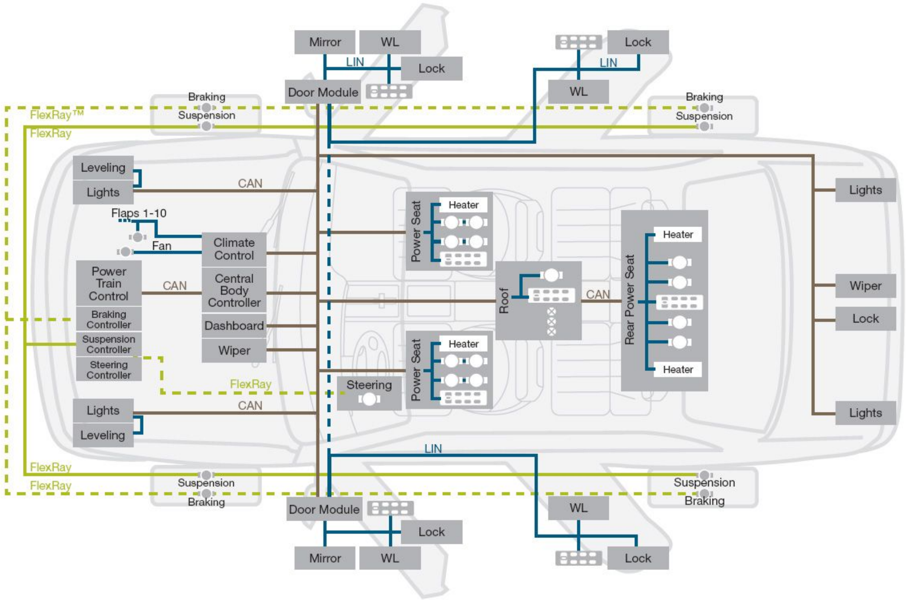
FlexRay

Communication Protocols

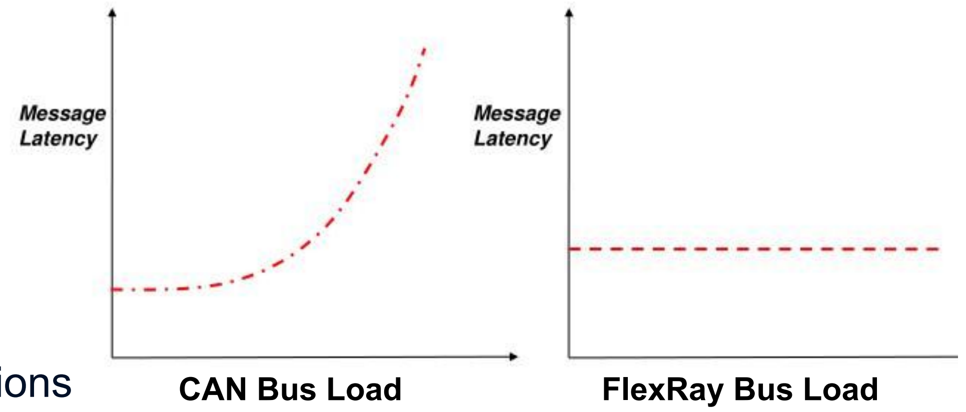
FlexRay

- ❖ FlexRay is a serial communication system, developed by FlexRay Consortium
- ❖ FlexRay is a deterministic and fault tolerant communication standard
- ❖ FlexRay is used in safety-critical, time-critical and high speed automotive applications
  - ➢ E.g. Steering system, braking system, engine control system and etc.
- ❖ The FlexRay consortium published version 3.0.1 of the specification in 2010
- ❖ ISO standardized FlexRay version 3.0.1 as ISO 17458 in 2013
- ❖ The ISO 17458 describes the physical layer and the data link layer of FlexRay



FlexRay - Why FlexRay?

- CAN has some limitations for more intensive and safety-critical applications
  - Lack of redundant mechanism to assure fault tolerance
  - Communication speed is limited to 1 Mbps (typically 500 kbps)
  - Unpredictable latency because of the event-driven communication and bitwise arbitration
    - Using bit arbitration to resolve collisions leads to a communication flow that cannot be determined until runtime and message latency is unpredictable. Specially in a high load bus.
- Main Properties of FlexRay
  - Fault-tolerant - 2 independent channels
  - High Speed - 10 Mbps per channel
  - Deterministic - message latency is predetermined
  - Static and dynamic data transmission
  - Flexible - can be customized for different applications

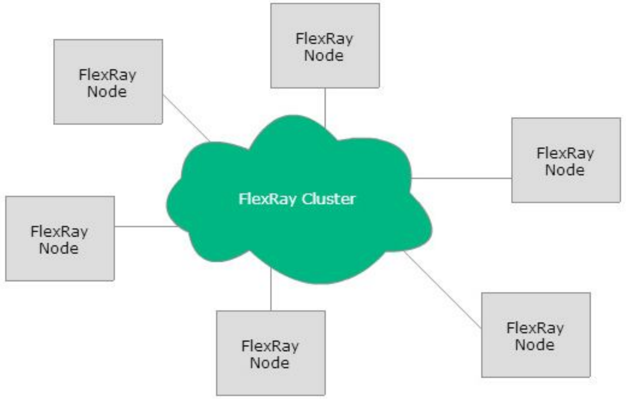


FlexRay - Features

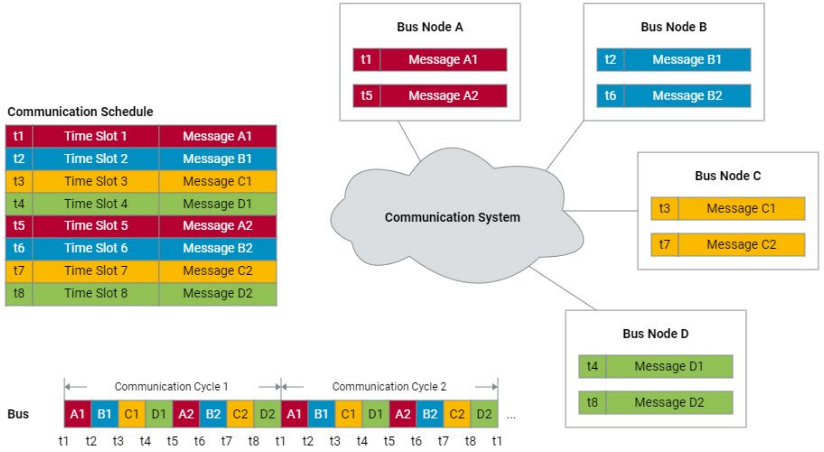
- Offers flexible configurations and support of different topologies
- Each FlexRay node has two communication channels to enable redundancy
  - Max data rate of 10 Mbps per channel
  - When redundancy is not required channels may be used to increase data rate to 20 Mbps
- Based on time-triggered communication principles and static bandwidth allocation
- Optional support for event-triggered communication and dynamic bandwidth allocation
- All communication is performed based on a communication schedule
- Fault tolerant and time triggered services implemented in hardware
- Is deterministic and fault-tolerant. It can be used in real-time and safety critical applications
- Support of optical and electrical physical medium
- More complex and expensive than LIN and CAN

FlexRay - Communication Architecture

- A FlexRay communication system (FlexRay Cluster) is made up of
  - A number of FlexRay nodes and
  - A physical medium that interconnects all of the FlexRay nodes.
- A FlexRay cluster is not restricted to any specific physical topology
- FlexRay provides a redundant communication channel
- FlexRay is based on a time-triggered communication architecture

