FlexRay®

Requirements Specification

Authors:

Ralf Belschner[2], Josef Berwanger[1], Christian Ebner[1], Harald Eisele[4], Sven Fluhrer[2], Thomas Forest[4], Thomas Führer[3], Florian Hartwich[3], Bernd Hedenetz[2], Robert Hugel[3], Andreas Knapp[2], Josef Krammer[1], Arnold Millsap[4], Bernd Müller[3], Martin Peller[1], Anton Schedl[1]

Version 2.0.2

9th of April, 2002

Table of Contents

1 Introduction

This document contains a requirement specification and high-level system description for a dependable automotive network. The low-level (implementation) specifications must adhere to this document and should be written by the suppliers of the communication controller and physical layer.

Throughout the document the following structure is followed; at the beginning of each chapter (section) the requirements for this topic are defined, afterwards more detailed descriptions are provided. In the document UPPER CASE letters are used to denote constants. For a summary of all constants used in the document please refer to Chapter 7.

1.1 Objectives

The objectives pursued in the development of the dependable automotive network are the following:

- Support of two communication paradigms, deterministic (statically defined) communication and dynamic event driven communication.
- Configurable static and dynamic part within one communication cycle. Fully static and fully dynamic configuration has to be supported.
- Flexible extendibility, even after deployment.
- High data rate and bandwidth efficiency.
- Scalable fault tolerance (i.e., single channel and dual channel operation must be supported).
- Reliable error detection (bus guardian[1] mechanism in the time domain, CRC in the value domain).
- Support of electrical and optical physical interfaces.
- Enable very low system level failure in time ratings
- Allow use of crystal oscillators and low tolerance ceramic resonators
- Support of active star and bus topologies
- Low overall system cost
- Enable re-use of carry over components without embedding knowledge of future platform partitioning.

[1] See glossary.

Objectives of the static segment:

- Deterministic communication behaviour in the time domain.
- Global time implemented by a fault tolerant clock synchronisation algorithm.
- Immunity against accepting error-free sub-sequences of a message as valid messages (i.e. short message rejection).

Objectives of the dynamic segment:

- Event driven dynamic communication possibility.
- Flexible bandwidth allocation (for different nodes during runtime).
- No interference with the static segment.
- Support for prioritised bus access.
- Support of variable length messages with at least 200 data bytes.

Global Requirements:

- Support for fault tolerance, but operation without fault tolerance must also be possible, i.e., a single bus (channel) connection must be possible for non-fault-tolerant nodes.
- The communication network has to support a system architecture, where no single fault may lead to a functional degradation.
- Protection against faults in accordance with a well-defined fault hypotheses.
- Protect against up to and including five random bit errors per frame.
- The communication protocol should be independent as far as possible from the topology.
- For highly dependable and fault-tolerant applications an independent bus guardian to prevent the monopolisation of the communication medium by a communication controller is required.
- Errors in hardware and configuration data have to be detected during initialisation and operation by error detection mechanisms (EDMs). In case a critical error is detected the controller and transceiver must not be allowed to enter normal operation or immediately abort normal operation and report an error to the host.
- Support of serviceability of system- and component-level faults.
- The bit encoding technique must not introduce data dependent changes in the length of the resulting bit stream, e.g., bit stuffing is not allowed.
- Automotive qualification of the communication controller, bus guardian, and the physical layer is required.
- Configuration data must be readable/writeable by the host. It must be possible to prohibit writing during run-time.

- Support of comprehensive self test at system communication startup.
- Support of timely and highly reliable component re-integration and system-level startup.
- Support of master-less system startup and shutdown.
- Support of traceability of system- and component-level faults to identify root causes of failures.
- Support of synchronized system shutdown without error indications.
- Support of synchronous distributed application startup and shutdown with acceptable timing and fault tolerance characteristics.
- Support of node and network moding with high security against critical inadvertant mode changes.
- Logical line compatibility to the byteflight protocol[2], when using an optical physical layer must be possible.

[2] See byteflight specification at http://www.byteflight.com

2 Basic Concepts

The communication protocol for the dependable automotive network described in this document has the following properties:

- Synchronous and asynchronous frame transfer.
- Multi-master clock synchronisation.
- Guaranteed frame latency times and jitter during synchronous transfer.
- Prioritisation of frames during asynchronous transfer.
- Error detection and signalling.
- Error containment on the physical layer through an independent bus guardian device.
- Scalable fault tolerance, e.g., one controller, one/two channels, one bus guardian for each channel.

The layered architecture of the FlexRay protocol is shown in Figure 1.

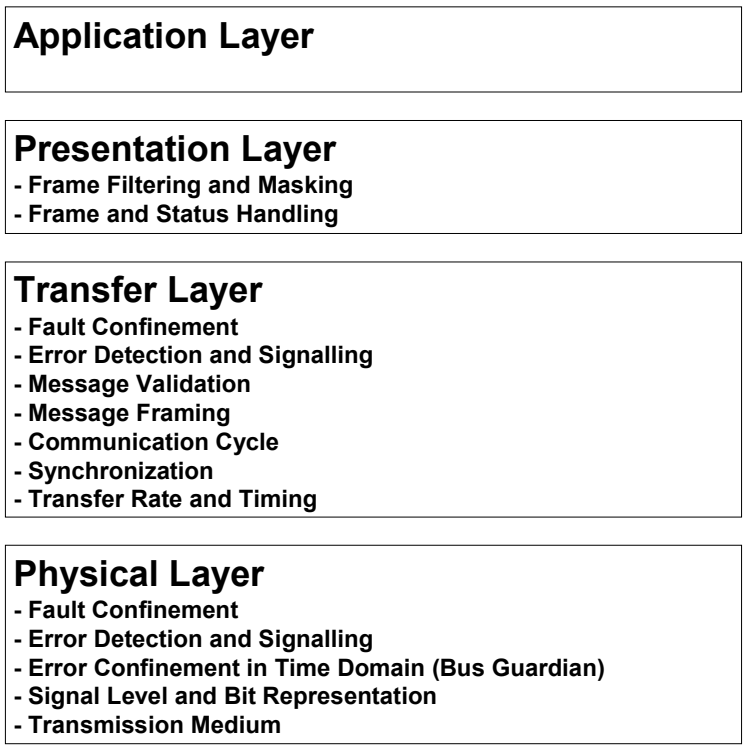

Figure 1: Layered structure of a communication node.

- The Physical Layer defines how signals are actually transmitted. One task of the Physical Layer is to detect errors of the communication

controller in the time domain. This is done by the so-called Bus Guardian (Chapter 4.8).

- The Transfer Layer represents the kernel of the protocol. It presents frames received to the presentation layer and accepts frames to be transmitted from the presentation layer. The transfer layer is responsible for timing, synchronisation, message framing, error detection and signalling, and fault confinement (Chapter 3).

- The Presentation Layer is concerned with frame filtering and masking, frame status handling and contains the communication controller host interface (Chapter 5.1).

- The Application Layer is not part of this specification.

2.1 Node (ECU) Architecture

Figure 2 shows the architecture of a node (ECU). Every node consists of the five sub-components host, communication controller, bus guardian, bus driver, and power supply. This specification describes the requirements for the communication controller, the bus guardian, the bus driver and the interfaces to the host and the power supply.

Two implementations for the communication controller are possible, one configuration of a communication controller that sends and receives on two redundant physical channels, and a second configuration which is solely connected to one physical channel.

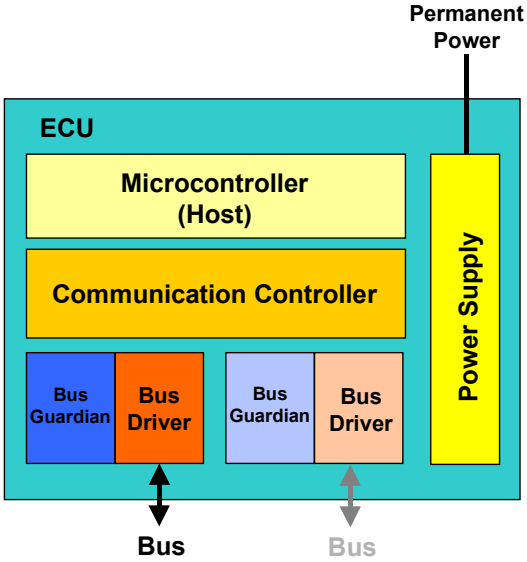

Figure 2: Architecture of a node (ECU).

2.2 Topology

Figure 3 shows the possible topology configuration of the communication network. A node can either be connected to both channels 1 and 2 (node A, C, and E) or only channel 1 (node B) or only channel 2 (node D). A configuration, where all nodes are connected by 1 channel only is also possible.

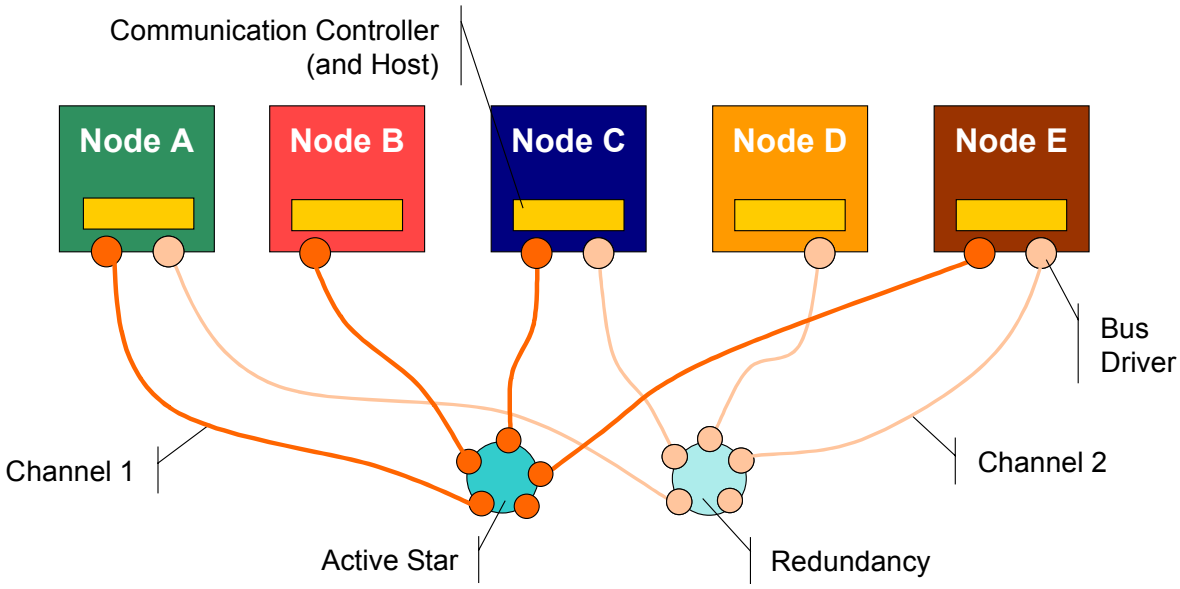

Figure 3: Possible network configurations.

