Solutions for FlexRay™
Agenda

> FlexRay Solutions
  Design of Distributed Systems
  ECU Test, Simulation, and Analysis
  ECU Calibration
  ECU Software
  Network Interfaces
  Summary
  FlexRay Main Characteristics
  FlexRay Communication Structure
  Startup
  FIBEX Explorer
  PDU Support
  Analysis
  Stimulation
<table>
<thead>
<tr>
<th>Design of Distributed Systems</th>
<th>ECU Test, Simulation, Analysis</th>
<th>ECU Calibration</th>
<th>ECU Software</th>
<th>Network Interfaces</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network Designer FlexRay</td>
<td>CANoe.FlexRay</td>
<td>CANape</td>
<td>MICROsAR FlexRay: FR Interface</td>
<td>VN3300</td>
</tr>
<tr>
<td>FIBEX Explorer pro</td>
<td>CANAnalyzer.FlexRay</td>
<td>Embedded Module</td>
<td>FR State Manager</td>
<td>VN3600</td>
</tr>
<tr>
<td></td>
<td>FRstress</td>
<td>XCP-on-FlexRay</td>
<td>FR TP / FR ISOTP</td>
<td>VN7600</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>FR NM</td>
<td>FlexCard Cyclone II SE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>FR XCP</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>MICROsAR CAL: FR Driver</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>MICROsAR COM: COM Manager</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>PDU Router</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>osCAN, MICROsAR OS</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>XCP-on-FlexRay</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Flash Bootloader</td>
<td></td>
</tr>
</tbody>
</table>

Application and Integration Services, Training
Agenda

FlexRay Solutions

> Design of Distributed Systems
  ECU Test, Simulation, and Analysis
  ECU Calibration
  ECU Software
  Network Interfaces
  Summary
  FlexRay Main Characteristics
  FlexRay Communication Structure
  Startup
  FIBEX Explorer
  PDU Support
  Analysis
  Stimulation
Easy designing of a FlexRay communication network
- Define a cluster with one or two channels and the ECU topology
- Define the signal database
- Define FlexRay specific frames and their properties
- Define Tx/Rx Frame relations per ECU
- Define static and dynamic schedule
- Specify global and local FlexRay configuration parameters (e.g. cycle length, number of slots, slot duration, ...)
- Minor changes to the architecture and communication schedule are possible without the need to completely reschedule

Support of FIBEX version 1.1.5, 1.2, and 2.0
- Consistency check for FIBEX format and FlexRay settings

Configuration management based on Vector eASEE or other configuration management systems
Agenda

FlexRay Solutions
Design of Distributed Systems

ECU Test, Simulation, and Analysis

ECU Calibration
ECU Software
Network Interfaces
Summary
FlexRay Main Characteristics
FlexRay Communication Structure
Startup
FIBEX Explorer
PDU Support
Analysis
Stimulation
ECU Test, Simulation, and Analysis

CANoe.FlexRay

- **Measurement and analysis**
  - Trace window for messages and bus events
  - Data window
  - Graphic window for plotting of signals
  - Message statistic and bus statistic window
  - Logging with triggers
  - Filter
  - Offline mode (replay of a log)
  - GUI extensions for user friendly input and output using panel windows
  - Import of network data definitions and network interface controller configuration from the FIBEX database

- **Simulation, stimulation, and test**
  - CAPL modeling language
  - Total bus simulation
  - Remaining bus simulation
  - Gateway functionality with CAN, LIN, MOST, FlexRay, J1587, etc.
**ECU Test, Simulation, and Analysis**

**Mid-Size HIL Environment**

**CANoe RT System**

- **CANoe RT User PC**
  - Front end for CANoe GUI
  - Host for logging data and test results
- **CANoe RT Server**
  - Real-time execution server for:
    - remaining-bus and environment simulation
    - test script execution

**CANoe RT User PC**

<table>
<thead>
<tr>
<th>CANoe RT Server Platforms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compact Real-Time Box</td>
</tr>
<tr>
<td>Real-Time Rack</td>
</tr>
</tbody>
</table>