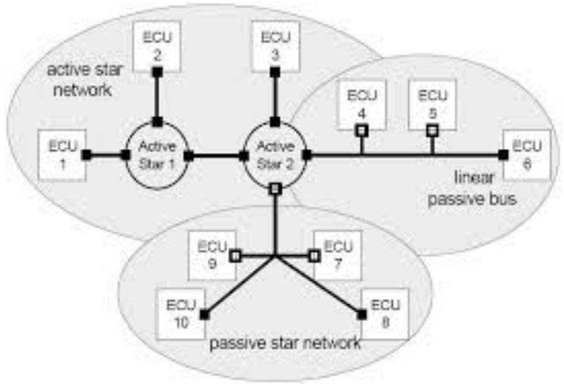


- FlexRay uses TDMA(Time Division Multiple Access) method to implement the time-triggered principles
  - Time is divided to communication cycles and each communication cycle is divided to a number of slots
  - A specific time slot is assigned to each FlexRay message per communication cycle and each message must be transmitted in its own time slot
- Communication schedule is required



FlexRay - Network topology

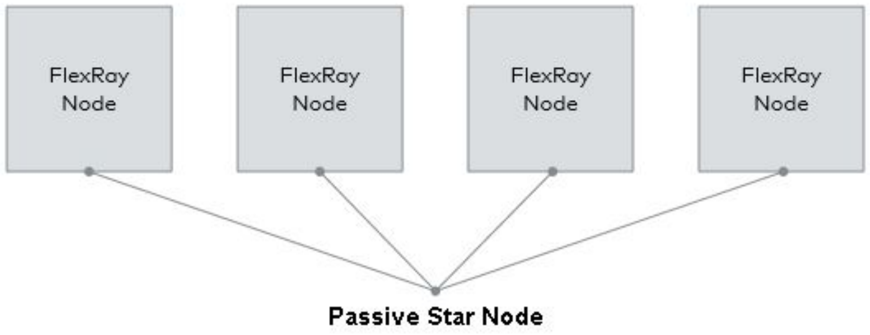
- There are several ways to design a FlexRay cluster. It can be configured as
  - A single channel or dual channel **bus** network
  - A single-channel or dual-channel **star** network
  - In various **hybrid** combinations of bus and star topologies



- Each node in a cluster has 2 channels; Channel A and Channel B
  - Nodes may be connected to one or both of them
  - A node connected to Channel A can communicate with all other nodes connected to Channel A
  - A node connected to Channel B can communicate with all other nodes connected to Channel B
  - If a node needs to be connected to two clusters then the connection to each cluster must be made through a different communication controller (an **active star coupler**)

FlexRay - Passive Star and Bus Topologies

- In passive topologies there is no active node
- No more than 24 meters line length is allowed between any two FlexRay nodes
- No more than 22 FlexRay nodes should be connected to a passive cluster
- By increasing the number of nodes and the lines length, the speed of the communication will be decreased



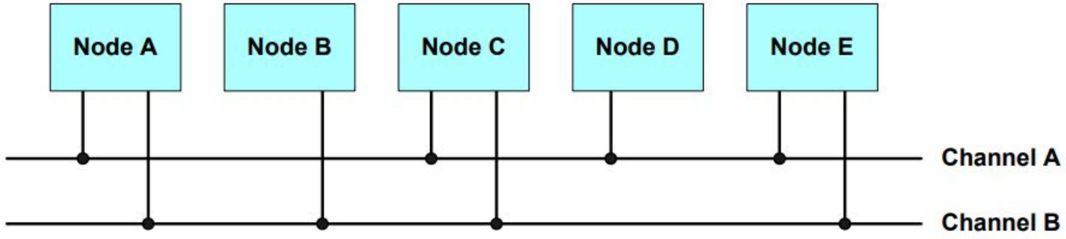
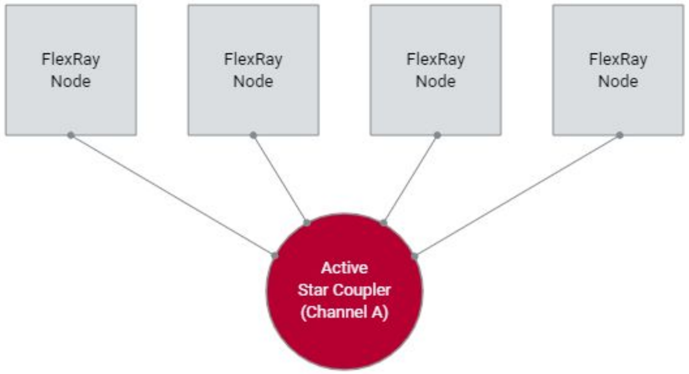


Figure 1-5: Dual channel bus configuration.

FlexRay - Active Star Topology

- An **active star coupler** actively amplifies and distributes received signal of a branch to all the other communication branches (like a hub)
- There can be no more than **two** active star couplers on a network channel
- The maximum distance between an active star coupler and any FlexRay node may not exceed 24 meters and no more than 22 FlexRay nodes shall be connected to an active star
- The advantages of the active star topology
  - It avoids propagation of errors by disconnecting faulty branches from the active star coupler
  - It makes possible to extend the number of nodes and the length of the cluster
  - The length of a FlexRay cluster may be extended from 24 meters to a max. 72 meters (ideally) by connecting two active star couplers in series.



FlexRay - Active Star and Hybrid Topologies (Examples)

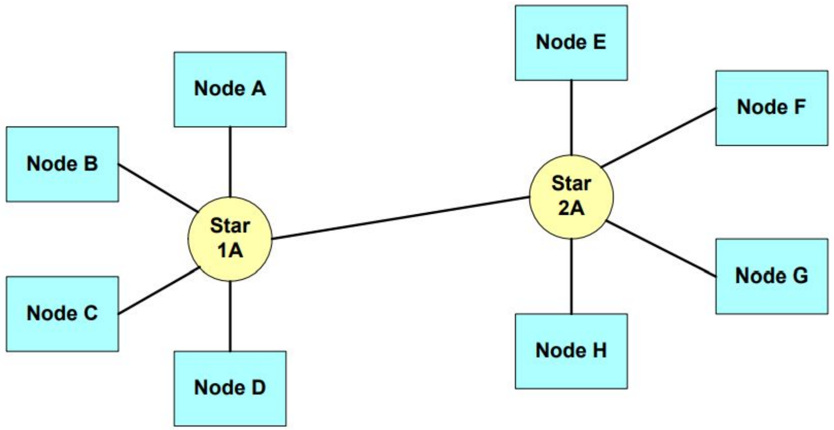


Figure 1-7: Single channel cascaded star configuration.

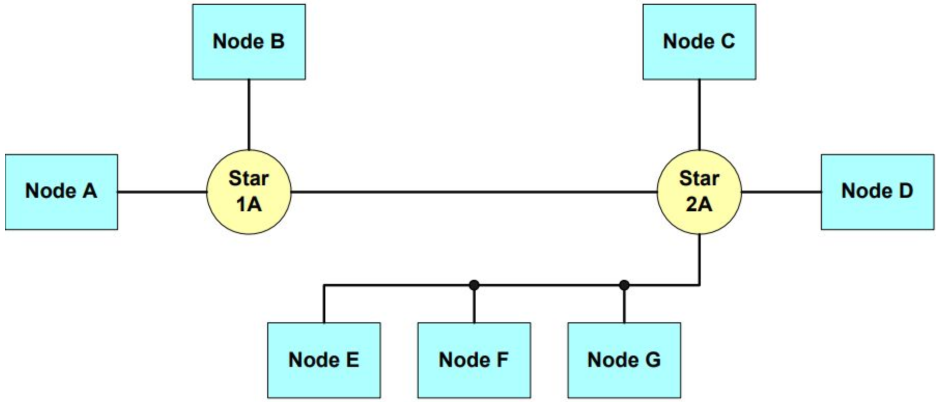


Figure 1-9: Single channel hybrid example.