2.3 Frame Transfer

Communication is done in a communication cycle consisting of a static and a dynamic segment, where each of the segments may be empty. The first frame ID in a system with a static segment is ID number 1 (see Figure 4). In a pure dynamic system with the SOC symbol (see Figure 5). The sending slots are represented through the ID numbers that are the same on both channels.

The sending slots are used deterministically (in a pre-defined TDMA strategy) in the static segment. In the dynamic segment there can be differences in the phase on the two channels (see Figure 4). Nodes that are connected to both channels send their frames in the static segment simultaneously on both channels. Two nodes, that are connected to one channel only, but not the same channel, may share a slot in the static segment.

To guarantee the consistency of the clock synchronisation only nodes can participate that send frames, which are received by all other nodes (e.g., node A, C and E in Figure 3). All nodes execute the clock synchronisation algorithm, but only the frames of the static segment are considered. It is possible to send different data in the same sending slot on different channels.

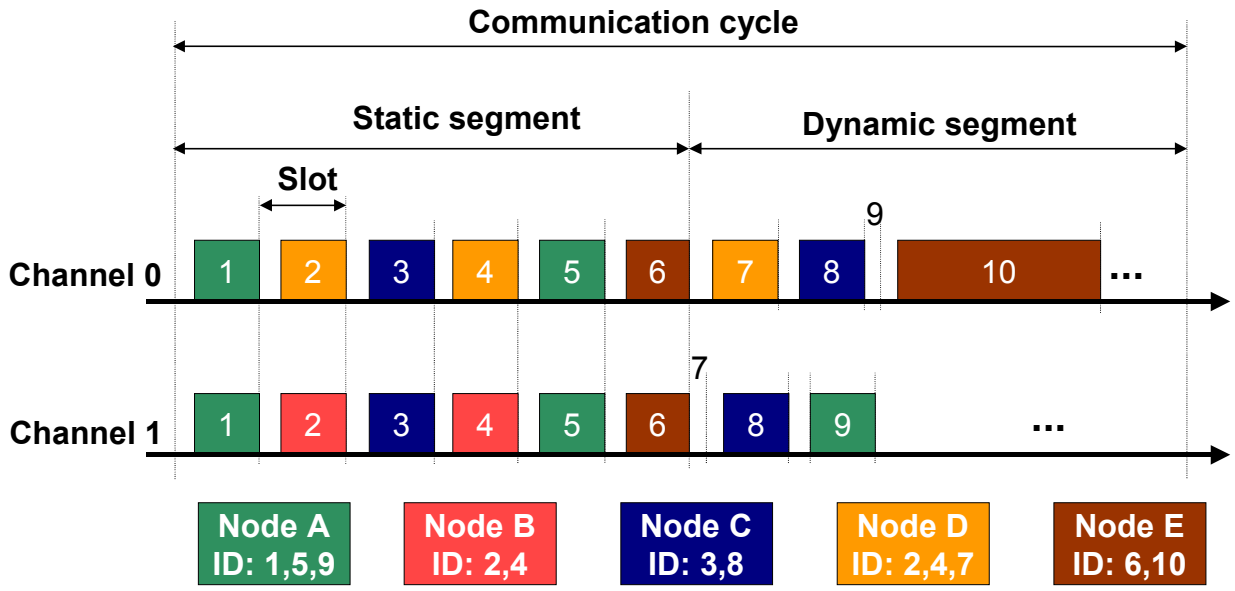

Figure 4: Definition of a communication cycle with static segment.

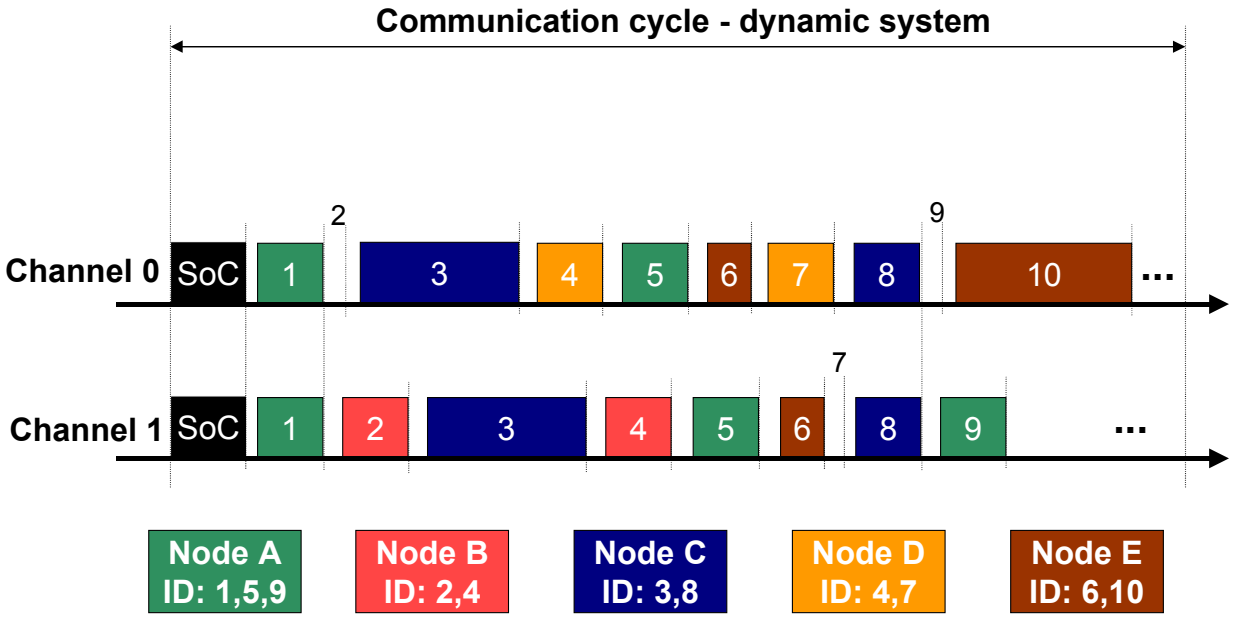

Figure 5: Definition of a communication cycle in a pure dynamic system.

2.4 Constraints

- The communication controller must allow to interface to an optical or an electrical physical layer.
- The communication controller must support a net data rate of at least 5 Mbit/s real application data transfered per seconds under the

constraints of frame overhead (including CRC) and protocol timing overhead (IFG) in static communication mode.

- It must be possible to connect 2 up to CONTROLLER_MAX controllers to one communication channel.
- The maximum number of slots in the static segment is set to STATIC_SLOTS_MAX.
- The power supply for the bus driver (including the bus guardian) and the communication controller must meet automotive requirements.

2.5 Comment

Considering the FlexRay protocol as described in the previous sections the communication scheme of networked FlexRay nodes can be briefly characterised as follows:

- each node must be able to make use of the distributed clock
- each node must send frames inside a predefined static slot
  or/and
  inside the dynamic segment (collision free access)

the transmission of frames must be subdivided into 3 phases
1st a bus guardian must enable the access to the bus
2nd it must be signaled that a frame should be transmitted
3rd the transmission of the frame itself

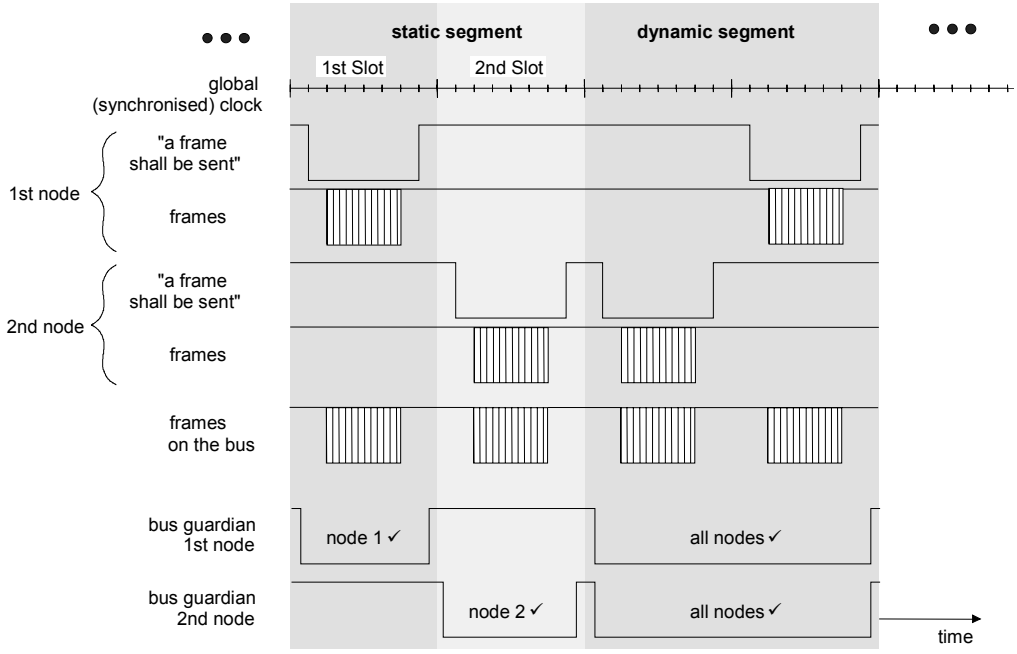

Figure 6: Typical communication scheme of two FlexRay nodes.