- CAN
- LIN
- MOST
- FlexRay

**Digital I/O**

**Analog I/O**

**ECU**
**Objectives:**

- Identify network/ECU behavior caused by physical bus failures
- Identify network/ECU behavior caused by disturbing and manipulating FlexRay frames

**Features:**

- Triggering by specific values of frame fields or external trigger input and output
- Configuration of bit stream disturbances or manipulations
- Operation as: Normal node, star coupler, trigger engine

---

**Example Triggering:**

- FRstress

**Trigger Output:**

```
... e. g. Frame ID ... 0011001001101
                      00110001100001 ... Trailer ...
```
Agenda

FlexRay Solutions
Design of Distributed Systems
ECU Test, Simulation, and Analysis

> **ECU Calibration**

  ECU Software
  Network Interfaces
  Summary
  FlexRay Main Characteristics
  FlexRay Communication Structure
  Startup
  FIBEX Explorer
  PDU Support
  Analysis
  Stimulation
ECU Calibration

- Measurement and Calibration of ECU parameters
- Time-synchronous data acquisition and visualization of:
  - Internal ECU data
  - Signals from CAN, LIN, FlexRay bus
  - Signals from external measuring equipment
- Video and Audio measurement for verification of HMI devices or driver assistant systems
- Online calibration via CCP/XCP and real-time stimulation via XCP
- Offline calibration
- Seamlessly integrated diagnostics via KWP2000 and UDS
Agenda

FlexRay Solutions
Design of Distributed Systems
ECU Test, Simulation, and Analysis
ECU Calibration

> ECU Software
Network Interfaces
Summary
FlexRay Main Characteristics
FlexRay Communication Structure
Startup
FIBEX Explorer
PDU Support
Analysis
Stimulation
ECU Software

FlexRay Embedded Software Components

- Configuration by AUTOSAR ECU Configuration Description or FIBEX with GENy (included in DaVinci Configurator Pro)
- Operating system independent (OSEK, AUTOSAR SC1-4, or TT OS)
- All modules are available based on AUTOSAR 3.x
- Support of Development Error Tracer
- Support of Diagnostic Event Manager
- FlexRay Timer Services
- Cycle Multiplexing
- In-Cycle Repetition
- Rx and Tx Interrupt Mode
- Rx indication and timeouts, Tx confirmation
- Small and runtime efficient implementation
- Gateway function to CAN, LIN, FR realized through MICROSAR COM (PDUR, COM)

© 2011. Vector Informatik GmbH. All rights reserved. Any distribution or copying is subject to prior written approval by Vector.
Agenda

FlexRay Solutions
Design of Distributed Systems
ECU Test, Simulation, and Analysis
ECU Calibration
ECU Software

> Network Interfaces

Summary
FlexRay Main Characteristics
FlexRay Communication Structure
Startup
FIBEX Explorer
PDU Support
Analysis
Stimulation
VN-Family and FlexCard

- FlexRay Controller: Bosch E-Ray
- Physical Layer Transceivers: NXP TJA1080
- Transmit and receive data and null frames
- Detect invalid frames
- Support cycle multiplexing, In-cycle response
- Support 254 byte maximum payload
- Autonomous network start
- Startup and asynchronous monitoring
- Time synchronization with Vector interfaces
- Configurable trigger output
- Driver library for own applications
RT Server Platform as Extended CANoe Interface

- Typical use case: CAN, LIN and FlexRay simulation with high real-time requirements
- Recommended for FlexRay simulations
- Dedicated interface for remaining bus and mid-size HIL simulations
- Deterministic execution platform for simulations
- Very small jitters
- Fast boot-up time (10 seconds)
- Ethernet connection to host PC running CANoe
- Competitive pricing due to the use of standard products
- Can be ordered as fully pre-configured system from Vector (including operating system)
Network Interfaces

VN8900 as Real Time Execution Platform

- CANoe/CANalyzer GUI
- USB
- VN8900
- USB Master Ethernet
- CAN1 … CAN4
- LIN 1 … LIN 4
- I/O Digital Analog
- 4 Network Interfaces
- ECU
- Intel Atom CPU-Board
- Test Execution
- Network Simulation
- CAPL Execution
- Logging

