission bandwidth for low-priority messages. For special cases, it is possible to configure the software driver to allow a purely synchronous or purely asynchronous operating mode. Furthermore, the protocol permits easy expansion and adaptation for different variations in the equipment and functions. The transmission rate of 10 Mbps should meet the high requirements of next vehicle generations.

2.2 FTDMA (FLEXIBLE TIME DIVISION MULTIPLE ACCESS)

The basis for *byteflight* data communication (2) is formed by cyclical synchronization pulses (SYNC pulses). These pulses make a common time-base available for all

* Corresponding author. e-mail: Josef.Berwanger@bmw.de

** Numbers in parentheses designate references at the end of paper.

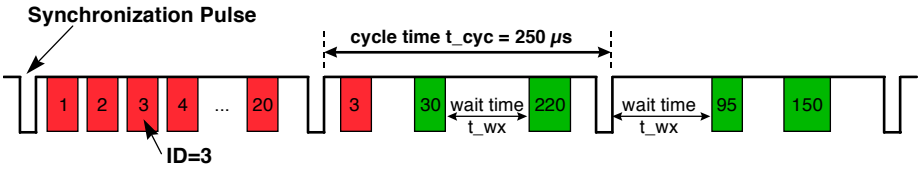

Figure 1 - Flexible Time Division Multiple Access - FTDMA

nodes and for the application software. Any *byteflight* node can be configured as a SYNC master that generates the clock base for all nodes. The cycle time is the time interval between the synchronization pulses. In the current implementation of *byteflight*, this interval is set to 250 µs at data rate of 10 Mbps (Figure 1).

In between the synchronization pulses, all nodes can send messages. In this process, all nodes are on the same footing. Bus access is governed autonomously and time-controlled by the protocol controller implemented in the hardware. This is done in such a way that an ascending identifier sequence occurs within a cycle. An arrangement for all bus system nodes establishes that an identifier may only be used by one single node. This ensures that there are no collisions on the bus. The bus protocol controller, which works autonomously from the host CPU, also ensures that a single given identifier is not sent more often than once per cycle. The point of time for bus access is controlled solely by the bus controller, and cannot be influenced by the host CPU. The CPU can only present the data to be transmitted via a Dual Port RAM, or read out the received data.

Bus access is regulated according to the Flexible Time Division Multiple Access method (FTDMA). With this method, all nodes start 'slot counters' that are triggered by the synchronization pulse. The slot counters begin at zero and count up to the highest identifier value possible. When a slot counter reaches an identifier value for which a transmission request is present, the corresponding message is transmitted via the bus, and all the slot counters stop at the current value for the duration of the transmission. Once the transmission is complete, the slot counters begin to count upwards again. Figure 2, which was recorded with a logic analyzer, shows an example of this transmission process (3).

The slot counters for nodes A and B are shown. Node A sends identifier 4, node B sends identifiers 1 and 7. Transmission slots 1, 4, and 7 (shown here in red) last as long as necessary to transmit the message. Since there are no transmission requests present for identifiers 2 and 3 in this cycle, slots 2 and 3 are not used and appear only as very short waiting slots (shown here in green). For a message to be sent in a cycle, its transmission request must have been present at the rising edge of the sync pulse for that cycle.

The waiting period between the message with identifier ID_{t-1} and the following message with identifier ID can be calculated with the following formula (2):

Waiting period $t\_wait = t\_0 + t\_delta * (ID - ID_{t-1})$

The fixed portion of the waiting period t_0 is used for maintaining the minimum interval of 1100 ns between two messages. Since the optimal values for t_0 and t_delta are derived from the signal propagation delays within a bus system, these parameters have been made programmable in the *byteflight* modules.

The FTDMA procedure described above is thus a purely time-controlled bus-access process. Nevertheless, it allows the guaranteed or deterministic transmission of a specific number of high-priority messages in every communication cycle even when the bus capacity is fully used, and simultaneously permits flexible and statistic assignment of the bandwidth for the rest of the messages if bus load is low enough.

This can also be seen in Figure 3. In the operational mode depicted, the 10 highest-priority messages are transmitted synchronously and cyclically every 250 µs. The second portion of the communication cycle can be used for event-controlled messages.