3 Protocol Description

Throughout the document the following notation is used:

Req: Requirements.

Comment: *Contains additional descriptions and explanations.*

3.1 General Requirements

⇨ Req: The communication protocol shall be independent from the data rate.

Comment:

It shall be possible to implement low end controllers e.g., 500 Kbit/s and high end controllers beyond 100 Mbit/s.

⇨ Req: The first communication controller must support a net data rate of at least 5 Mbit/s.

Comment:

Net data rate: Real application data transfered per seconds under the constraints of frame overhead (including CRC) and protocol timing overhead (IFG) in static communication mode.

⇨ Req: A CRC code with a Hamming Distance of at least 6 must be used.

⇨ Req: The communication controller shall be able to operate in a current byteflight environment, i.e., the two protocol controllers have to support the same physical interface and the same representation at the logical line level. The byteflight compatibility is required for the optical physical layer only.

Comment:

Compatibility of the interfaces between host CPU and the protocol controller (CHI) is not required. The electrical physical layer does not need to support byteflight compatibility. The byteflight specification can be downloaded from the following web address: www.byteflight.com.

3.2 Frame Transfer

Data transfer in FlexRay is done in cycles, called communication cycles.

⇨ Req: The communication cycle consists of a static and a dynamic segment as shown in Figure 4. Each of the segments may be empty, that means there are three possible configurations of the communication cycle (pure static, mixed static and dynamic (a mixed system consists of at least two static slots) and pure dynamic).

In a pure dynamic system the communication cycle starts with an SOC symbol. There are two different SOC symbols (alarm condition, normal

condition). The sending slots are represented through the identifiers that are the same on both channels (see Figure 4).

The sending slots are used deterministically (in a pre-defined TDMA strategy) in the static segment. In the dynamic segment there can be differences in the phase on the two channels (see Figure 4). Nodes that are connected to both channels send their frames in the static segment simultaneously on both channels. A node that is connected to only one channel may share an identifier with another node that is only connected to the other channel.

The current communication cycle is determined by a cycle counter that is consistently incremented in every cycle (see Figure 7).

⇨ Req: The length of the communication cycle has to be stored in the configuration data.

Fit Criteria:

The maximum cycle length is defined by CYCLE_LENGTH_MAX.

⇨ Req: A check mechanism has to be designed that ensures that no frame transmission is started within a certain interval so called forbidden region before the end of the communication cycle, to ensure that the beginning of the static segment in the next communication cycle is not disturbed.

⇨ Req: Multiplexing of sending slots of one controller must be supported in such a way, that the contents of frames can be multiplexed for a certain sending slot in different communication cycles.

Comment:

So a communication matrix with nearly any possible communication patterns (periods of certain frames) based on the principle of communication cycles can be built up.

3.2.1 Static Segment

⇨ Req: If the static segment of the communication cycle is not empty it consists of STATIC_SLOTS_MIN ≤ NUMBER_OF_SLOTS ≤ STATIC_SLOTS_MAX.

⇨ Req: The static segment is subdivided into a sequence of time slots. In each of these static slots only one controller may send one frame on each channel.

Comment:

In the static segment of the communication cycle a TDMA media access strategy is used.

⇨ Req: There is one configuration parameter for the slot length (slot_length) in the static segment, that defines this value.. The length of the slots is configurable off-line but fixed during runtime.

3.2.2 Dynamic Segment

⇨ Req: In a pure dynamic system the communication cycle starts with the SOC symbol.

⇨ Req: The dynamic segment of the communication cycle consists of zero or more dynamic identifiers (slots) within the communication cycle.

⇨ Req: Bus access in the dynamic segment is done via static frame priorities according to the byteflight specification.

Comment:

The byteflight specification can be downloaded from the following web address: www.byteflight.com.

⇨ Req: In the dynamic segment the media access strategy is based on wait times (mini-slotting scheme) and the priority of identifiers. Controllers transmitting frames with higher priority identifiers send before controllers transmitting lower priority frames.

⇨ Req: The frame length in the dynamic segment is variable during runtime.

⇨ Req: In pure dynamic mode an external triggered SOC generation and with it the start of the communication cycle has to be supported. The timing behaviour of the external trigger has to be monitored by the communication controller.

3.3 Frame Format FlexRay

⇨ Req: Two frame formats as specified below must be supported.

Comment:

The mixture of the two frame formats need not be supported, i.e., all nodes connected to a FlexRay communication system can be configured using only the FlexRay format or the byteflight format.

3.3.1 FlexRay Frame Format

⇨ Req: It must be possible to use the FlexRay format in a pure static, in a combined static and dynamic, and in a pure dynamic configuration.

⇨ Req: The first section of the data field in a frame must be configurable as a message ID field. This data field must be filterable by the receiver.

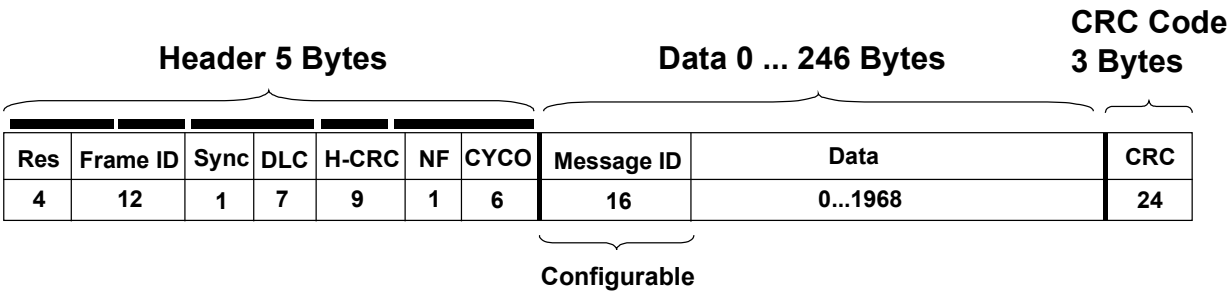

Figure 7: FlexRay frame format.

Res: Reserved bits, 4 bit, for future protocol extensions.

ID: Identifier, 12 bit, value domain: (1_{10} ... 4095_{10}), defines the slot position in the static segment and defines the priority in the dynamic segment, see Figure 4. A lower identifier determines a higher priority. The identifier of a frame must be unique within a cluster. Each controller can have one or more identifiers (in the static and the dynamic segment).

SYNC: Synchronisation field, 1 bit, indicates that the slot is used for clock synchronisation.

DLC: Data length code field, 7 bit, DLC * 2 = number of data bytes (0_{10}, 2_{10},..., 246_{10}).

H-CRC: 9 Bit Cyclic Redundancy Check – Sequence. The H-CRC is calculated over the SYNC- and DLC-field.

NF: Null frame indication field, 1 bit, indicates that the corresponding data buffer is not updated by the host before sending.

CYCO: Cycle Counter, 6 bit, the cycle counter is increased simultaneously in all nodes by the controller at the start of each new communication cycle.

Message ID: The Message ID field is configurable to be used as the message identifier or as the first two data bytes.

D0 ... D246: Data bytes, 0-246 bytes.

CRC: 24 Bit Cyclic Redundancy Check – Sequence. The CRC is calculated over the complete frame.

3.3.2 *byteflight* Frame Format

⇨ Req: The byteflight frame format must be supported for pure dynamic configurations.

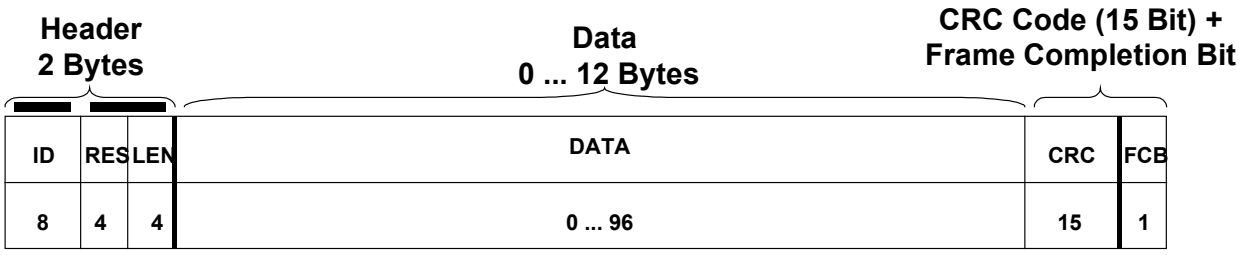

Figure 8: *byteflight* frame format.

ID: Identifier, 8 bit, value domain: (1_{10} ... 255_{10}), defines the priority in the dynamic segment, see Figure 8. A lower identifier determines a higher priority. The identifier of a frame must be unique within a cluster. Each controller can have one or more identifiers.

Res: Reserved bits, 4 bits, for future protocol extensions.

LEN: Length field, 4 bit, LEN = number of data bytes (0_{10} ... 12_{10}), a value higher than 12 is handled as LEN=12.

D0 ... D11: Data bytes, 0-12 Bytes

CRC: 15 Bit Cyclic Redundancy Check – Sequence ($x^{15} + x^{14} + x^{10} + x^8 + x^7 + x^4 + x^3 + 1$).

FCB: Fill completion bit: an additional bit is added to the 15 bit CRC to fill the complete word. The bit is set to "0" as LSB.

3.4 Frame Scheduling – Multiplexing of Sending Slots

⇨ Req: The cycle counter can be used to distinguish between different frame contents.