VN8900 as Real Time Execution Platform

© 2011. Vector Informatik GmbH. All rights reserved. Any distribution or copying is subject to prior written approval by Vector.
Agenda

- FlexRay Solutions
- Design of Distributed Systems
- ECU Test, Simulation, and Analysis
- ECU Calibration
- ECU Software
- Network Interfaces

**Summary**

- FlexRay Main Characteristics
- FlexRay Communication Structure
- Startup
- FIBEX Explorer
- PDU Support
- Analysis
- Stimulation
Summary

- FlexRay solutions are available for:
  - Network design
  - Analysis, simulation, stimulation, and test
  - Bus stress
  - Measurement and calibration
  - Embedded software
  - Network interfaces

- CANoe.FlexRay provides a comprehensive FlexRay development and analysis solution

- Support of the FIBEX format offers a scalable solution for interfacing different development tools

Vector – Your competent FlexRay partner!
FlexRay

Short Introduction
Agenda

FlexRay Solutions
Design of Distributed Systems
ECU Test, Simulation, and Analysis
ECU Calibration
ECU Software
Network Interfaces
Summary

> FlexRay Main Characteristics

FlexRay Communication Structure
Startup
FIBEX Explorer
PDU Support
Analysis
Stimulation
FlexRay Main Characteristics

Activation

- Event driven
  - System services are activated in response to events
  - Serial bus system:
    - Bus access based on incoming events
    - Bus access if signal changes

- Time driven
  - System services are activated at predefined times based on a (real-time) clock
  - Serial bus system:
    - Bus access is periodic based on the schedule
    - No signal change := “Nullframe”
FlexRay Main Characteristics

Example: Personal transportation

- **Event driven**
  - Individualized departure time
  - Departure time and therefore trip duration unknown
  - Trip might not even be completed

- **Time driven**
  - Periodic departure time
  - Departure time and trip duration known
  - Reliable
Agenda

FlexRay Solutions
Design of Distributed Systems
ECU Test, Simulation, and Analysis
ECU Calibration
ECU Software
Network Interfaces
Summary
FlexRay Main Characteristics

> **FlexRay Communication Structure**

  Startup
  FIBEX Explorer
  PDU Support
  Analysis
  Stimulation
Data transmission with FlexRay:
- Time-driven, organized in cycles
- Cycle time is constant – Global clock
- Cycles are subdivided into segments
- Segments repeat themselves at equidistant time intervals

Communication Cycle consists of:
- Static Segment
- Dynamic Segment (optional)
- Symbol Window (optional)
- NIT: Network Idle Time

PS 2.1: cCycleCountMax := 63
FlexRay Communication Structure

Static Slots

- Characteristics of the Static Slots:
  - Continuous flow
  - All static slots have same length
  - Same number of slots in each cycle: `gNumberOfStaticSlots`
  - Only one node may send per slot
  - Slot IDs have unique assignments to TX nodes
  - At least 2 static slots necessary for synchronization

- Static Segment:
  - TDMA bus access
  - Deterministic time behavior
  - Time window for real-time applications & safety-critical applications

`PS 2.1: cStaticSlotIDMax := 1023`
Characteristics of Dynamic Slots:

- Minislot: « Placeholder »
- Dynamic Slot IDs are allocated to TX nodes
- Same number of MiniSlots in each cycle: $g\text{NumberOfMiniSlots}$
- Transmission as needed, then Minislot becomes a dynamic slot
- Dynamic slots can have different lengths

Dynamic Segment:

- FTDMA bus access
- « Arbitration » based on the Minislot ID
- Time window for event-driven data transfer

PS 2.1: $c\text{SlotIDMax} := 2047$
FlexRay Communication Structure

Static Frames

Characteristics of Static Frames:

- Frame is transmitted in the Static Slot
- Static slot duration is declared in the global parameter: `gdStaticSlot`
- Header := Identifier + Suppl. info
- Payload := Actual useful data
- Trailer := Data protection
- CID := Channel Idle Delimiter
- Reliable start of transmission at the Action Point, global parameter `gdActionPointOffset`
Characteristics of Dynamic Frames:

- Frame is transmitted in the Dynamic Slot
- Layout of Header, Payload, Trailer and CID is identical to that of Static Frame
- Payload length is variable
- DTS:= Dynamic Trailing Sequence
- Possible to start transmission at MiniSlot Action Point, parameter \( gdMiniSlotActionPointOffset \)
FlexRay Communication Structure
Overview: Static - Dynamic

FlexRay Communication Structure
Symbol Window

- Characteristics of the Symbol Window:
  - Optional
  - MTS := Media Access Test Symbol
    - Length is defined by \( cdCAS := 30 \)
    - Test of the Bus Guardian
  - Only one symbol per cycle, therefore no arbitration
  - Possible to start transmission at Action Point, parameter \( gdActionPointOffset \)
- Characteristics of the NIT «Network Idle Time»:
  - Time period for synchronization
  - Defined time slot at end of a cycle, time duration \( gdNIT \)
  - Channel Idle := No data on the bus, available time for calculating synchronization parameters
  - Offset Correction Segment := Used for synchronization.
Agenda

FlexRay Solutions
Design of Distributed Systems
ECU Test, Simulation, and Analysis
ECU Calibration
ECU Software
Network Interfaces
Summary
FlexRay Main Characteristics
FlexRay Communication Structure

> Startup
  FIBEX Explorer
  PDU Support
  Analysis
  Stimulation
State change:
- State before startup: All nodes are in Power-On and in Ready State (awake)
- State after startup: All nodes are Power-On and in Normal Active State
- « Creation of the common time base » for all nodes utilizing at least two Sync nodes

Characteristics of the Startup:
- At least two coldstart nodes are necessary
  - « Leading Coldstart Node »: Initiates setup, prescribes schedule
  - « Following Coldstart Node »: Synchronizes itself to Leading CSN
- Startup is simultaneously on both channels
- Leading Coldstart Node sends « CAS » Collision Avoidance Symbol
- After a specified time the Leading Coldstart Node sends its first Startup Frame (SyncFrame) -> Start of Cycle
- After a minimum of 4 cycles the Following Coldstart Node may also send its Sync Frames
**Startup Procedure**

- **Leading ColdStart Node:**
  - Host puts CC in Startup
  - POC goes to Coldstart Listen State
  - CC sends CAS Symbol
  - CC sends its Sync frames over 4 cycles

- **Following ColdStart Node:**
  - CCs detect Sync frames
  - Attempt to synchronize during the 4 cycles
  - Send their own Sync frames for 5-7 cycles

Source: FlexRay PS2.1
Node A: Leading Coldstart Node
Node B: Following Coldstart Node
Node C: Integrating, Non Coldstart Node

Legend:
- CAS: CAS symbol
- S: startup frame of node A
- B: startup frame of node B
- C: frame of node C

Source: FlexRay PS2.1
CANoe & CANalyzer.FlexRay

Features
Agenda

- FlexRay Solutions
- Design of Distributed Systems
- ECU Test, Simulation, and Analysis
- ECU Calibration
- ECU Software
- Network Interfaces
- Summary
- FlexRay Main Characteristics
- FlexRay Communication Structure
- Startup
  - **FIBEX Explorer**
    - PDU Support
    - Analysis
    - Stimulation
FIBEX Explorer

- User friendly visualization of FIBEX and FIBEX+ databases
- Detailed views for communication (Frames/PDUs, RX/TX relations, Signals, and ECUs), scheduling, and network parameters

© 2011. Vector Informatik GmbH. All rights reserved. Any distribution or copying is subject to prior written approval by Vector.