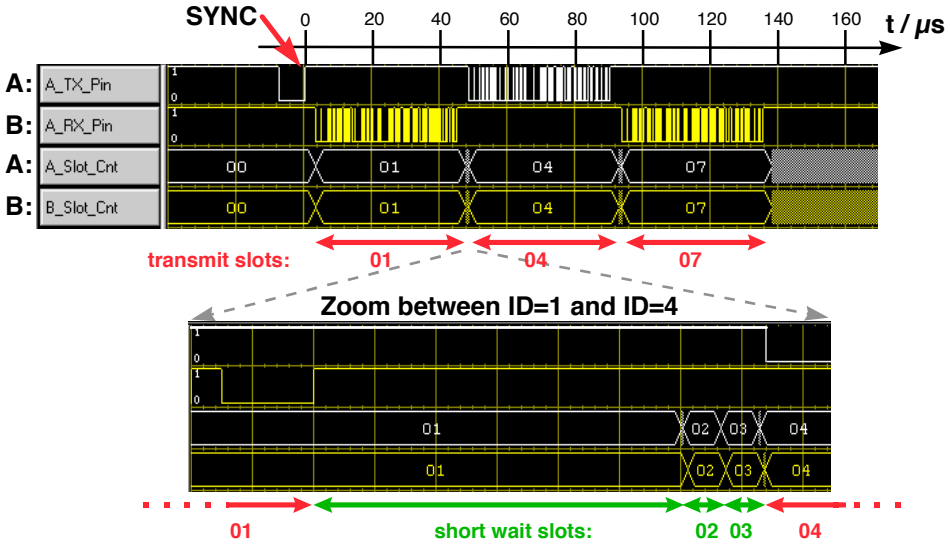

Figure 2 - Distributed synchronized slot counters

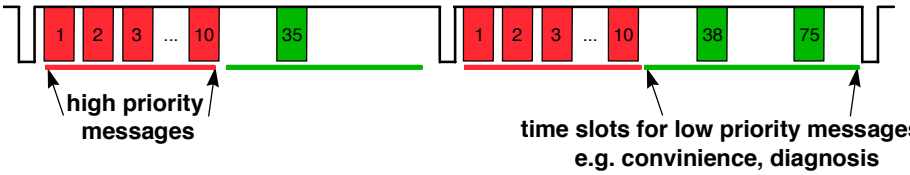

Figure 3 - Synchronous and asynchronous transmission

Despite the guaranteed transmission of a specific number of high-priority messages, flexibility is not lost. This is a factor of growing importance as development periods shorten and the number of vehicle and equipment variations rise. In this way, it is possible to add high-priority messages, and the guaranteed transmission of these messages can be proved analytically. It is also possible to add asynchronous messages. In this case, verification of the latency times can be accomplished via statistical observations, as with the CAN protocol, for example. It is not necessary to make software changes at nodes that are already used in the system.

2.3 MESSAGE STRUCTURE

byteflight messages (Figure 4) consist of a 6-bit starting sequence, an identifier byte, a length byte, up to 12 data bytes, and two CRC (Cyclic Redundancy Check) bytes. The CRC used ensures a hamming distance of h = 6 to provide a high level of data integrity. For bit synchronization, each byte is framed by one start bit and one stop bit. The bit length is 100 ns at a gross data rate of 10 Mbps.

2.4 REDUNDANT SIGNAL CHANNEL FOR THE ALARM STATUS

The *byteflight* protocol offers another special characteristic that is particularly necessary for passive safety systems: To show an alarm or arming status, the SYNC master has the capability of changing the type of synchronization pulse that is used. This status is recognized by all *byteflight* controllers and the physical driver modules, and can be used to enable specific safety functions. The switching of the synchronization pulse has no impact on the communication sequence or protocol behaviour.