Comment:

For a sending slot different send and receive buffers can be defined in different cycles (slot multiplexing).

Comment:

The cycle counter can be used as a logical extension to the identifier (in the case of multiplexing).

3.4.1 Frame and Bit Coding

⇨ Req: The coding algorithm in the communication controller has to be robust against:

- glitches

3.4.1.1 Optical Physical Layer

⇨ Req: The controller must support at least the byteflight optical bit encoding.

Comment:

In byteflight, frames on the communication media are composed of individual

bytes consisting of a start bit, eight data bits and a stop bit.

In addition, transmission of each frame begins with a start sequence consisting of 6 logical "0"bits. This is illustrated by the diagram in Figure 9.

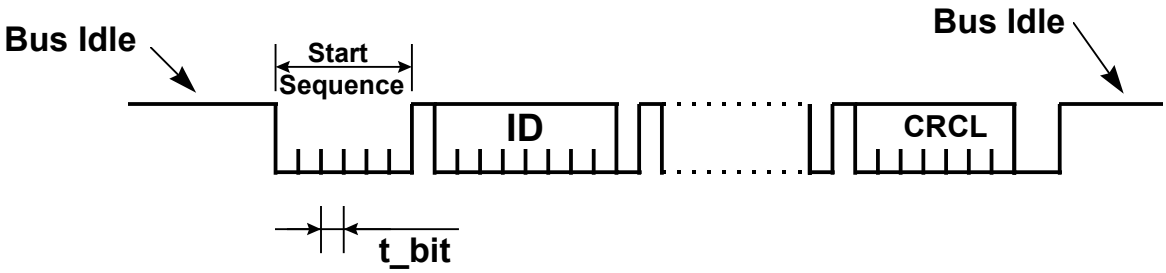

Figure 9: Transmission of the start sequence.

Due to certain effects in the optical transmission, it is possible for the start sequence to be become shorter or longer during optical transmission. This is why the receiver accepts start sequences in the region of 1-9 logical "0" bits. This is illustrated by the following diagram in Figure 10.

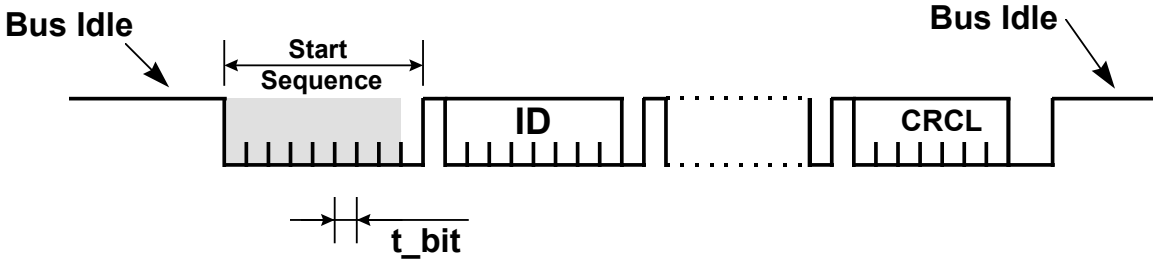

Figure 10: Reception of the start sequence.

3.4.1.2 Electrical Physical Layer

⇨ Req: Frames on the communication media for the electrical physical layer are composed as shown in Figure 11. The frame end sequence may be empty.

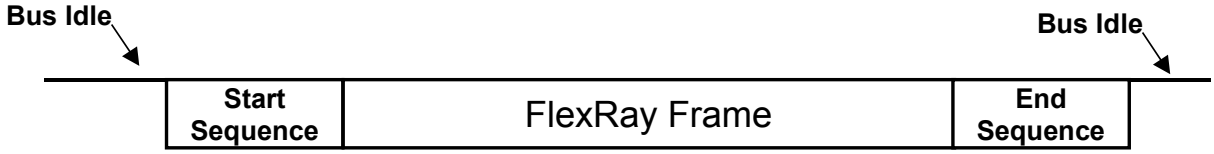

Figure 11: Frame format for electrical transmission.

⇨ Req: A suitable bit coding schema has to be selected, in accordance to bandwidth efficiency and EMC requirements.

3.5 Frame Timing

⇨ Req: Frame timing of different communication controller implementations must be interoperable.

3.5.1 Frame Timing for Static Segment

⇨ Req: The reception start window shall be defined in relation to the precision. I.e., the start window must be greater than the precision.

Comment:

The actual value for the reception start window must be defined in the implementation specification.

⇨ Req: In the static segment accurate timing requirements have to be ensured for correct frame reception. Frames may only start within the reception start window.

Comment:

The length of the frame determines the frame duration. Correct reception is given if the respective frame does not violate the temporal borders given by the access scheme. The judgement of temporal correctness is based on a rigid timing scheme.

⇨ Req: The time difference between the predicted start StartNom and the observed start of the frame (SOF) is used by the clock synchronisation algorithm.

⇨ Req: The length of the IFG has to be minimised, in order to optimise the net communication rate.

3.5.2 Frame Timing for Dynamic Segment

⇨ Req: The frame timing in the dynamic segment must be defined due to the *byteflight* specification.

3.6 Start-up

3.6.1 Requirements

⇨ Req: For each configuration (Pure Static Systems, Pure Dynamic and Mixed System) the start-up of the communication network has to be possible as soon as two nodes are able to communicate.

⇨ Req: The integration of controllers that are powered on later must not disturb the start-up procedure of the other nodes.

⇨ Req: The start-up and re-integration of controllers shall not disturb the normal operation of the network.

⇨ Req: The worst-case start-up time under the fault conditions given above has to be provided and guaranteed by the supplier of the communication controller. The communication network must be operational after 100 ms.

Comment:

The application designer has to consider the start-up time during the

determination of the configuration parameters. Typical automotive applications require a worst case startup time of 100ms. For system configurations with extremely long communication cycles longer startup times are acceptable.

⇨ Req: During start-up a communication controller sets the cycle counter according to the value in the received frame. The cycle time is derived from the frame ID and set accordingly.

⇨ Req: The startup must work without reliance on collision detection.

Comment:

Collisions can occur on the bus during start-up and in the case of faults. In star topologies collision detection is not always feasible.

3.6.2 Principle of Operation – Protocol Modes

3.6.2.1 Pure Static System and Mixed System

⇨ Req: The start-up has to work as a distributed algorithm.

⇨ Req: Only controllers that participate in clock synchronisation (sync bit set) are allowed to start up the system.

Comment:

In a dual channel system only controllers connected to both channels are allowed to execute the start-up in a heterogeneous topology (mixing controllers with single channel and controllers with dual channel).

Controllers connected to only one channel are not allowed to start-up the bus because they may corrupt the traffic of this channel in case of an incoming-link failure by sending frames after the listen-timeout (listen_timeout).

⇨ Req: The start-up of the communication network, the integration of nodes powered on later and the re-integration of failed nodes must be fault-tolerant against:

- the temporary/permanent failure of one or more communication controllers (down to one controller sending in the static segment for mixed or pure static configurations),
- the temporary/permanent failure of one or more communication channel(s) in a redundant configuration, and
- the loss of one or more frames.

3.6.2.2 Pure Dynamic System

⇨ Req: A single master sends the SOC symbol. The master shall be defined at design time.

3.7 Shutdown

⇨ Req: The co-ordinated shutdown of the FlexRay cluster, including all nodes and all stars initiated by the application must be possible. The interference with the wake-up mechanism must be handled.

⇨ Req: The communication system has to support a synchronized system shutdown without error indications.

3.8 Clock Synchronisation

Comment:

The proper synchronisation of the individual clocks of the communication controllers is a pre-requisite for the TDMA communication scheme.

This chapter contains the description of the FlexRay clock synchronisation mechanism (based on the Fault-Tolerant Midpoint algorithm).

3.8.1 Pure dynamic system

⇨ Req: In a pure dynamic operation the clock synchronisation is performed by a master (SOC).

3.8.2 Pure Static and Mixed System

⇨ Req: The global time is a vector of two values. Global time = < cycle_counter, cycle_time >.

⇨ Req: The cycle time is a counter incremented in units of macroticks. The cycle time is reset to 0 at the beginning of each communication cycle.

⇨ Req: The macrotick defines the resolution of the global time within one cluster. A resolution of 1µs must be achievable in realistic configurations.

⇨ Req: The macrotick shall be independent of oscillator frequency.

Comment:

In the implementation each (local) macrotick is an integer multiple of the (local) clocktick, i.e. depends on the oscillator frequency, but the factors of two different macroticks can be different, so over a cycle, independence can be achieved.

⇨ Req: The microtick defines the accuracy of the clock difference measurement. A resolution of <=50 ns is required for the microtick.

Comment:

Typical automotive applications require a resolution of 50 ns. For system configurations with low bandwidth, higher values are acceptable.

⇨ Req: The clock synchronisation mechanism must be able to keep all fault-free controllers within the precision. A clock synchronisation precision within the different controllers of better than 1 microsecond is required.

Comment:

Typical automotive applications require a precision of 1 microsecond. For system configurations with lower precision requirements, greater values are acceptable.

⇨ Req: The absolute value of the global time must be the same at every controller, within the limits defined by the precision. During start-up the first sending node determines the value of the global time.

⇨ Req: The fault tolerance of the clock synchronisation mechanism must be scalable with the number of controllers. The level of fault tolerance depends on the number of actual nodes in the system (3k+1 to tolerate k asymmetric faults). . In a reduced fault-tolerant configuration of less than four controllers (e.g., 2 or 3, in a degraded mode of operation) the synchronisation must be possible. For 4 to 6 controllers the clock synchronisation mechanism must be fault-tolerant against 1 (asymmetric) fault. For 7 or more controllers the clock synchronisation mechanism must be fault-tolerant against 2 (asymmetric) faults.

⇨ Req: The clock synchronisation mechanism must prevent the formation of cliques with different notions of time within the network.