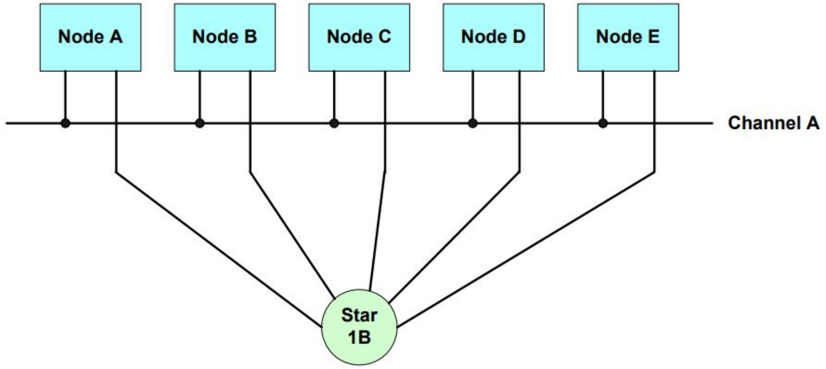


Figure 1-10: Dual channel hybrid example.

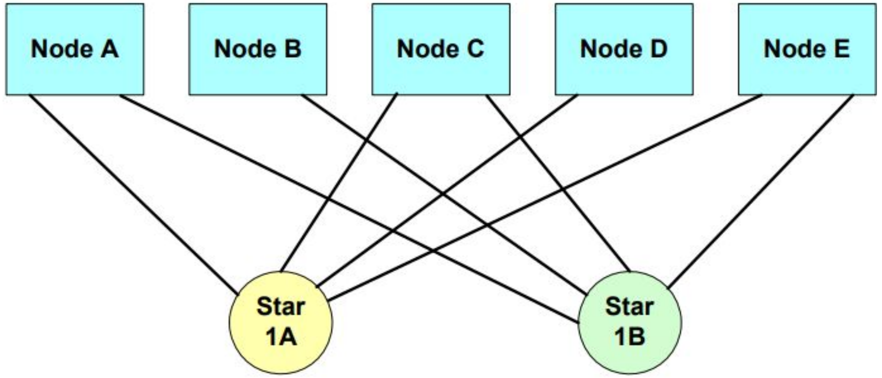
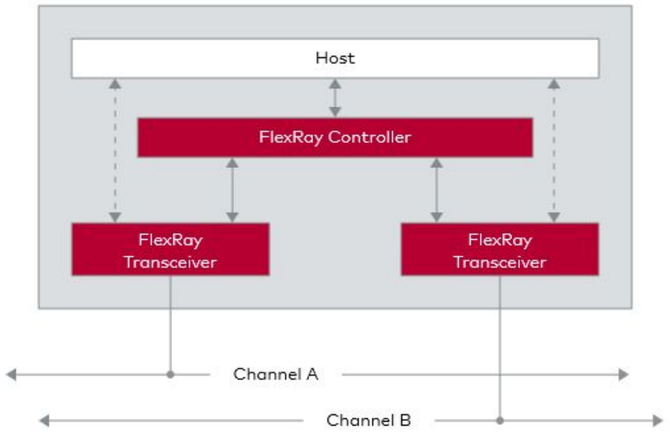


Figure 1-6: Dual channel single star configuration.

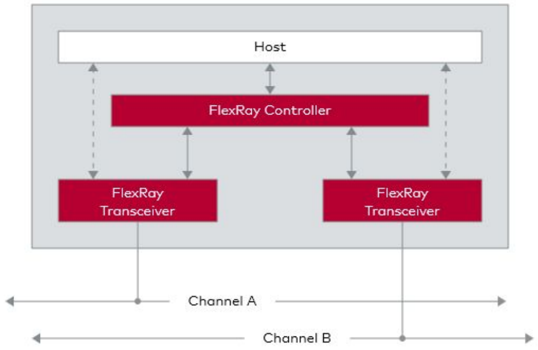
FlexRay - Node

- A FlexRay node is an ECU, connected to a FlexRay cluster via a FlexRay interface
- The FlexRay interface is made up of
  - A FlexRay controller which executes the FlexRay communication protocol; primarily it
    - Does framing, media access, error detection and handling, synchronization, putting the FlexRay bus to sleep and wake up modes, filtering and as well as coding TX messages and decoding RX messages
    - Can be standalone or integrated in host (microcontroller)
  - One or two FlexRay transceiver
    - Connects the FlexRay controller to the physical media
    - Converts the logical signal into a physical signal and vice versa.
    - Can have four different states:
      - Normal, Standby and optionally Sleep or ReceiveOnly
    - Has also an interface to the host: Two control lines STBN (Standby) and EN (Enable Input)
      - The host uses these lines to control the state of the transceiver



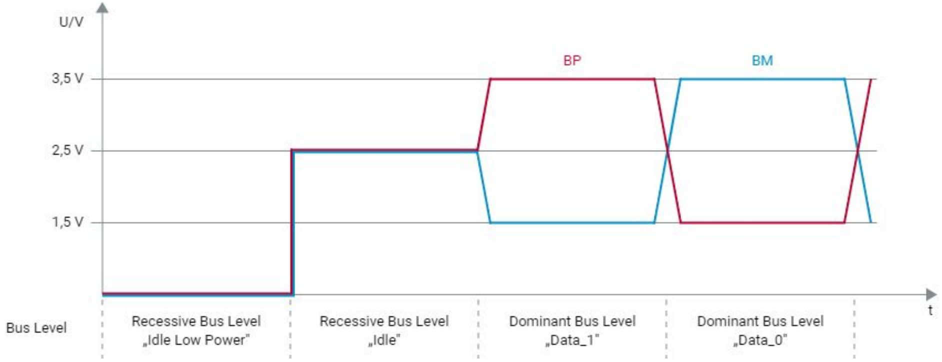
FlexRay - Node Types

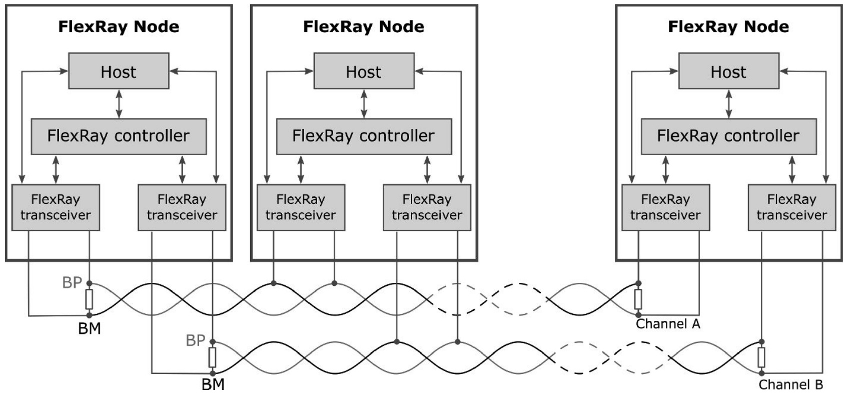
- There are three types of node in a FlexRay cluster
  - Sync Nodes
    - They actively participate in the clock synchronization performed by the cluster.
    - They transmit sync frames that are evaluated by all nodes of the cluster to determine the cycle length, clock offset and the position of the cycle start.
  - Coldstart Nodes
    - They initiate the communication and are allowed to start transmitting startup frames in the non-synchronized state in order to establish a schedule. Nodes integrate onto the new schedule by evaluating the content and timing of the received startup frames
    - A coldstart node is always also a sync node and a startup frame is always also a sync frame
  - Non-sync Node
    - Nodes that are not coldstart and sync nodes.