2.5 STRATEGIES FOR ERROR HANDLING

Since the data transmission of safety-related signals is generally cyclical and has an update rate of 250 μs, there is no repeat mechanism at protocol level for transmission errors. The protocol has been configured such that at the latest, even when there is heavy interference, communication resumes when the next synchronization pulse is sent. Furthermore, in order to detect even the smallest deviations of the clocks in the *byteflight* nodes, there are high-performance mechanisms for recognizing synchronization errors at both the cyclical and message levels and error-handling strategies based on this. Only messages that have been received with a valid CRC are made available for the host CPU to read.

Due to the optical data transmission that is available, a star network topology is used in the pilot application. In addition, a specially developed star coupler module makes it possible to shut off nodes that, for example, do not follow the bus protocol or cause transmission errors. Thanks to their steady light shut-off function, the optical transceivers can prevent the bus being blocked by a permanently on transmitter.

With bus topologies, it is not possible to shut off faulty nodes from a central point. Here, however, thanks to the purely time-controlled access, the decentralized use of a bus guardian with its own clock supply is possible at every node.

If the SYNC master, and thus the synchronization pulse, fails to function, a specific node (substitute SYNC master) can be configured by its μC as a master. This node then sends the sync pulses, and the data transmission is guaranteed.

Additionally, it is possible to select a redundant system configuration. Two *byteflight* clusters can, for example, be operated in parallel, each having a SYNC master.

3 AVAILABLE BUS CONTROLLERS

For the design of a *byteflight* node, a Motorola **MC68HC912BD32** with an integrated *byteflight* controller (4) can be used. The 16-message buffer can be freely configured as a transmission buffer, FIFO-receive buffer or dedicated receive buffer. The FIFO-receive function also has a programmable acceptance filter. Each dedicated receive buffer filters just one specific identifier out of the flow of data and always contains the most up-to-date message for this identifier. The microcontroller is available in an 80-pin QFP package and is compatible with the pins and software of Motorola's 68HC912B family. The controller has 1 kByte of RAM, a 768-Byte EEPROM and 32-kByte flash EEPROM. The flash memory can be programmed via a bus download and therefore makes it possible to perform a quick software update for the vehicle without the time-consuming step of removing the control units.

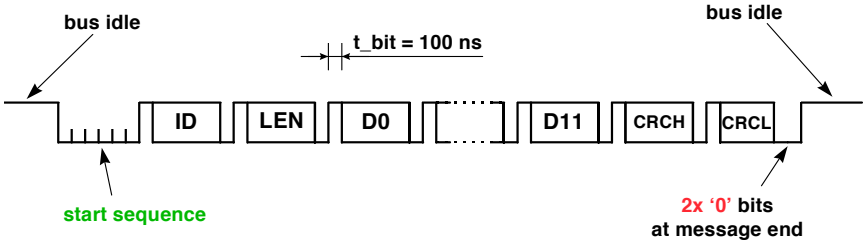

Figure 4 - Message format

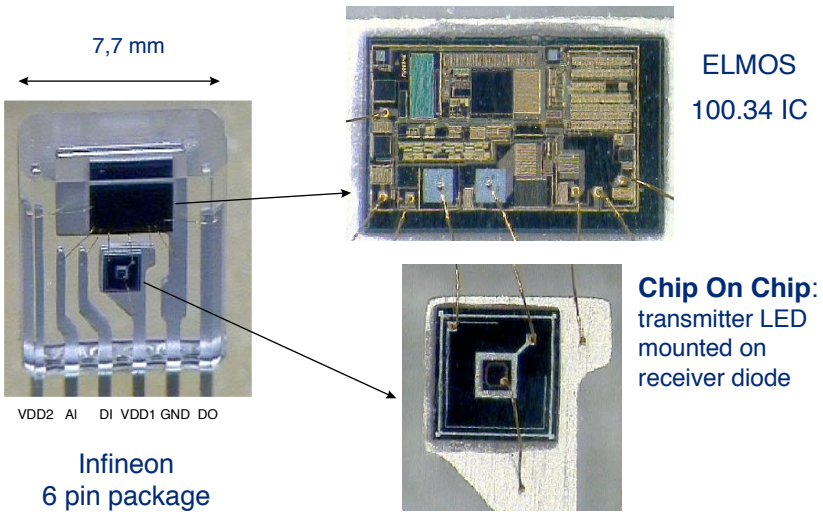

Figure 5 - Optical transceiver for bidirectional transmission

The **E100.38** standalone *byteflight* controller (5) available from ELMOS is register-compatible to the HC12BD32. It offers the possibility of building up *byteflight* control units using other processor types from the HC12 family, Motorola PowerPC, Siemens C16x, or other microcontrollers commonly available on the market. The standalone controller is housed in a 44-pin QFP package.
All modules support optical diagnosis, optical wake-up via the data bus, and sleep modes. A standby current lower than 50 μA per *byteflight* node is reached.