⇨ Req: The clock synchronisation mechanism must be able to operate with oscillators that have automotive quality. In particular the clock synchronisation mechanism must be able to deal with the physical phenomena (drift, deterioration) that can occur during an automobile lifetime.

⇨ Req: A subset of controllers must be configured to send sync-frames (a frame with a set SYNC bit). In a dual channel system only controllers connected to both channels may belong to this subset.

⇨ Req: Only one static sending slot of each controller is allowed to contribute to the clock synchronisation mechanism, i.e., a controller may send at most one sync-frame per communication cycle.

⇨ Req: Only correctly received sync-frames are used for clock synchronisation.

⇨ Req: Every node has to use all available sync-frames for clock synchronisation.

⇨ Req: The clock synchronisation and the implementation of the clock synchronisation shall be as resilient as possible against design violations resulting from environment or possible misuse.

3.8.3 Principle of Operation

3.8.3.1 Obtaining the time values

⇨ Req: In the static segment every node measures the time difference between the actual receive time and the expected receive time for the sync-frames with a resolution of a microtick.

⇨ Req: This time difference measurement is done for all channels.

⇨ Req: An SOF-window is placed around the expected receive time of an SOF. The length of the receive-window is equal to the length of the SOF-window.

⇨ Req: Time values are obtained for correct frames only.

Comment:

Note that one of the reasons why a frame is considered incorrect is reception outside the receive window. The stringent application of the receive-window mechanism ensures that synchronisation errors of nodes are detectable.

3.8.3.2 Measurement method

⇨ Req: The measurement of the clock deviations is done through measuring the differences between the expected arrival time and the actual arrival time . The expected arrival time of a frame is defined by the internal view of the cycle time.

3.8.3.3 Synchronisation algorithm

⇨ Req: The synchronisation algorithm uses a fault-tolerant midpoint algorithm (FTM) that operates with an arbitrary number of controllers.

Comment, Description of the FTM:

The measured values are sorted and the k *largest and smallest values are discarded.* k *is adapted dynamically so that at least 2 measured values are remaining.*

The largest and the smallest of the remaining values are selected for the calculation of the midpoint value, i.e., average of those two values. The resulting value in a node describes the deviation of the own clock from the global time base and serves as a correction term.

⇨ Req: The resulting correction term(s) shall be used for error detection of the communication controller. If the correction term cannot be applied, an error has to be signalled to the host.

⇨ Req: The clock correction term(s) calculated in the previous step shall be applied to the local clock.

⇨ Req: If equal or more than half of the frames are received outside the reception start window a synchronisation error is indicated to the host. The synchronisation error is a fatal error the controller has to reintegrate.

Comment:

This mechanism prevents the formation of cliques.

3.8.4 External Synchronisation

⇨ Req: External synchronisation must be supported.

Comment:

External synchronisation is necessary for the synchronisation of a FlexRay network to an external time reference, e.g., a GPS receiver or a DCF77 receiver, or to synchronise several FlexRay networks.

⇨ Req: Each controller can add an additional external clock correction term(s) to the calculated clock correction term(s).

⇨ Req: The resulting clock correction term(s) shall not be greaterthan the maximum allowed correction term(s) or smaller than the minimum allowed term(s). The communication controller shall limit the applied correction term(s) to allowed values.

⇨ Req: The host shall be able to read the current (local) correction term(s) (current clock value) and set the external correction term(s).

⇨ Req: A hardware input signal at the communication controller for external synchronisation is required.

⇨ Req: The hardware input signal shall be connected to the internal soft-reset of the controller. It shall be possible to release the controller from soft reset at a specific time by the host.

3.9 Support of Application Agreement Protocols

⇨ Req: The protocol has to support the realisation of application agreement protocols. This requires multiple sending slots to achieve agreement within one communication cycle. .

3.10 Support of Network Management

⇨ Req: Support of synchronous distributed application startup and shutdown with acceptable timing and fault tolerance characteristics.

⇨ Req: Support of node and network modes with high security against critical inadvertant mode changes.

4 Hardware Specification

This chapter contains a description of hardware-related requirements for a FlexRay system.

4.1 General Requirements

Regarding the communication hardware, a distributed system of FlexRay nodes must offer some properties when being designed by using active stars and passive busses:

⇨ Req: One and two channel solutions have to be supported.

⇨ Req: An electrical and optical physical layer must be supported.

⇨ Req: Communication via redundant physical links is optional. The communication system must support both communication via redundant and non-redundant physical links. A mix of redundant and non-redundant physical links must be supported.

⇨ Req: When using active stars several 1:1 links must be used.

⇨ Req: Wake-up of nodes and stars via the communication system must be supported. Signalling on one channel is sufficient for wake-up.

⇨ Req: A baud-rate from 500 Kbit/s up to 10 Mbit/s must be supported.

⇨ Req: A power mode management must be supported.

4.2 Topology

⇨ Req: The protocol has to be independent from the topology as far as possible. Mixed and homogeneous system topologies must be supported.

⇨ Req: A FlexRay network using a passive bus must be possible.

⇨ Req: A FlexRay network using a passive star must be possible.

⇨ Req: A FlexRay network using a active star must be possible.

⇨ Req: Support for different topologies/physical layers on different channels is desirable.

⇨ Req: Support for different physical layers on one channel is desirable.

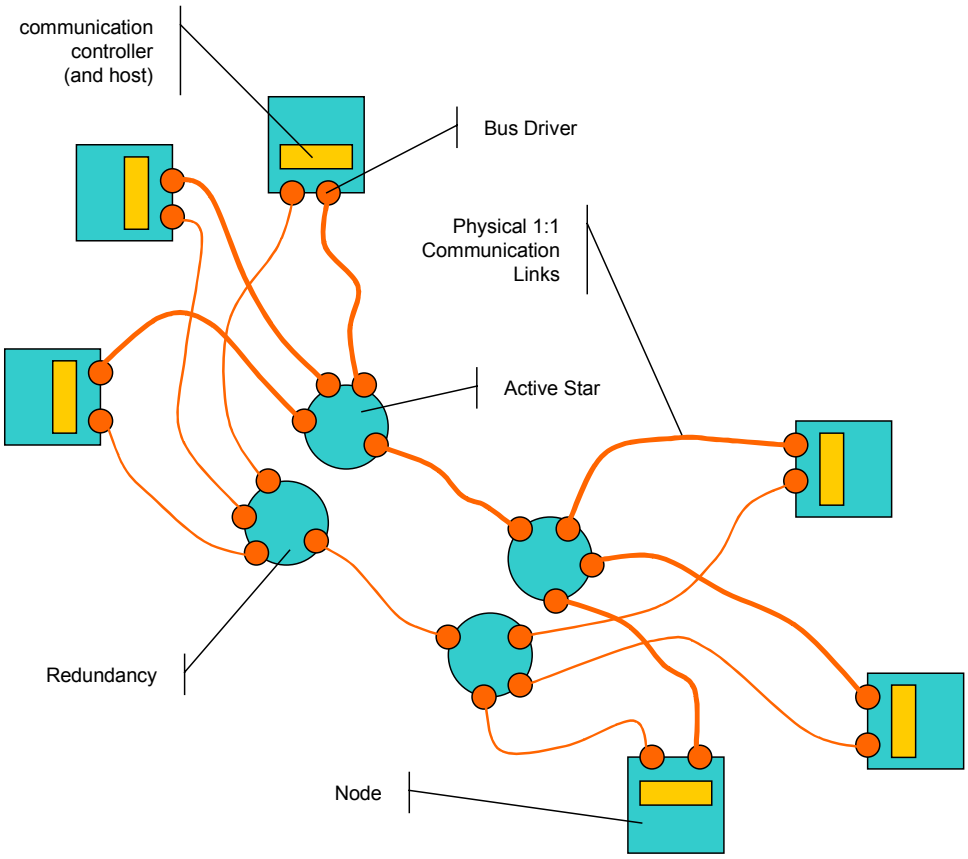

Figure 12: Example: Topology of a FlexRay network using active stars.

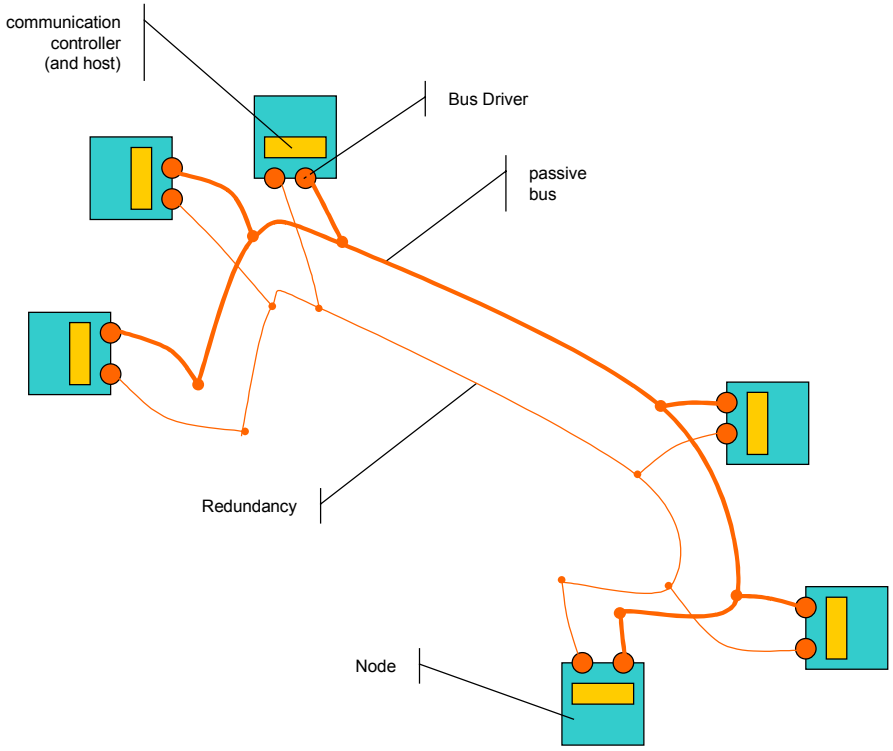

Figure 13: Example: Topology of a FlexRay network using a passive bus.