FlexRay - Physical Layer

- ❖ Medium: Plastic optical fiber or twisted pair wires
- ❖ Bit rates: 2.5 Mbps, 5 Mbps and 10 Mbps per channel
- ❖ Uses differential signaling and NRZ encoding
  - ➢ Two lines: Bus Plus (BP) and Bus Minus (BM)
  - ➢ Differential voltage of dominant bus levels
    - ■ Logical 1 is 2V and logical 0 is -2V
  - ➢ Recessive bus level of Idle is 2.5V and of Idle low power is 0V.
  - ➢ Recessive differential voltage is 0V (Idle & low power)
- ❖ Termination resistors are required to prevent reflections in the cluster (80 - 110 Ω)
- ❖ Max length depends on the topology
  - ➢ Ideally 24 for a passive and 72 meters for an active bus





FlexRay - Operational states

- A FlexRay node has several operational states:
  - **Configuration** (default config/config)
    - For making all kinds of initial settings,
      - Including the communication cycle and data rate
  - **Ready**: For making internal communication settings
  - **Wakeup**: For waking up a node that is not communicating
    - The node sends a wakeup signal to other nodes, which wakes up and enables the communication controller, bus driver and bus guardian.
  - **Startup**: For starting clock synchronization and getting ready for communication
  - **Normal** (Active/Passive): Communication available states (active state and error state)
  - **Halt**: For indicating that communication has stopped

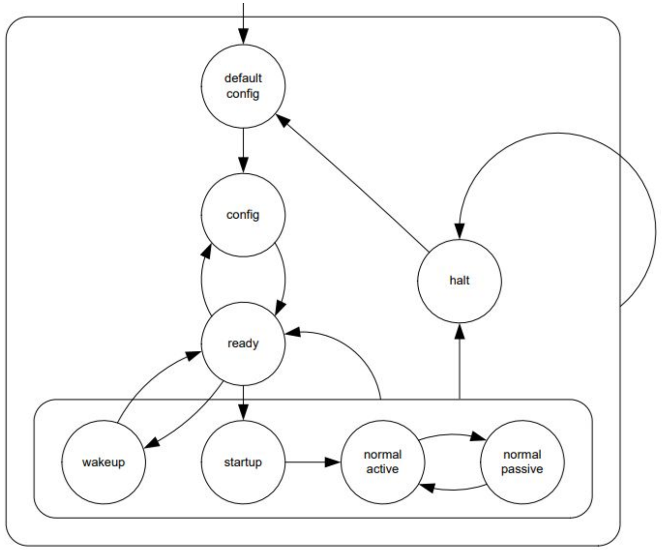


Figure 2-3: Overview of protocol operation control.

FlexRay - Operational states

- Each node also has state transitions related to error handling
- Each node has an error counter for counting
  - Clock synchronization and clock-correction errors
    - When a node clock differs from the FlexRay sync node clock
  - Frame transmission/reception status including
    - Syntax, content, bus-violation and transmission conflicts errors
- The transitions are managed based on the error counter
  - The transitions between normal active, normal passive, ready and halt
- When a node detects an error, it notifies the host processor
  - The use of the error counter depends on the application
  - Error handling is done in the application layer and is determined during system design
  - E.g. Depending on the error condition, a node can transit to halt state.

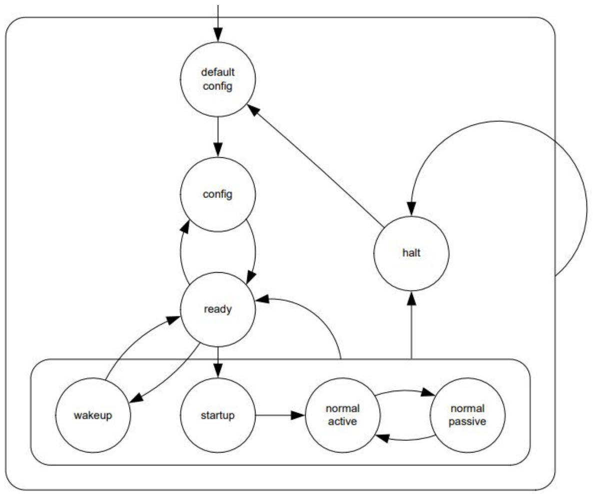
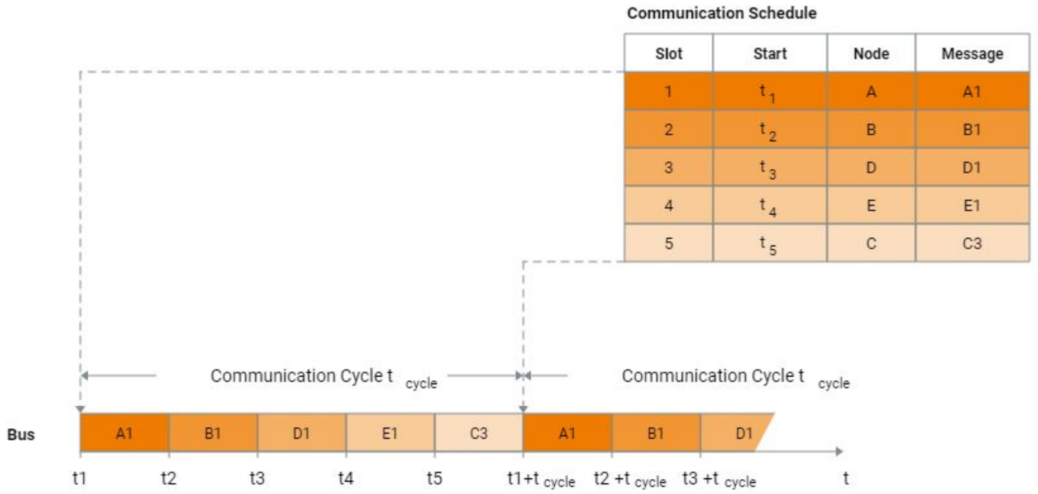


Figure 2-3: Overview of protocol operation control.

