# The FlexRay Bus

**Raj Rajkumar**
Lecture #18



## Outline

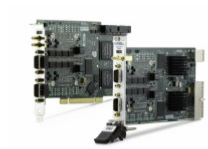
- FlexRay Basics
- FlexRay Topology and Layout
- The FlexRay Protocol
- The FlexRay Communication Cycle
- Signals
- Clock Synchronization and Cold Starting
- In-Cycle Control
- FIBEX: The FlexRay Network Database

## Communication Demands in Automobiles

- Automobiles continue to improve safety, increase performance, reduce environmental impact, and enhance comfort, the speed, quantity and reliability of data communicated between a car's electronic control units (ECU) must increase.
- Advanced control and safety systems, combining multiple sensors, actuators and electronic control units, require synchronization and performance past what CAN can provide.
- The FlexRay network standard has emerged as a new in-vehicle communications bus.



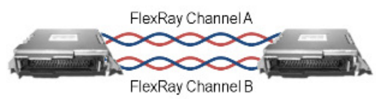
## FlexRay Positioning



| Bus | LIN | CAN | FlexRay |
|---|---|---|---|
| Speed | 40 kbit/s | 1 Mbit/s | 10 Mbit/s |
| Cost | $ | $$ | $$$ |
| Wires | 1 | 2 | 2 or 4 |
| Typical Applications | Body Electronics (Mirrors, Power Seats, Accesories) | Powertrain (Engine, Transmission, ABS) | High-Performance Powertrain, Safety (Drive-by-wire, active suspension, adaptive cruise control) |

## FlexRay Basics

- Uses **unshielded twisted pair** cabling
  - Differential signaling on each pair of wires reduces the effects of external noise on the network without expensive shielding.
  - Most FlexRay nodes typically also have power and ground wires to power transceivers and microprocessors.
- Supports single- and dual-channel configurations which consist of one or two pairs of wires respectively.
- Dual-channel configurations offer enhanced fault-tolerance and/or increased bandwidth.
- Most first-generation FlexRay networks only utilize one channel to keep wiring costs down.
- FlexRay buses require **termination** at the ends.



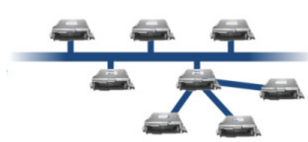



## FlexRay Topology and Layout

FlexRay can be used in any of the following configurations:

- Multi-Drop Bus

- Star Network



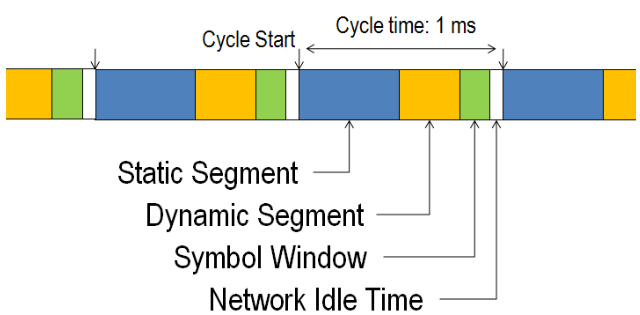
- Hybrid Network

## The FlexRay Protocol

- Supports unique time-triggered protocol that provides options for
  - deterministic data that arrives in a predictable time frame, and
  - CAN-like dynamic event-driven data.
- FlexRay manages multiple access with a **Time Division Multiple Access** or TDMA scheme.
  - Every FlexRay node is synchronized to the same clock, and each node waits for its turn to write on the bus.
- For a TDMA network to work correctly, **all** nodes on the network must be configured correctly.

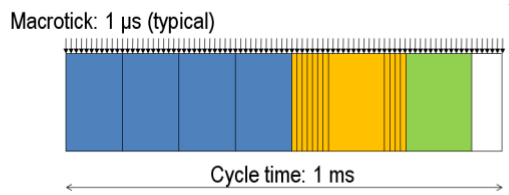


## The FlexRay Communication Cycle




- **Static Segment:** Reserved slots for deterministic data arriving at a fixed period.
- **Dynamic Segment:** The dynamic segment behaves in a fashion similar to CAN and is used for a variety of event-based data that does “not” require determinism.
- **Symbol Window**: Typically used for network maintenance and signaling for starting the network.
- **Network Idle Time:** A known "quiet" time used to maintain synchronization between node clocks.