⇨ Req: A distributed system of FlexRay nodes can be designed by combining the active star and the passive bus approach. Several nodes may be connected to a branch.

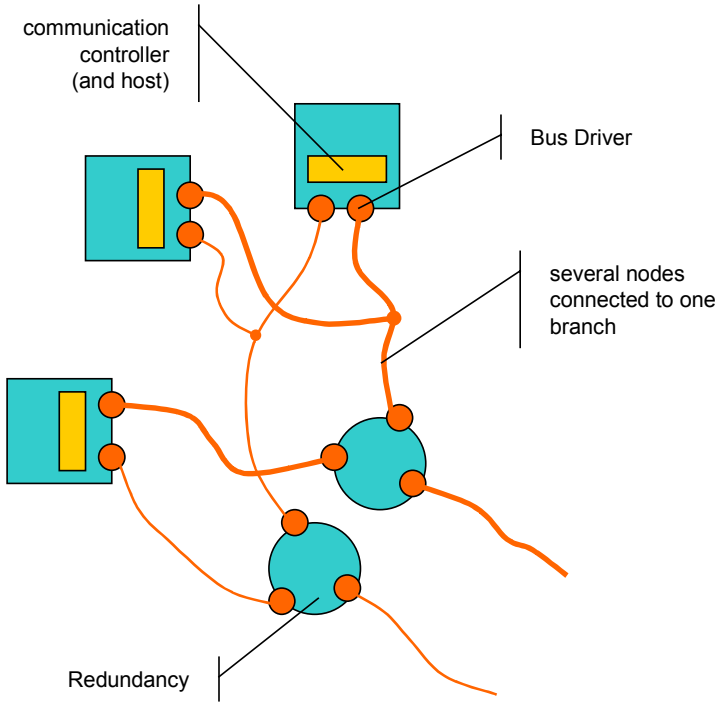

Figure 14: Example: Topology of a FlexRay network using an active star combined with a passive bus.

Each node must adhere to the following requirements:

⇨ Req: Within a node 1 or 2 bus drivers must be connected to a single communication controller

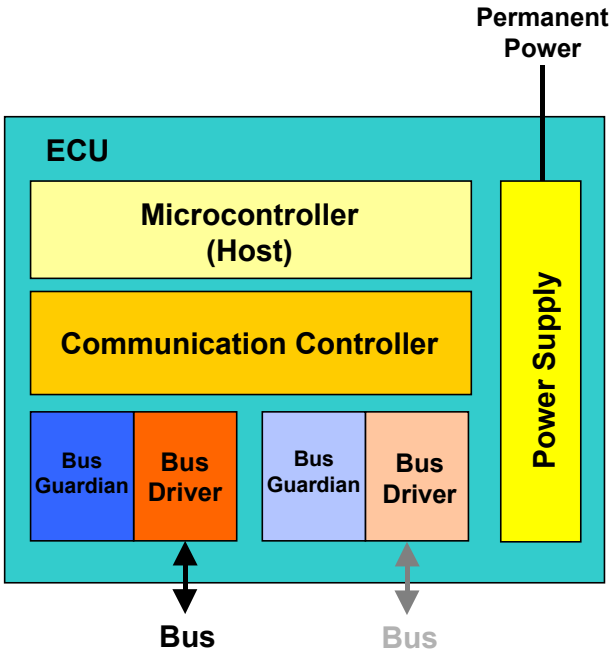

Figure 15: Block chart of a FlexRay node.

Each active star must adhere to the following requirements:

⇨ Req: No communication controller is required to perform the star functionality.

⇨ Req: No host is required to perform the star functionality.

Comment:

An implementation may integrate the star within an ECU.

⇨ Req: A branch of an active star has to be de-activated if a faulty communication signal is detected:
1) permanent “0” on the bus
or
2) permanent “1” on the bus
or
3) permanent noise on the bus

⇨ Req: A de-activated branch may not influence the communication of the active modules (fail silent).

⇨ Req: A de-activated branch has to be re-activated if the failure condition which leads to a faulty communication signal is no longer available.

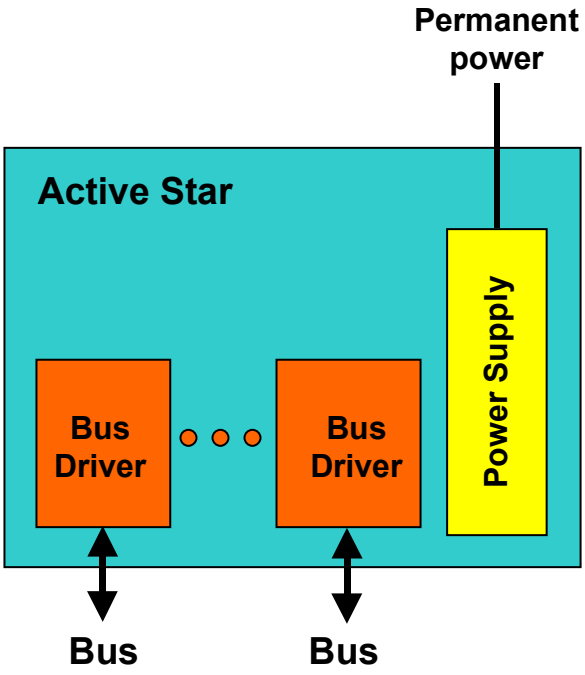

Figure 16: Block chart of an electrical active star.

4.3 Automotive Constraints

⇨ Req: FlexRay devices must meet automotive temperature requirements.

Comment:

General temperature requirements include a range of –40 to +125 degrees Celsius. Special applications may require higher temperatures, e.g., near braking actuators.

⇨ Req: Each product has to be optimised to meet the automotive and legal EMC requirements[3]. External filters may not be required.

Comment:

Listed severity levels as named won't be achievable when using a passive bus.

⇨ Req: The power consumption during the normal operating mode and the low power mode has to be minimised.

Comment:

Typical values are given in the following table:

[3] Strongest severity level; DaimlerChrysler: A0000007199, BMW: GS 95002

Function	Min.	Typ.	Max.	Unit	Remark
Quiescent current (from the permanent power) of the bus driver		10		µA	The bus driver monitors wake-up events, the voltage regulator is switched off
Current for the bus driver and the communication controller		10		mA	bus free – no actual communication
Current for the bus driver and the communication controller		50		mA	bus busy – communication active

Table 1: Typical current consumption.

⇨ Req: The voltage supply for the communication controller should be the same as for commercially available ECUs.

Comment:

Today most ECUs support 5 V. For optimisations e.g., 3 V are allowed.

⇨ Req: All inputs and outputs of the bus driver and the communication controller which are directly coupled to the wire harness have to fit the known electric requirements[4]. Support of future high supply voltages (36/42 V instead of 12 V) must be supported.

4.4 Architecture – Power Modes

The chapter summarises the requirements on the communication controller and the bus driver to run an ECU in several modes.

⇨ Req: The power modes of the ECU must be sensitive to control signals from the host and wake-up signals from the transmission media, from the ECU internally (e.g., from the host) and optionally from the ECU externally (e.g., by a switch).

⇨ Req: At least 3 power modes must be distinguished for communication controllers and bus drivers and stars:
Normal (voltage regulator(s) active, communication possible)
Standby (voltage regulator(s) active, communication not possible)
Sleep (voltage regulator(s) not active, communication not possible)

[4] Schaffner pulses (high ESD), power voltage up to 60 V, see EMC Target Specifications.

Power Mode of the Node	Communication	Power Supply
Normal	available	available
Standby	not available	available
Sleep	not available	not available

Table 2: Power modes.

⇨ Req: The power modes of the active star must be controlled by the bus drivers automatically. It is not desireable that a dedicated 'wake-up' and 'shut-down' command is send to the star or additional wiring is required.

4.5 Communication Controller

⇨ Req: A communication controller must include an interface[5] to connect a host.

⇨ Req: Parts of the behaviour visible to the host, e.g., application control, configuration, message area, communication controller status, interrupts, etc. have to be confirmed among the suppliers and specified.

⇨ Req: The time base of the redundant channels must be the same within each node (e.g., by a single state machine).

⇨ Req: The functionality of the communication controller must be independent of the existence of a bus guardian.

⇨ Req: For a stand-alone controller the pin-out must be completely specified and documented.

4.5.1 States and Operating Modes

⇨ Req: Power-on and NOT (power-on) must be distinguished at least.

⇨ Req: The controller has to be passive outside the mode power-on.

⇨ Req: The controller has to be resetable externally.

⇨ Req: The controller has to support at least one low-power mode.

⇨ Req: The controller modes have to be defined in accordance with the bus driver modes.

[5] Has to be confirmed among the suppliers, the most commonly used µCs have to be connectable.

4.5.2 Logical Line Operation

⇨ Req: At least the following information has to be distinguished:

- bus busy: data or SOC symbol are transmitted.
- bus idle.

⇨ Req: The encoding/decoding method has to allow both optical and electrical communication networks. At least one method has to be supported:

- NRZ (see byteflight specification).

⇨ Req: Bit sampling must be robust against disturbances typically inside vehicles e.g., signal delay, edge jitter, baud-rate jitter

⇨ Req: Bit sampling must be able to deal with e.g., temperature variations or tolerances of electrical and physical parameters.

4.6 Optical Driver

⇨ Req: See the byteflight specification.

4.7 Electrical Bus Driver

⇨ Req: For a bus driver the pin-out must be completely specified and documented.

⇨ Req: For redundant configurations an implementation has to be chosen which minimises the probability of common mode failures of both bus drivers (→ each redundant communication is disturbed).

Comment:

Two bus drivers to support redundant communication by a single communication controller may possibly not be implemented on a single die.