FlexRay - Bus Access

- In a FlexRay cluster, nodes are granted access to the bus by two different methods
  - TDMA (Time Division Multiple Access) - for static/time-triggered data communication
  - FTDMA (Flexible Time Division Multiple Access) - for dynamic/event-triggered data communication
- The TDMA method is based on a communication schedule
  - Is organized into a number of time slots (static slots) of equal length
  - Each static slot is assigned to a FlexRay node
  - During the communication a FlexRay node is granted exclusive access access to the bus according to this schedule
  - The communication schedule defines the FlexRay **communication cycle** and is repeated periodically by all nodes



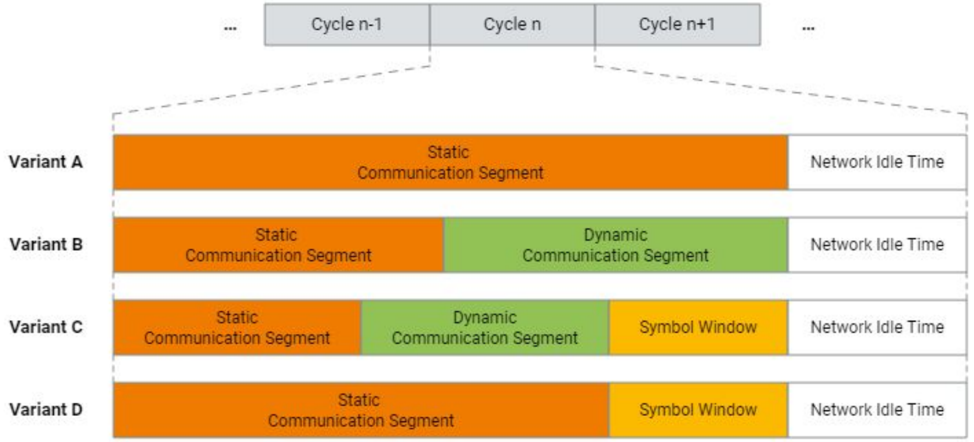
FlexRay - Bus Access

- FlexRay uses FTDMA for sporadic transmission of messages
  - FlexRay extends the cycle by a dynamic segment for messages
  - Dynamic(event-triggered) messages should also be transmitted on a fixed time schedule
  - The difference between the FTDMA and the TDMA method
    - Dynamic messages can be transmitted by relevant FlexRay nodes as needed
    - No fixed assigned between slots and nodes
    - Time of a message transmission is not predictable

  - The dynamic segment like the static segment has a finite length
    - There may be FlexRay nodes wishing to send that will not be able to transmit their dynamic messages in the current cycle
      - Such nodes should wait to the dynamic segment of the next communication cycle

FlexRay - Communication Cycle

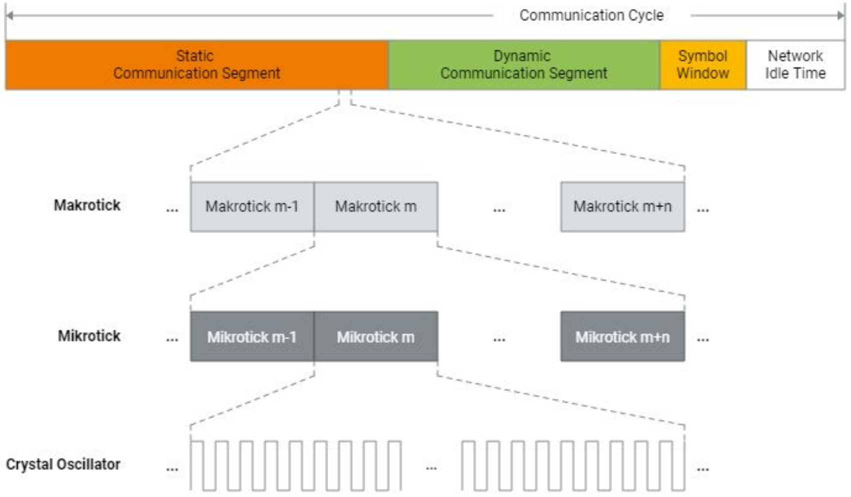
- Data communication in a FlexRay cluster is periodical and is based on a schedule
- The FlexRay communication cycle is the fundamental element of the media-access
- The duration of a cycle is fixed when the network is designed (typically 1-5 ms)
- The communication cycle is composed of at least two and at most four time segments
  - The static segment
  - The network idle time (NIT) segment
  - The dynamic time segment (optional)
  - The symbol window segment (optional)



- In dual channel systems
  - Cycles and segments start at the same time on both channels but the schedule may be different

FlexRay - Timing Hierarchy

- Time in a node is based on **communication cycles**, **macroticks** and **microticks**
- A communication cycle consists of an integer number of **macroticks**
  - Established as a result of the cluster clock synchronization
  - FlexRay nodes actively synchronize themselves and adjust their local clocks so that a macrotick occurs at the same point of time on every node across the network
  - Macroticks are synchronized at cluster level and data that relies on them, is also synchronized
- Macroticks are composed of **microticks**
  - The smallest time unit of the local clocks
  - Derived from the internal oscillator of the node
  - A macrotick consists of an integer number of microticks
  - In order to get a synchronized macrotick, the number of microticks per macrotick may differ between nodes, and depends on the oscillator frequency

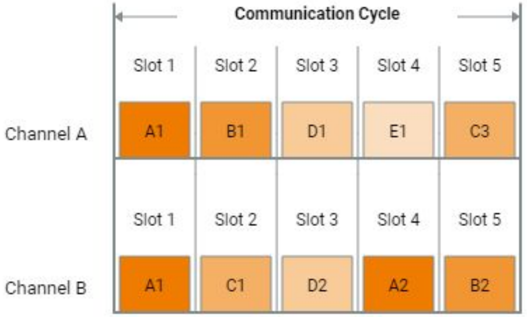


Yrkes Akademin
Vi hjälper dig att lyckas!

FlexRay - Static Segment

- It is used for deterministic transmission messages using the TDMA method
- It is organized into a number of time slots (static slots) of fixed and equal macroticks
- Fixed assignment between slots and nodes; more than one slot can be used by a node
- Nodes transmit messages in the static slots assigned to them
- Synchronized local counters are used to monitor static slot precedence
  - Each counter is incremented at the beginning of every static slot
  - The counter value refers to a specific static frame and FlexRay node
- A maximum of 1023 static slots may be defined
  - At least two sync nodes are needed in a cluster, so at least two static slots should be defined
  - There is only one FlexRay frame in a static slot

Slot	Node	Message	Channel
1	Node A	A1	A
		A1	B
2	Node B	B1	A
	Node C	C1	B
3	Node D	D1	A
		D2	B
4	Node E	E1	A
	Node A	A2	B
5	Node C	C3	A
	Node B	B2	B



FlexRay - Static Segment

- Static segment is usually used for critical messages
- Sync frames on both channels; other frames optionally 1 or 2 channels
- All static slots are repeated in order in every communication cycle
- All static time slots are consumed in a cycle whether they are used or not
  - If a node owns the current slot, and its frame is not ready/scheduled to transmit
    - A special null frame is always sent
- Slots are in order on both channels
- TDMA order by ascending frame IDs
  - A frame ID number is used by the
  - application to determine the slot number
- No delay or collision can occur