4 TOPOLOGY

Due to the high data rate, the issue of electromagnetic compatibility takes on special significance. Electrical links would have to be protected against external influences by implementing special measures during the manufacturing of the vehicle harness. This is usually accomplished by using shielded or twisted cables. Despite the high production complexity (and thus expense) associated with this, it does not provide electrical isolation of the individual components. For this reason, a physical layer was developed based on plastic optical fiber (POF) technology and implemented as a star topology.

For data bus systems, optical transmission has the advantage that it remains undisturbed by electromagnetic interference. Additionally, connected system components are electrically isolated from one another, which rules out the possibility of the individual modules influencing each other in an undesired manner.

Thanks to the star-shaped bus structure, the rest of the network (with all safety components) is still fully functional when one of the system components is removed.

The individual nodes are connected together via fiber-optic cables and an active star coupler. The fiber-optic lines are operated bidirectionally using half-duplex transmission. In order to connect each node, there is an optical transceiver module (6) made by Infineon (formerly Siemens Semiconductor) inside the star coupler. Transmitting and receiving diodes, and the ELMOS **E100.34** (7) transceiver chip are integrated into this transceiver module. This module (Figure 5) contains the LED driver circuit and the receiver amplifier with peak detection for transforming the photo-current into digital levels. There are also several logical functions (such as a shut-off for constant light, and detection of the synchronization pulse). Furthermore, there is an integrated optical diagnostic function for monitoring the optical transmission quality. The transceiver's light-emitting diode (LED) and photodiode are mounted one above the other using chip-on-chip technology in order to achieve optimal coupling of both components with the optical fiber.

The transceiver modules have been integrated directly into the control unit plug connector, since optimization of the coupling losses was to be combined with the elimination of the need for a cost-intensive adjustment process. To make this possible, Tyco EC (formerly Siemens EC) – working in close cooperation with BMW – and Infineon designed a special transceiver housing (8) that significantly simplifies installation of the plug connector and simultaneously permits very precise positioning of the cable's plug-in elements. Furthermore, new cable harness components and production processes were jointly developed in order to make it possible to realize reliable and cost-effective mass production with conventional, fully automatic equipment while simultaneously meeting the high requirements for processing reliability and the capacity for handling mechanical and thermal loads.

Figure 6 shows the schematic diagram of a *byteflight* network.

The active star coupler consists of the ASIC **E100.39** (9) from ELMOS. It receives the bus signals from the individual nodes via the transceiver, performs a logical AND operation, and then distributes the signals back to all nodes. It is also possible to link nodes to the ASIC star coupler electrically. The star coupler offers the error-handling and optical diagnosis possibilities that were described earlier. It is located in a 64-pin QFP package and makes it possible to link together 22 nodes, to read out diagnosis information via an SPI interface, to turn off the connected nodes individually (prevention of a bus blockade caused by a "babbling idiot"), and to regenerate the signal pulse form.