Two bus drivers to support redundant communication by a single communication controller can be integrated in one package, if any common mode failure can be excluded.

⇨ Req: The bus driver must provide status information and diagnostics information which can be read by any µ-controller optionally.

⇨ Req: The bus driver must be protected against electrical over-voltage and short-circuits.

4.7.1 Voltage Monitoring

⇨ Req: The bus driver must monitor the battery voltage and has to provide status information.

⇨ Req: The bus driver must detect an interrupted connection to the battery and has to provide status information.

4.7.2 States and Operating Modes

The bus driver has to support several states or operating modes:

⇨ Req: Power-on and NOT(power-on) must be distinguished
→ permanent power
→ regulator voltage

⇨ Req: The bus driver has to support at least two low-power modes.

⇨ Req: The bus driver has to be able to signal an internal power down mode to an external voltage regulator.

⇨ Req: the bus driver has to support a “shutdown mode”
→ this mode has to be reached by secured mechanisms on demand of the host
→ this mode must be left only by a power down
→ the bus driver has to be passive and has to signal to the voltage regulator to switch off.

⇨ Req: The bus-levels have to be chosen by the bus driver automatically to support any net-wide power down modes.

4.8 Bus Guardian

⇨ Req: The failure of a communication controller in the time domain, e.g., a communication controller sends in a time slot when it is not allowed to send, must be prevented by a bus guardian. The probability for common mode failures in the time domain affecting both, the communication controller and the bus guardian must be sufficiently low.

⇨ Req: The bus guardian must protect the static slots (controller) from each other. In the dynamic segment the bus guardian grants all controllers access to the bus.

Comment:

One of the main reasons for an error in the time domain is an erroneous internal state that leads to an incorrect (timing) access to the communication media.

⇨ Req: The bus guardian must be able to detect errors in the physical clock source as well as errors in the internal representation of the time base of the communication controller.

Comment:

One of the main reasons for an error in the time domain is an error in the clock source of the communication controller. Hence, the clock source check mechanism of the bus guardian must concentrate on the main physically possible failure modes of the clock source of the communication controller.

The bus guardian may have a clock source of its own. Two bus guardians, which are connected to the same communication controller can use the same clock source.

⇨ Req: The bus guardian must not disable access to more than one channel, i.e., one bus guardian per channel is required.

⇨ Req: It must be possible to implement the bus guardian as a stand-alone circuit. This circuit has to be combinable with the known state of the art physical layers.

Comment:

The bus guardian could be integrated in the bus driver. The interface(s) towards the communication controller (and the bus driver) must be defined.

⇨ Req: It must be possible to implement the bus guardian as a stand-alone circuit in the star coupler. This circuit has to interact with the known state of the art physical layer.

⇨ Req: The bus guardian has to check the correct enabling of the driver output stage.

⇨ Req: The mechanism of separating the communication controller from the communication media must be checked. At least once per driving cycle (power on / power off) is sufficient.

⇨ Req: The bus guardian is configured via a configuration data interface.

⇨ Req: The bus guardian has to enable and disable the bus driver output stage according to a predefined timing pattern. If the bus guardian detects an error in the timing pattern of the communication controller it permanently disables access to the communication media and signals this.

⇨ Req: If an error in the bus guardian occurs the communication channel must not be disturbed (monopolised).

⇨ Req: The configuration data interface must be specified and documented. This mainly includes the logical contents of the timing pattern.

4.9 Wake-up

⇨ Req: Several wake-up mechanisms have to be taken into consideration.

4.9.1 ECU → Bus Driver

⇨ Req: The bus driver has to be woken up by any source inside or outside the ECU (local wake-up).

⇨ Req: The wake has to be two edge sensitive.

Example:

e.g., edge at a WU pin of the BD

4.9.2 Bus → Bus Driver

⇨ Req: From the host´s point of view a general wake-up mechanism is required for both electrical and optical systems.

⇨ Req: The bus driver should be woken up via standard communication (message-pattern).

⇨ Req: The wake-up detector has to be robust against disturbances in vehicles like common mode signals by emission.

⇨ Req: The bus driver may not be woken up by any noise.

4.9.3 Bus Driver → Controller

⇨ Req: The bus driver has to wake up the controller by any signal on the interface. A dedicated wake-up line is not required.

Example:

e.g., edge at the Rx pin produced by the bus driver

4.9.4 Bus Driver → Power Supply

⇨ Req: The bus driver has to signal its sleep state, e.g., to control the voltage regulator.

Example:

inhibit signal

4.9.5 Controller → Bus Driver

⇨ Req: The controller must be able to wake-up the bus driver by any signal on the interface. A dedicated wake-up line is not required.

Example:

Edge at the Tx pin

4.10 Selective Sleep

⇨ Req: The realisation of selective sleep has to be supported.

5 Interfaces

The interfaces between the single modules (host, controller, bus driver, bus guardian, and power supply) have to be agreed upon among the suppliers according to the general requirements defined in this document. Figure 17 shows an overview of all interfaces.

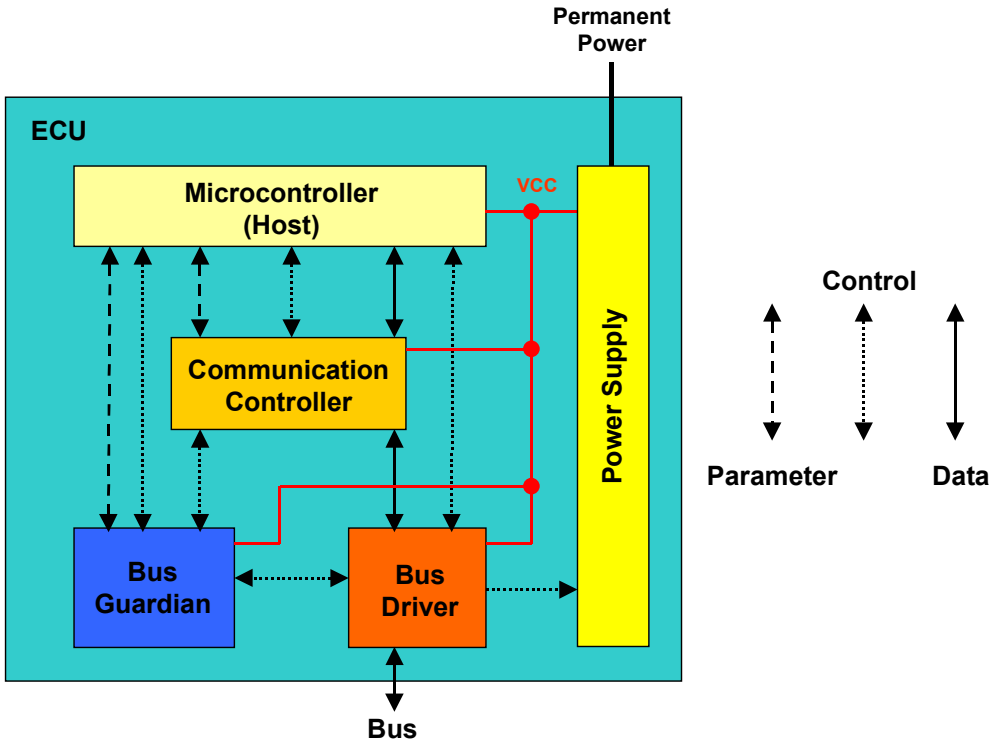

Figure 17: Overview of the interfaces.

5.1 Host ⇔ Communication Controller (CHI)

5.1.1 General Requirements

⇨ Req: The configuration data of the communication controllers must be securable (e.g., by soft reset) against accidental access and modifications.

⇨ Req: Protect against improper host modification of BG and CC configuration data.

⇨ Req: The interface between the host and the communication controller should be implemented as a 16 bit interface (selectable multiplexed/non-multiplexed bus interface).

⇨ Req: Functional compatibility between different suppliers has to be guaranteed at CHI level.

⇨ Req: The CHI has to be configurable into transmit and dedicated receive buffers and receive buffers with FIFO behaviour.

⇨ Req: If the FIFO queue is full and new data arrives the oldest data is overwritten. The FIFO queue must set a diagnosis bit when data in the buffer is overwritten.

⇨ Req: MSB first is used for frame transmission.

Comment:

The status area of the CHI contains communication controller status fields, which are written by the controller and which are read-only for the host. The status fields will be defined in the protocol specification and interface specification.

The control area in the CHI contains fields that allow the host to control the behaviour of the communication controller. The control fields will be defined in the protocol specification and interface specification.

The message area in the CHI contains a memory area where the frames to be sent / received are stored together with status information for each frame. The layout of this memory area is determined in the configuration data of each communication controller. The message buffer status fields will be defined in the protocol specification and interface specification.

⇨ Req: Support of traceability of system- and component-level faults to identify root causes of failures.

⇨ Req: The hardware implementation should verify that only one combination of frame ID and sync bit is considered valid for transmission.

5.1.2 Frame Filtering and Masking

⇨ Req: Message Reception, every message buffer has to contain a channel, frame ID and cycle counter which are used for message filtering. Optionally the first two data bytes of each message buffer are used as message ID filter.

Options for filtering:
1) Frame ID + channel
2) Frame ID + cycle counter + channel
3) Message ID + cycle counter + channel

Comment:

Filtering: Filtering of messages means that for each message buffer the message´s frame ID, cycle count and message ID are parameters that define in which message buffer the correctly (correct CRC, time, etc.) received messages is stored or if the message is discarded.

⇨ Req: There must be at least one mask register per communication controller and channel that allows all combinations of masking.

Comment:

Masking: Masking of message filters means that some (parts) of the filtering parameters may be configured to be disregarded (set to "don´t care").

⇨ Req: Message transmission, filtering parameters for transmission are frame ID and cycle counter; there is no masking possible for the frame ID. Each

transmit buffer has it´s own mask for the cycle counter.