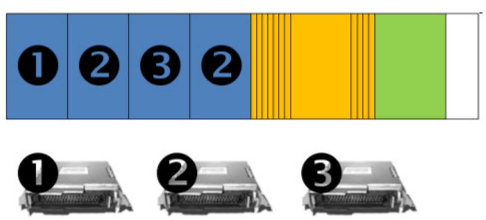
## The FlexRay macrotick




- **Macrotick** is the smallest practical unit of time on a FlexRay network
- FlexRay controllers actively synchronize themselves and adjust their local clocks
  - Macro-tick occurs at the same point in time on every node across the network.
- While configurable, macro-ticks are normally 1 µs long.
- Because the macro-tick is synchronized, data that rely on it are also synchronized.



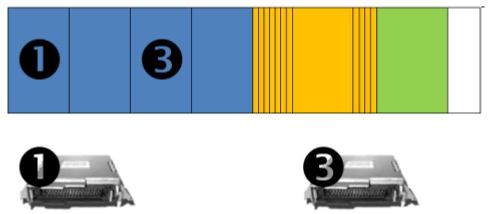
## The Static Segment



- The static segment shown in blue is that portion of the cycle dedicated to scheduling a number of time-triggered frames.
- The static segment is broken up into slots, each slot containing a reserved frame of data.
- At each slot, the corresponding ECU can transmit its data into that slot.
- Once that slot passes, the ECU must wait until the next cycle to use that slot.
- Due to clock synchronization, data transmission is deterministic and programs know exactly how old the data are.
- FlexRay networks may contain up to several dozen static slots.

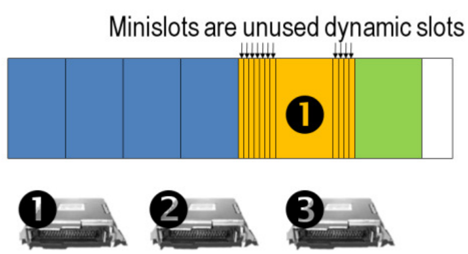
## Static Segments and ‘Missing’ ECUs

- If an ECU goes offline or decides not to transmit data, its slot remains open and is not used by any other ECU





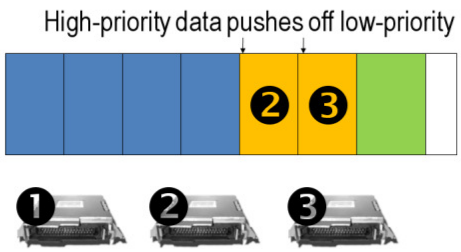

## The Dynamic Segment




ECU #1 broadcasts in its mini-slot since the first 7 mini-slots chose not to broadcast.

- The dynamic segment allows “occasionally” transmitted data.
  - The segment is a fixed length.
  - To prioritize the data, **mini-slots** are pre-assigned to each frame of data that is eligible for transmission in the dynamic segment. A mini-slot is typically a **macro-tick** long.
  - Higher priority data receive a mini-slot closer to the beginning of the dynamic frame.
- Once a mini-slot occurs, an ECU has a brief opportunity to broadcast its frame.
- If it does not broadcast, it loses its spot in the dynamic frame and the next mini-slot occurs. This process repeats until an ECU elects to broadcast data.
- If data is broadcast, future mini-slots must wait until transmission is complete.
- If the dynamic window ends, lower-priority mini-slots must wait for the next cycle.

## Priorities in the Dynamic Segment




ECUs 2 and 3 broadcast in their mini-slots and leave no time for the lower-priority mini-slots.