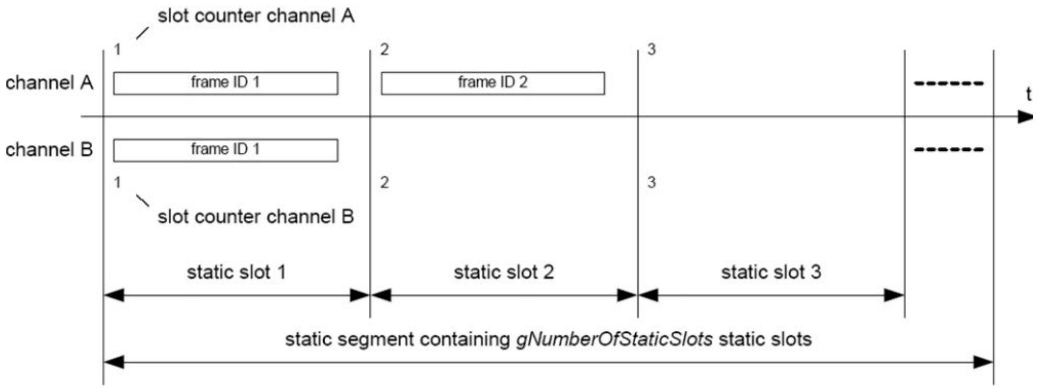
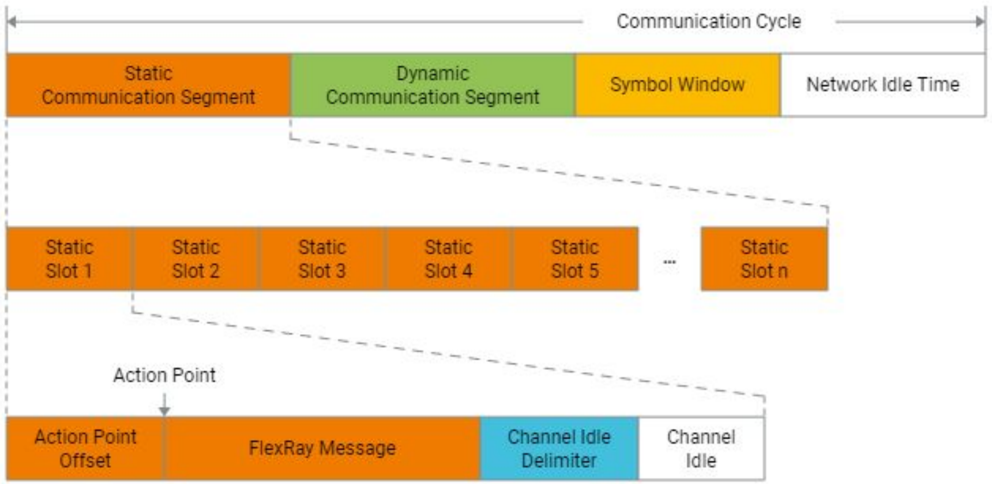


Figure 5-3: Structure of the static segment.

FlexRay - Static Slot



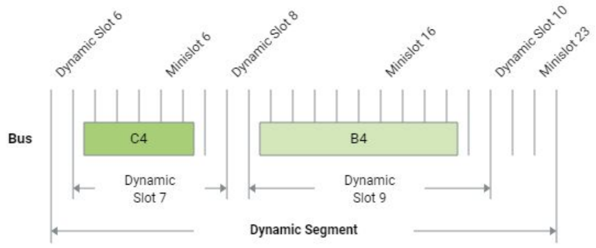
- ❖ Length of the static slots is determined by the
  - ➢ Longest FlexRay message
  - ➢ Largest transmission delay (max up to 2.5 µs)
- ❖ A static slot consists of four segments
  - ➢ Action Point Offset
    - ■ Each static slot begins with an offset
    - ■ It is a global constant for a given cluster
    - ■ It ensures that a message can be received within the given static slot, even with a maximum signal delay and FlexRay nodes with clock drifts and errors
  - ➢ FlexRay Message
  - ➢ Channel Idle Delimiter (11 recessive bits)
  - ➢ Channel Idle: It is a pause and its duration corresponds to the action point offset

FlexRay - Dynamic Segment

- ❖ Used to transmit event-triggered messages
- ❖ Always has the same duration (fixed number of macroticks) and follows the static segment
- ❖ Based on a schedule but transmission is done only if the corresponding events occur
- ❖ Splitted into fixed length minislots which are occupied by dynamic slots
  - ➢ The number of minislots is a global constant for a given cluster
- ❖ Dynamic slots are used to send dynamic messages until no space is left
- ❖ The dynamic segment uses the FTDMA method
- ❖ No fixed assigned between slots and nodes
- ❖ Length of dynamic slots can be variable
- ❖ A dynamic slot can occupy more than one minislot.
- ❖ Each minislot is assigned an index, starting from 1

Communication Schedule (Dynamic Segment)

Slot	Node	Message	Event
6	A	A4	
7	C	C4	
8	D	D3	
9	B	B4	
10	E	E3	

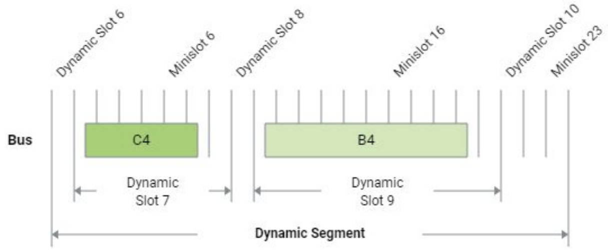


FlexRay - Dynamic Segment

- An outgoing dynamic frame (of variable size) is associated with an ID number
- The dynamic segment begins, with all nodes incrementing their local counters
- If no message matches the slot index, there is no transmission and after the duration of one minislot the index is incremented
- If there is a message matching the slot index, the message is transmitted and all the minislots occurring during the message transmission retain the same index and the slot counter is incremented only after the message transmission ends and one minislot goes by without activity (idle minislot)
- In dual channel systems, minislots are aligned, but message transmissions are not and they can occur in any minislot

Communication Schedule (Dynamic Segment)

Slot	Node	Message	Event
6	A	A4	
7	C	C4	
8	D	D3	
9	B	B4	
10	E	E3	

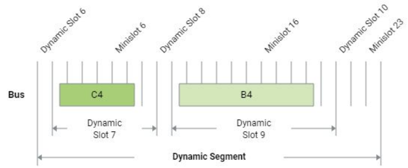


FlexRay - Dynamic Segment

- Dynamic frame ID number is used for slot numbering
  - First dynamic Frame ID = last static Frame ID + 1
- When dynamic segment time is up, unsent messages wait for next cycle
- Event-triggered messages and arbitration
  - Messages with lower Frame ID are sent first
- Each Frame ID number can only send ONE message per cycle
- As many messages as will fit in dynamic segment are sent
  - This means that only higher priority messages are sent in each cycle
- Idle minislots consume the dynamic segment bandwidth
  - But minislots are very smaller than messages
- Sync frames, startup frames, and null frames are not allowed in this segment

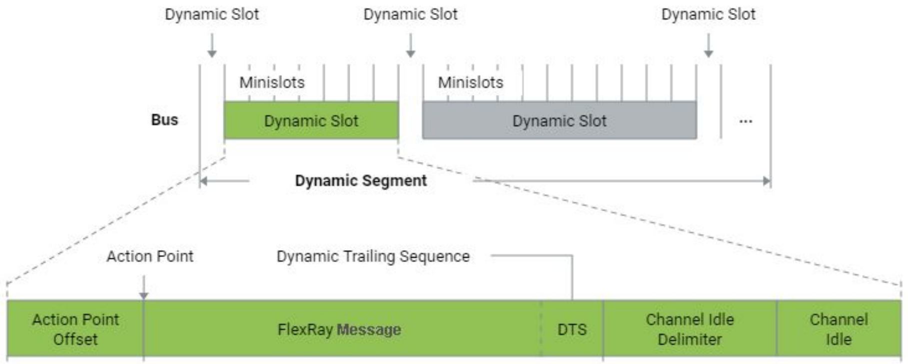
Communication Schedule (Dynamic Segment)

Slot	Node	Message	Event
6	A	A4	
7	C	C4	
8	D	D3	
9	B	B4	
10	E	E3	



FlexRay - Dynamic Slot

- Essentially, the same conditions, applied to design of a static slot, are applied to design of a static slot.
  - But messages can have different sizes
- A dynamic slot is similar to a static slot
- Each minislot contains an action point that is offset from the start of the minislot
- A frame transmission starts at the minislot action point of the first minislot of the corresponding dynamic slot and ends at a minislot action point
  - This is achieved by means of the dynamic trailing sequence (DTS)
  - Theoretically, DTS may be a maximum of one minislot in length



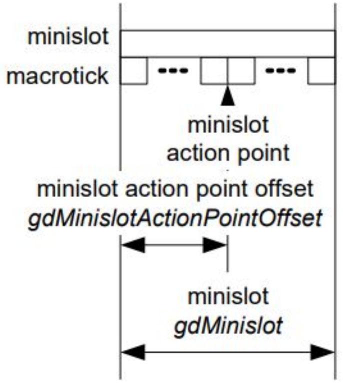


Figure 5-6: Timing within a minislot.