5.1.3 Interrupts

⇨ Req: The host computer has to be able to request different interrupts from the communication controller: at least read interrupt (buffer), write interrupt (buffer), 2 independent timer interrupts.

⇨ Req: Timer interrupt: the host can request a time interrupt at any absolute point in the global time (across communication cycle borders).

⇨ Req: One interrupt line is required for a standalone controller implementation.

⇨ Req: Interrupts can be mapped to one or more interrupt lines in an integrated controller.

5.2 Host ⇔ Bus Guardian

⇨ Req: The bus guardian configuration data is written during download and then stored in a local memory of the bus guardian.

⇨ Req: During normal operation no configuration data transfer from the host to the bus guardian is allowed.

⇨ Req: The bus guardian periodically updates a status field which can be accessed by the host / bus guardian interface containing at least the following status information:

- State of the bus
- State of the controller

5.3 Communication Controller ⇔ Bus Guardian

⇨ Req: At least the following control information is required:

- Arm signal

5.4 Communication Controller ⇔ Bus Driver

⇨ Req: This interface has to be confirmed among the suppliers.

Example:

Tx, TxEnable

Rx, RxEnable

5.5 Bus Guardian ⇔ Bus Driver

5.6 Bus Driver ⇔ Power Supply

⇨ Req: To perform the wake-up and sleep functionality an interface between bus driver and power supply is required.

6 Error Handling

The communication system and its components shall offer adequate error management mechanisms to deal with faults arising from the following levels:

- media
- bit (coding)
- frame
- data
- topology and
- time.

The communication system furthermore shall offer diagnosis information to the host computer with respect to controller, bus (channel), and incoming/outgoing link failures.

6.1 Requirements

⇨ Req: The error management shall follow the “never-give-up” philosophy.

Comment:

This means that the communication protocol has to support proper operation until a certain critical error states is reached.

⇨ Req: The non-arrival of periodic messages shall not be unrecognised.

Comment:

It is okay, if, e.g. one, periodic message is missed, but this has to be detected. The fact, that a periodic message was missed should be signaled to the host.

⇨ Req: If a periodic message was missed, no random data shall be given to the host.

⇨ Req: Data content of messages, (periodic and spontaneous) must not be changed by the communication protocol.

⇨ Req: The change of data content shall be signalled to the host.

⇨ Req: After an error was detected at a communication partner in the network, the functionality of the other communication partners shall not be influenced.

Comment:

The correct function may not depend from the correct function of a certain host, of a certain communication controller or of a certain power supply.

The communication controller shall detect the following list of errors:

⇨ Req: Synchronisation error. The communication controller is not any more synchronised to the global time on the bus.

⇨ Req: The communication network must offer diagnosis information to the host computer with respect to the bus (channel), incoming/outgoing link failures.

⇨ Req: The communication network must offer diagnosis information to the host computer within a defined maximum delay after the occurrence of the failure of the diagnosis element.

⇨ Req: The communication network is not required to provide consistent and agreed diagnosis information to the host computer.

6.2 Hardware Units

The following faults have at least to be detected by the communication controller:

⇨ Req: Defect time source (e.g., broken crystal).

⇨ Req: Low voltage.

The following faults has to be recognised by the bus driver as errors:

⇨ Req: Faulty communication signals caused by e.g. any faulty transmission media (e.g., a broken line, short circuit to ground,...).

⇨ Req: Incorrect communication with the host e.g., communication via the data interface.

⇨ Req: Incorrect communication with the communication controller e.g., bus-blocking transmit signals.

⇨ Req: De-activated branch.

6.3 Interfaces

⇨ Req: Status information on detected errors must be provided. Additionally it is required that maskable interrupts for certain detected errors can be requested by the host.

7 Constant Definitions

In this section the constants for a number of design parameters defined throughout the document are set to actual values.

7.1 Communication Network Constants (Min/Max)

Name	Value	Description
CONTROLLER_MAX	64	Maximum number of controllers connected to one communication channel.
CYCLE_LENGTH_MIN	250 µs	Minimum length of the configurable communication cycle.
CYCLE_LENGTH_MAX	64 ms	Maximum length of the configurable communication cycle.
DATA_BYTES_MAX	246	Maximum number of data bytes.
DYNAMIC_IDS	4095	Maximum number of dynamic identifiers.
STATIC_SLOTS_MIN	2	Minimum number of static slots in a static segment of a communication cycle
STATIC_SLOTS_MAX	4095	Maximum number of static slots in a static segment of a communication cycle

Table 3: Definition of the constants used throughout the specification.

8 Glossary

Bus ..... Consists of one or several *channels*.

Bus Driver ..... A bus driver connects a communication controller to one channel.

Bus Guardian ..... A bus guardian protects one channel from timing failures of the communication controller. It is therefore connected to one communication controller and one bus driver. The bus guardian must be independent from the protocol communication controller.

byteflight ..... Communication network developed by BMW AG, Motorola, ELMOS, Infineon, Siemens EC, Steinbeis Transferzentrum für Prozessautomatisierung, IXXAT. See http://www.byteflight.com

Channel ..... A channel is a physical connection between several communication controllers. A redundant channel consists of two channels connecting the same communication controllers.

CHI ..... Controller Host Interface.

Clique ..... Set of communication controllers having the same view of certain system properties, e.g., the global time value, or the activity state of communication controllers.

Cluster ..... Synonym for network within this specification.

Cluster time ..... Same as *cycle time.*

Communication Controller ... A communication controller is connected to one or two channels where it can send and receive frames.

Communication Cycle ..... Periodic data transfer mechanism. Structure and timing are statically defined. However, a static and a dynamic segment allows for the transmission of both, state and event information.

Controller ..... see, *Communication Controller*.

CRC ..... Cyclic Redundancy Code attached to a frame.

CYCLE ..... The CYCLE field is used to transmit the cycle counter. The cycle counter is increased simultaneously in all nodes by the communication controller at the start of each new communication cycle.

Cycle Counter ..... Contains the number of the current communication cycle.

Cycle time ..... Contains the time within a communication cycle in units of macroticks. Same as *cluster time.*

DATA ..... Data field in a frame.

DLC: ..... Data length field

Dynamic Segment..... Segment of the communication cycle where frames are transmitted according to a mini-slotting algorithm. The sending order is defined by a statically determined identifier. Identifiers with smaller numbers have priority over identifiers with higher numbers. A communication cycle may consist of the static segment only.

EOF ..... End Of Frame. An optical or electrical physical layer may require different end of frame sequences.

ECU ..... Electronic Control Unit. Same as *node*.

EMC ..... Electro Magnetic Compatibility.

FIFO ..... First In First Out. Buffers can be configured to work as a FIFO memory for frames.

Frame..... A frame consists of all information transmitted in one slot (with one identifier) on one channel.

FTA ..... Fault Tolerant Average. The FTA is a fault tolerant clock synchronisation algorithm that is able to tolerate up to a pre-defined number *k* of maliciously faulty clocks. This algorithm is based on a sorted array of clock deviations. The lower and upper k clock deviation values are discarded. From the remaining clock deviation values the average value is calculated and then used for the clock correction.

FTM..... Fault Tolerant Midpoint. The FTM is a fault tolerant clock synchronisation algorithm that is able to tolerate up to a pre-defined number *k* of maliciously faulty clocks. This algorithm is based on a sorted array of clock deviations. The lower and upper k clock deviation values are discarded. From the remaining clock deviation values the median value is chosen for the clock correction.

Gateway..... A node may function as a gateway and connect two or more networks.

Global time..... Contains the combination of cycle counter and cluster time.

Hamming Distance ..... Minimum distance of any two code words within a code.

Host ..... The host is the part of an ECU where the application software is executed, separated by the CHI from the communication network.

ID..... The frame identifier defines the slot position in the static segment and defines the priority in the dynamic segment. A lower identifier determines a higher priority. Identifier 0 is reserved for the SOC symbol. The identifier of a frame must be unique within a cluster. Each controller can have one or more identifiers (in the static and the dynamic segment).

Identifier ..... see, *ID*.

IFG..... Inter Frame Gap.

LEN ..... Length field of a frame.

LLI ..... Logical Line Interface.

LSB..... Least Significant Bit/Byte.

MAC ..... Media Access Control.

Macrotick..... Basic unit of time measurement within a network of communication controllers. The clock synchronisation mechanism guarantees that the clock values at all non-faulty controllers are equal. The uncertainty in the clock values is bounded by the precision.

Message ..... Application data transported within a frame. Several messages may be packed together to constitute a frame.

Microtick ..... Basic unit of time measurement within a communication controller for measuring the time difference between the controllers clock and the clocks of the other communication controllers. Clock correction is done in units of Microtick.

MSB..... Most Significant Bit/Byte.

Network ..... A network consists of a set of nodes (more than one node) connected by communication subsystem. Networks are connected by special nodes (gateways). Same as *Cluster*.

Node ..... A node may contain one or more communication controllers. Equivalent to *ECU* (Electronic Control Unit).

Nominal Precision ..... The nominal precision is the clock synchronisation precision that can be reached by the local clock synchronisation of a cluster.

NRZ ..... Non Return to Zero physical layer coding scheme.

Precision ..... The precision is a time interval bounding the deviations between the local clocks of all active communication controllers. If a communication controllers clock deviates more than the precision from the clocks of the other controllers it must not participate in the communication any longer.

Slot..... Duration of time during which one communication controller may access the communication medium.

SOC ..... Start of Cycle. Defines the start of a communication cycle in the pure dynamic mode, and the start of communication for start-up.

SOF ..... Start of Frame. An optical or electrical physical layer may require different start of frame sequences.

Star Coupler ..... A star coupler is connected to one channel.

Static Segment ..... Segment of the communication cycle where frames are transmitted according to a statically defined TDMA scheme. A communication cycle may consist of the static segment only.

SYNC ..... Synchronisation field. This field indicates that the slot is used for clock synchronisation.

TDMA ..... Time Division Multiple Access.

9 Index