## The "Symbol" Window

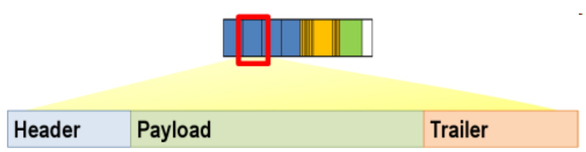
- Used for maintenance and identification of special cycles such as cold-start cycles.
- Most high-level applications do not interact with the symbol window.

## Network Idle Time

- The *network idle* time is of a pre-defined, known length by ECUs.
- The ECUs use this idle time to adjust for any drift that may have occurred during the previous cycle.

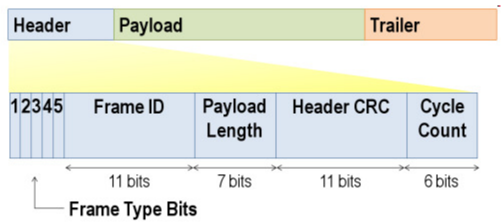


## The FlexRay "Frame"




- Each slot of a static or dynamic segment contains a FlexRay "Frame".
- A frame is divided into three segments: **Header**, **Payload**, and **Trailer**.

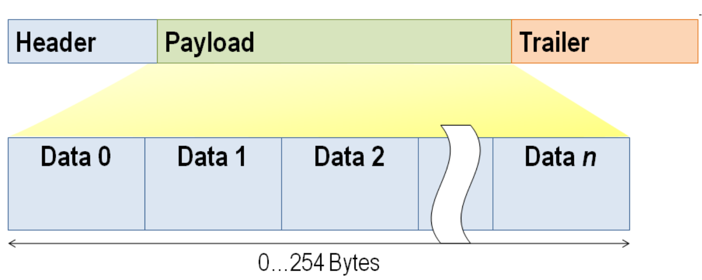
## The Frame Header




- The header is 5 bytes (40 bits) long and includes the following fields:
  - Status Bits (5 bits)
  - Frame ID (11 bits): defines the slot in which the frame should be transmitted and is used for prioritizing event-triggered frames
  - Payload Length (7 bits): # of words transferred in the frame
  - Header CRC (11 bits): detect errors during the transfer
  - Cycle Count (6 bits): contains the value of a counter that advances incrementally each time a Communication Cycle starts.

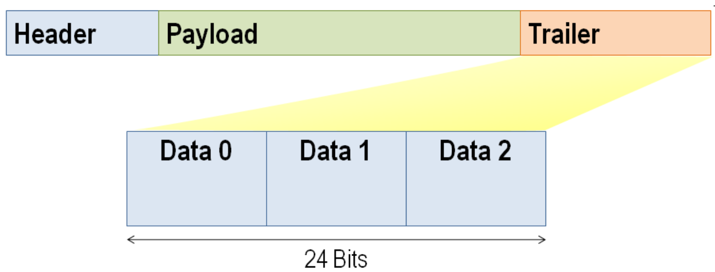


## The Payload




- The payload contains the actual data transferred by the frame.
- The length of the FlexRay payload or data frame is up to 127 words (254 bytes)
  - Over 30 times greater compared to CAN.

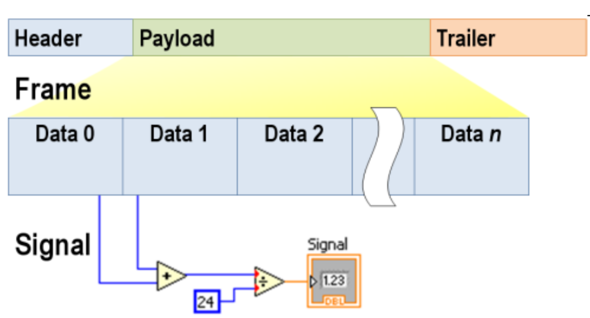
## The Trailer




- The trailer contains three 8-bit CRCs to detect transmission errors.