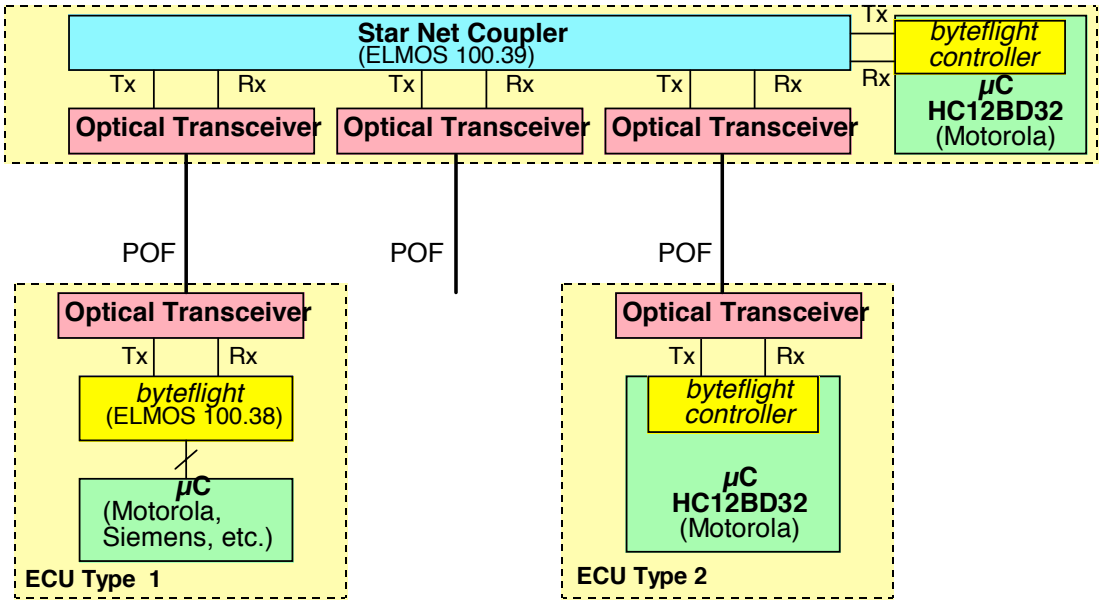

Figure 6 - Example of a *byteflight* topology

As an alternative, the bus protocol permits the electric linking of all nodes (when using suitable physical drivers) to form a bus topology. To accomplish this, the transmission rate must be lowered, since there are currently no suitable electrical transceivers available for automotive applications that permit a transmission rate of 10 Mbps.

5 DEVELOPMENT TOOLS AND TEST METHODS

5.1 *byteflight* DEVELOPMENT TOOLS

There are many development tools available. For example, there is a bus analysis device that permits online analysis, recording (with off-line analysis afterwards), and simulation the bus traffic. All the analysis device's functions can be controlled with the aid of a user-programmable DLL interface; this makes it possible for the device to be integrated into the developer's own simulation and test environments. The device is connected to the *byteflight* network optically, or – for laboratory tests – via an electrical interface. Beyond that, the IXXAT company is currently developing a PCMCIA card with *byteflight* and two CAN interfaces, and a *byteflight* analyzer tool based on the IXXAT CAN -Analyzer.

5.2 *byteflight* CONFORMANCE TEST

In order to guarantee the compatibility of the two *byteflight* implementations, Motorola HC12BD32 and ELMOS E100.38, a *byteflight* conformance test was developed together with Steinbeis Transferzentrum Prozessautomatisierung. This test compares the implemented protocol in each case with the specification. Both *byteflight* implementations have already been successfully in use together in test vehicles for some time now.

6 SOFTWARE STRUCTURE

The clock synchronization of all bus nodes is very important for the deterministic behavior of distributed safety-related and closed loop control systems. Using the synchronization pulses, which are transmitted at 250-μs intervals, the *byteflight* bus protocol makes a time base (accurate to 100 ns) available to the applications. With this signal, both the hardware and software can be synchronized at all network nodes involved – independently of the software. The sync pulse is used to initiate the synchronous (high priority) software routines; then the asynchronous (low-priority) routines are processed in the remaining cycle time. In the synchronous portion, no interrupts are permitted, but they are permitted in the asynchronous portion of the software.

In order to support fast processing of the high-priority routines, the high-priority messages are received in the dedicated receive buffers, while the low-priority messages are received in the FIFO mode. The dedicated receive buffers always contain the most up-to-date data received, and a bit in a control register indicates the arrival of a new message.