FlexRay - Symbol Window

- Used to transmit special bit patterns called symbols
- Symbols are used for establishing communication and network maintenance
- Flexray Symbols:
  - Collision Avoidance Symbol: Used by coldstart nodes to start the cluster up
  - Media Test Symbol: Used to test the bus guardian
  - Wake-up Symbol: Used to initiate the wake-up process
- If arbitration among multiple senders is required for the symbol window it has to be performed in the application
- The symbol window contains an action point that is offset from the start of the symbol window.
- A symbol transmission starts at the action point within the symbol window.

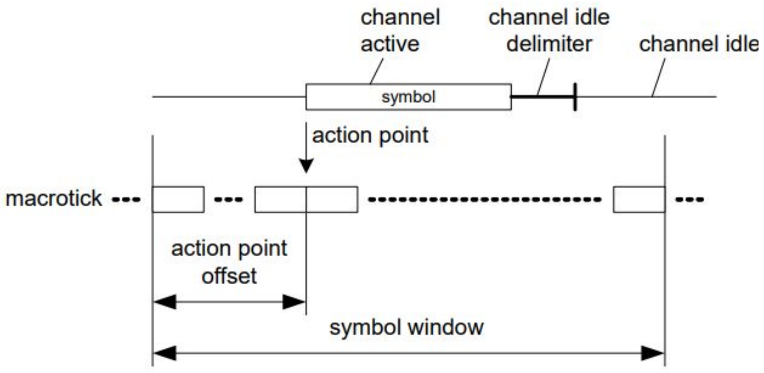
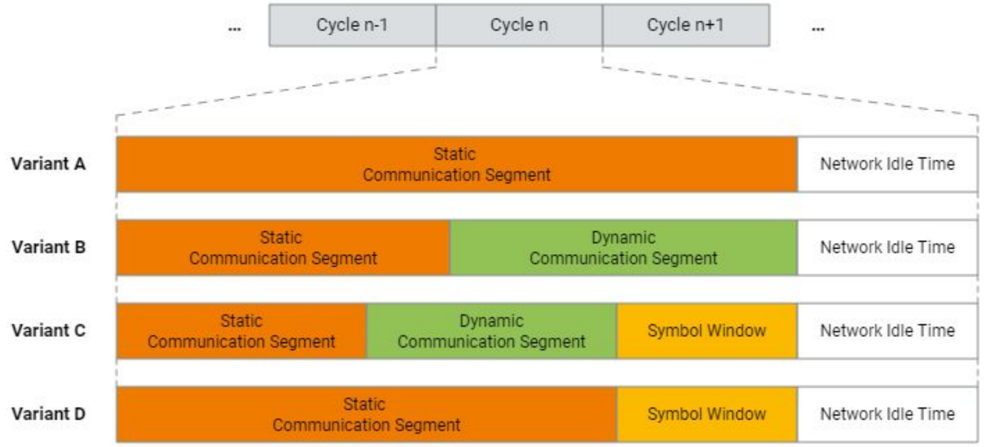


Figure 5-9: Timing within the symbol window.

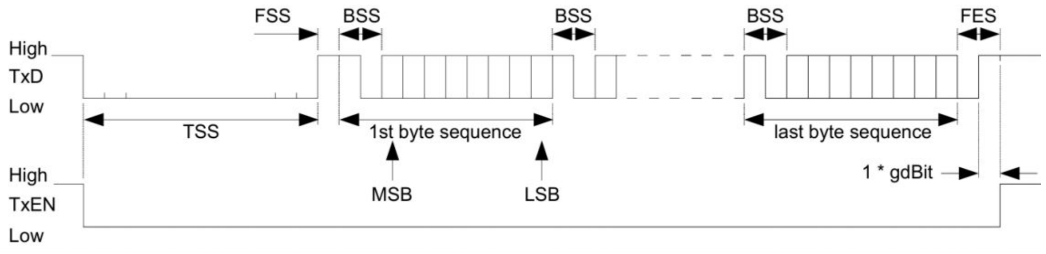
FlexRay - Network Idle Time

- It is a mandatory time segment
- Nodes calculate and apply clock correction during this segment
- It contains the remaining number of macroticks within the communication cycle that are not allocated to the static segment, dynamic segment, or symbol window
- It is a communication-free period that concludes each communication cycle

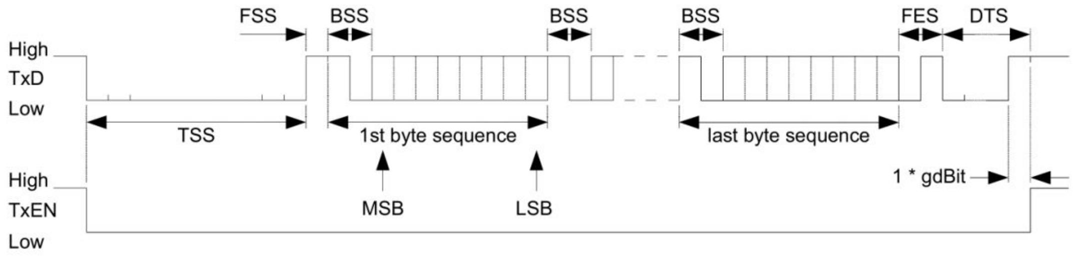


FlexRay - Frame Encoding

- Data sent as NRZ bytes
- TSS: Transmit Start Sequence (5-15 low bits)
- FSS: Frame Start Sequence (one high bit)
- BSS: Byte Start Sequence (one high bit followed by one low bit)
- FES: Frame End Sequence (one low bit followed by one high bit)
- DTS: Dynamic Trailing Sequence