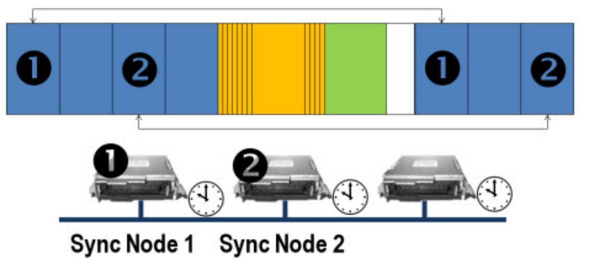


## Signals




- Most applications require data to be represented in real decimal values with units, scaling, and limits.
- Take one or more bits or bytes from a FlexRay frame, apply a scaling and offset to get a **signal**.
- Most ECU programs work with FlexRay data as signals
  - Convert signals to raw frame data using driver or lower-level communication protocols.
- A typical vehicle has hundreds to thousands of signals.
  - Scaling, offset, definitions, and locations of signals can change
  - FlexRay networks store these definitions in a FIBEX database.
  - Makes writing programs for FlexRay networks easier: refer to signal name in code.

## Clock Synchronization and Cold Start



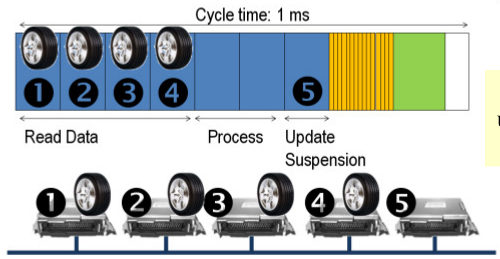

- FlexRay can sync nodes on a network <u>without</u> an external synchronization clock.
  - Uses 2 special types of frames: **Startup Frames** and **Sync Frames**.
  - To start a FlexRay cluster, at least 2 different nodes are required to send startup frames.
    - The action of starting up the FlexRay bus is known as a **cold-start** and the nodes sending the startup frames are usually known as **cold-start nodes**.
    - The startup frames are analogous to a start trigger, which tells all the nodes on the network to start.



## Clock Synchronization (cont'd)

- Once the network is started, all nodes must synchronize their internal oscillators to the network's macro-tick.
  - Two separate nodes called **synchronization nodes** are pre-designated to broadcast special sync frames when they are first turned on.
  - Other nodes wait for the sync frames to be broadcast, and measure the time between successive broadcasts in order to calibrate their internal clocks to the FlexRay time.
  - The sync frames are designated in the FIBEX configuration for the network.
- Once the network is synchronized and on-line, the network idle time ("white space" in the pictures) is measured and used to adjust the clocks from cycle-to-cycle to maintain tight synchronization.

# In-Cycle Control



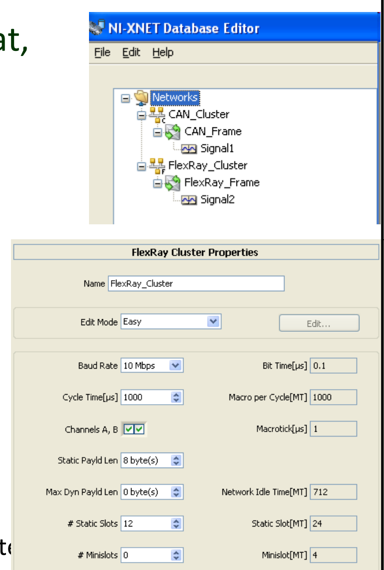

Reading 4 wheel positions and updating a vehicle control output in a single FlexRay cycle

- In the figure, wheel positions are read before the final update command from the central controller #5
  - The controller has time to process and make a rapid output within the *same* communication cycle.
  - Allows very high-speed control rates to be realized on a FlexRay network.



# FIBEX - The FlexRay Network Database

- FIeld Bus EXchange (FIBEX) format, an XML-based standardized file format is used for describing automotive networks.
- FIBEX contains many aspects of a particular network including:
  - Transmit and receive schedules
  - Frame definitions
  - Signal definitions
  - Bit-level encoding of signals
  - Network topology
  - ECU information
  - Network configurations, including baud rate and timings



## Conclusions

- The FlexRay communications network delivers deterministic, fault-tolerant and high-speed bus system performance requirements for next-generation automobiles.