Since the application software knows the identifier and the length of the high-priority messages, the point of time when the message arrives can also be determined with great accuracy. When a message is not transmitted (for instance, when a node fails), only the short time period of a waiting slot passes before the next message can be transmited, instead of the longer time period of a transmission slot. Consequently, the messages that follow are sent earlier than they would be if the system did not have a fault. This results in reception jitter of 5 to 17 μs, depending on message lengths. Since the application software knows how "old" the data is (the data has to be entered into the transmission register before the synchronization pulse) this avoids any disadvantage in the functioning of control loops or the time behavior for safety-related applications. Thanks to the protocol's deterministic behavior for high priority messages (which can be transmitted in every cycle) and to the fact that the reception jitter caused by the failure of a message is insignificant, the system has a good composability in the

time domain. The time behavior at the communication level is independent from the time behavior of the application software.

7 FLEXIBILITY

byteflight permits simple expansion of a system by adding further functions to control units that are already in place, or by incorporating more bus nodes (for example, optional equipment). The system can be supplemented by new messages without the software of existing control units having to be adapted. The system's expansion characteristics allow a high level of flexibility in production (different equipment variations with the same sub-assemblies). They also make it possible to use the same control units in different model series and with different configurations of the vehicle electrical system.

8 GATEWAY: NETWORKING WITH OTHER DATA BUSES

The growing complexity of future motor vehicles and the constantly rising demands on the performance of the vehicle communication system have led to a situation in which networking of the control units can no longer be accomplished with a single data bus. In order to limit the bus load and adapt the performance capabilities of a data transmission to the given requirements, it makes sense to consolidate certain functional areas to form sub-systems and to join the sub-systems together with a data bus to form a network. The need to exchange vehicle-specific data between the individual sub-systems or to ensure the overall diagnosis of all control units in all sub-systems using a single diagnostics interface, makes it necessary to connect the buses using a gateway. For system integration it is very important, that the *byteflight* protocol supports all services of the internationally standardized KWP2000 (Keyword Protocol 2000).

Such a gateway function for two CAN buses with different transmission rates, one diagnostics line, and *byteflight* has already been realized with the aid of the E100.38 standalone controller and a high-performance microcontroller, and is already working in test vehicles.

9 EXAMPLES OF APPLICATIONS

Due to its deterministic behavior and high data rate, the *byteflight* protocol is suitable for applications in the area of passive safety, and thanks to the high level of flexibility, it is suited for body and convenience functions.

The protocol can also be used in active safety applications. In order to reach the fault-tolerance levels required by such applications, the corresponding redundancies must be planned into the system configuration. Variations that can be used to accomplish this include a doubling of the network so that each node has two controllers (such as an HC12BD32 and a standalone E100.38), two transceiver modules, two fiber-optic interfaces and two star couplers available (Figure 7).

The transmission rate of 10 Mbps and high level of immunity to electromagnetic interference made possible by optical transmission, offer tremendous advantages for the use of the *byteflight* data bus system in industrial automation and in the aerospace industry. Due to the large production volumes that occur in the automobile industry, this system also features considerable cost advantages over current technologies.

10 CONCLUSIONS

The deterministic behavior of the *byteflight* protocol, flexible use of bandwidth, expandability of the system, high data transmission rates and data integrity, high level of communication interference immunity made possible by fiber-optic transmission, the possibility of download via the data bus, and the capability of online diagnosis pre-destine this protocol to become a transmission protocol standard in the automotive arena (Table1).

The first use of *byteflight* in mass production will take place at BMW within the next year in a networked passive safety system, into which body and convenience equipment functions have also been integrated.