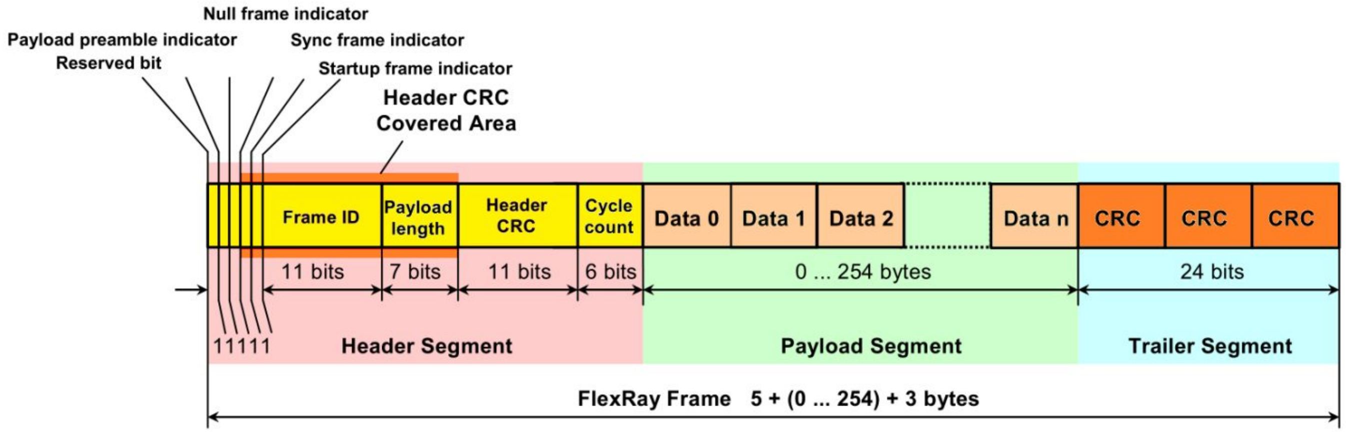
Static Frame Encoding



Dynamic Frame Encoding

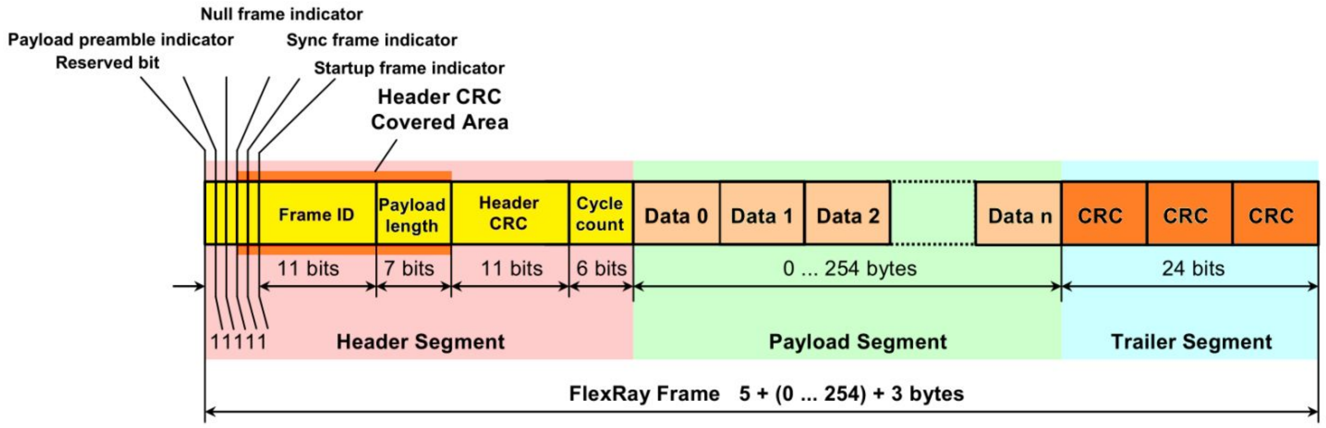
FlexRay - Message

- Each FlexRay message is composed of three parts: Header, Payload and Trailer
- The header consists of of Indicators, ID, Payload Length, CRC and Cycle Count
- The Indicator Bits: Specify the message type
  - Startup frame indicator: Indicates if the message in the static segment is a startup frame or not
  - Sync frame indicator: Indicates if message in the static segment is a sync frames or not
  - Null frame indicator: Specifies if the payload is regular or invalid
  - Payload Preamble Indicator
    - Specifies if a network management vector is sent in the payload or a message identifier (dynamic messages)



FlexRay - Message

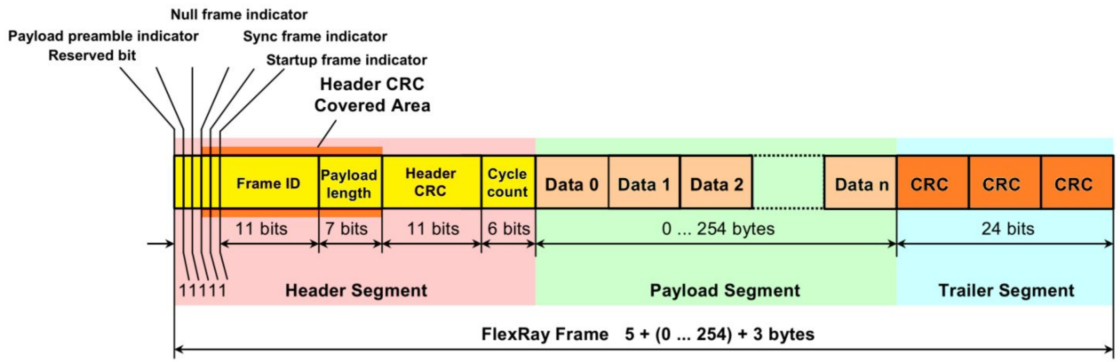
- The Frame identifier (ID)
  - It identifies a message and corresponds to a slot.
  - All IDs may be used freely. An exception is ID=0x00: this is used to identify invalid messages.
- The Payload Length
  - It shows size of the payload in words and consists of 7 bits.
  - A total of 254 bytes may be transported in a message
- The Header CRC
  - Is calculated over the identifier, payload length, sync frame indicator and startup frame indicator.



- The Cycle Counter
  - Represents the number of the cycle in which the message is sent. It always counts up to 63

FlexRay - Message Payload

- A maximum of 254 user bytes (payload) can be transported in one message
- The payload length parameter shows size of the payload in words (a word is two bytes)
- All messages in the static segment have the same payload size
- The payload size can have different values for dynamic messages
- In a static message, the first 0 to 12 bytes can be used to transmit the network management vector. This vector must be common to all nodes and it is used to implement network management in a cluster.
- In a dynamic message, the first two bytes can be the message identifier.
- The trailer CRC sequence is computed over the header and payload



FlexRay - Clock Synchronization

- All communication cycles should begin at the same time and are of equal length
- The synchronization process makes sure that length of the macroticks in each node is the same and they start at the same point of time in all nodes.
- Synchronization of the local clock of a node is done against an virtual global clock
- The virtual reference clock is derived from the sync frames of the sync nodes
  - The fault-tolerant midpoint algorithm is used for clock synchronization (FlexRay 3.0.1, section 8.6.1)
- A FlexRay network has 2 up to 15 sync nodes, which transmit sync frames
- On reception of each sync message, a node compares its clock with the sync node clock and makes any changes needed to synchronize.
- FlexRay nodes regularly correct the offset and rate of their local clocks
- Clock synchronization is applied during Network Idle Time (NIT)

FlexRay

- ❖ Bit Sampling
  - ➢ The receiver samples the data line 8 times per bit
  - ➢ The receiver uses the majority value over five consecutive samples
  - ➢ If the majority value is “0”, the bit value is a zero, otherwise a one
- ❖ Application
  - ➢ To maintain network configurations between nodes
    - ■ The FlexRay consortium provides a FIBEX (Field Bus Exchange Format) file format to store and transfer the FlexRay network parameters in order to configure the ECUs during designing, testing and etc.
    - ■ FIBEX is an XML-based file format defined by ASAM (Association for Standardisation of Automation and Measuring) which is compatible with many different automotive protocols and is used for describing automotive networks
    - ■ It makes writing code for FlexRay networks easier; name of the signals is referred in the code.

FlexRay

- Some useful links
  - Understanding FlexRay
  - FlexRay Configuration Overview NXP
  - Elektrobit FlexRay: A first glance - HQ
  - FlexRay
  - FlexRay Automotive Communication Bus Overview
  - Flexray Protocol in Automotive Network Communications