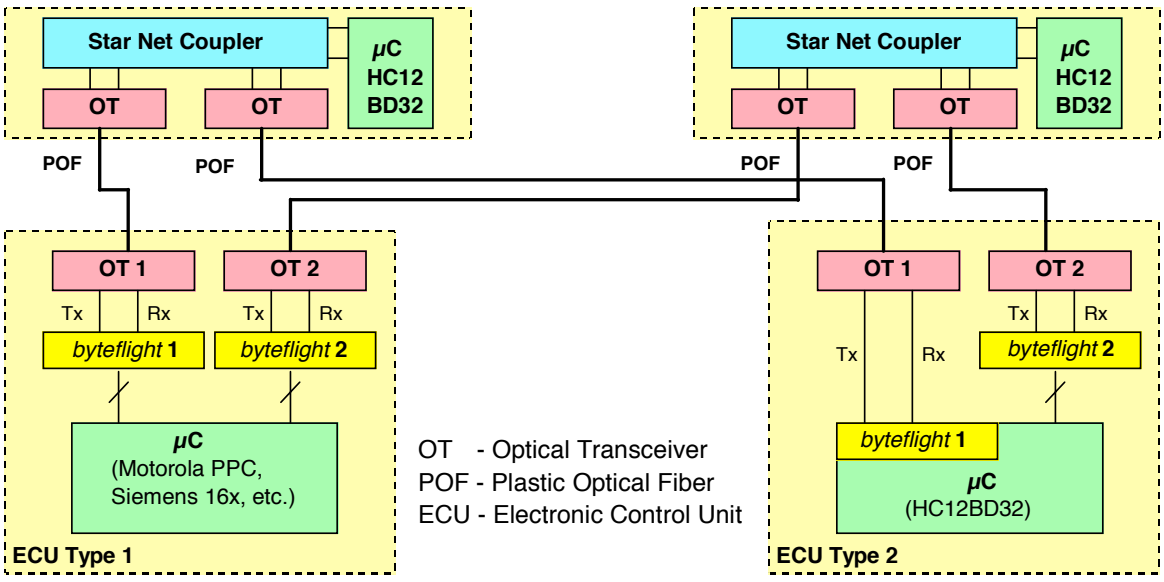

Figure 7 - Redundancy concept

Table 1 - Comparison of CAN / TTP / *byteflight*

Feature	**CAN**	**TTP** (10)	***byteflight***
Message transmission	asynchronous	synchronous	asynchronous and synchronous
Message identification	message identifier	time slot	message identifier
Data rate	1 Mbps gross up to 58 % net	2 Mbps gross up to 37 % net	10 Mbps gross up to 53 % net
Bit encoding	NRZ with bit stuffing	modified frequency modulation (MFM)	NRZ with start/stop bits
Physical layer	transceivers up to 1 Mbps	not defined	optical transceiver up to 10 Mbps
Latency jitter	bus load dependent	constant for all messages	constant for high priority messages according t_cyc
Clock synchronization	not provided	distributed, in μs range	by master, in 100 ns range
Temporal composability	not supported	supported	supported for high priority messages
Error containment (physical layer)	partially provided	provided with special physical transceiver	provided by optical fiber and transceiver chip
Babbling idiot avoidance	not provided	possible by independent bus guardian	provided via star coupler
Extensibility	excellent	only if extension planned in original design	extension possible for high priority messages with affect on asynchronous bandwidth
Flexibility	flexible bandwidth for each node	only one message per node and TDMA cycle	flexible bandwidth for each node
Availability of components	several μC families and transceiver chips	microcoded RISC chip available, physical transceiver and independent bus guardian not available	HC12BD32, E100.38 *byteflight* standalone controller, E100.39 star coupler ASIC, optical transceiver available

REFERENCES

[1] R. Griessbach, J. Berwanger, M. Peller: *byteflight* - neues Hochleistungs-Datenbussystem für sicherheitsrelevante Anwendungen, ATZ/MTZ „Automotive Electronics“, January 2000, Friedrich Vieweg & Sohn Verlagsgesellschaft mbH.

[2] BMW AG, EE-211, *byteflight* Specification.

[3] BMW AG, EE-211, Workshop SI Bus Publication, Munich, May 6th 1999.

[4] Motorola, 68HC912BD32TS/D Technical Summary.

[5] ELMOS AG, Product Specification E100.38.

[6] Infineon/ELMOS, Specification for SPFOBI 002 Transceiver.

[7] ELMOS AG, Product Specification E100.34.

[8] O. Schoenfeld, Transceivers for in-car optical buses, POF Conference 1999, Tokyo.

[9] ELMOS AG, Product Specification E100.39.

[10] Handouts, TTA/X-By-Wire Workshop, Vienna, December, 3rd 1998.

APPENDIX

More details about the *byteflight* protocol, the protocol specification, the controllers and the optical transceiver are available via:

http://www.byteflight.com