Slide: 39
PDU Support
CANoe’s AUTOSAR Stack

- Semantic interpretation controlled by PDU Layer
- FIBEX or AUTOSAR System Description used as database
Detailed analysis of the communication on network, node and frame level

Automatic check based on the expected communication defined in the database

Example: The ECU CANoe_Monitor does not send the expected number of frames.
Analysis

FlexRay Filter Block

- Filter functions for Frames, PDUs, ECUs, Status and Error Events
- Overview section
- Comfortable grouping possibility of its filter rules
Analysis

FlexRay Statistics Monitor

- Displays FlexRay-specific statistics on network and node level
- Statistics for Frames and PDUs
- Detailed statistics for protocol errors
Easy configuration of Tx frames considering all FlexRay specific elements
- Simplified data update without any CAPL program code
Easy configuration of Tx PDUs considering all FlexRay PDU specific elements (Update Bit, Update Counter,...) using FIBEX 3.0 or FIBEX+. Simplifies data update without any CAPL program code.
Stimulation
Replay Block

- FlexRay Replay Block
- Insertion of certain data sequence
- Provides the counter part of a real ECU
- Support of PDUs and Frames
Stimulation

Signal Generators

- Signal generator for FlexRay signals
- Simple generation of signal courses (ramp, sine, ...)
- Synchronized to the FlexRay schedule
Simulation

Configuration

- Automatic registration of TX frames of simulated nodes
- Activation via ECU assignment in the simulation setup
- Check and modification of the automatic registered TX frames
  - Additional manual deactivation of not needed TX frames to save TX buffers in the CC
  - One click modify of Startup/Sync flag for all frames
  - Simply simulate complete load of static segment when using database import wizard with creation of simulation nodes!

NEW 6.1
Execution of CAPL programs on the VN3x00 & VN7600 network interfaces

The CAPL code of the selected nodes will be transferred to the network interface during measurement preparation and executed on the modules

Use case:

Simulation of time critical tasks

Goal:

Deterministic execution of the CAPL programs

Reduced latency for fast responses
# Test Report

## Test Overview

- **Test begin**: 2008-12-11 13:16:59 (logging timestamp 0.791760)
- **Test end**: 2008-12-11 13:19:49 (logging timestamp 51.026500)

### Statistics

<table>
<thead>
<tr>
<th>Type</th>
<th>Count</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overall number of test cases</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>Executed test cases</td>
<td>13</td>
<td>100%</td>
</tr>
<tr>
<td>Not-executed test cases</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>Test cases passed</td>
<td>13</td>
<td>100%</td>
</tr>
<tr>
<td>Test cases failed</td>
<td>0</td>
<td>0%</td>
</tr>
</tbody>
</table>

## Test Case Results

1. **Test static requirements of the SUT**

   **Test Case TC101**: Lock statically: Passed
   - **Lock the car while engine off or on**
   - **Test begin**: 2008-12-11 13:18:59 (logging timestamp 0.969487)
   - **Test end**: 2008-12-11 13:19:00 (logging timestamp 2.269487)

### Test Case Sequence

<table>
<thead>
<tr>
<th>Timestamp</th>
<th>Test Step</th>
<th>Description</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.009467</td>
<td>Test pattern begin</td>
<td>Test pattern begin</td>
<td>Pass</td>
</tr>
<tr>
<td>0.009467</td>
<td>Set specified value</td>
<td>CAN signal 'Velocity' on bus CAN 0</td>
<td>Pass</td>
</tr>
<tr>
<td>0.069467</td>
<td>Set specified value</td>
<td>CAN signal 'CrashDetected' on bus CAN 0</td>
<td>Pass</td>
</tr>
<tr>
<td>0.069467</td>
<td>Set specified value</td>
<td>CAN signal 'LockRequest' on bus CAN 0</td>
<td>Pass</td>
</tr>
<tr>
<td>0.069467</td>
<td>Set specified value</td>
<td>CAN signal 'EngineRunning' on bus CAN 0</td>
<td>Pass</td>
</tr>
<tr>
<td>0.069467</td>
<td>Stimulation of the input parameters</td>
<td>1</td>
<td>Pass</td>
</tr>
<tr>
<td>1.069467</td>
<td>Waited for 100 ms</td>
<td>2</td>
<td>Pass</td>
</tr>
<tr>
<td>1.069467</td>
<td>Test pattern end</td>
<td>Test pattern end</td>
<td>Pass</td>
</tr>
</tbody>
</table>

2. **Test signals of Windows system: Passed**

   **Test Case TC09**: Test pattern begin
   - **Set specified value**
   - **CAN signal 'KeyUp' on bus CAN 0**
   - **Set specified value**
   - **CAN signal 'KeyDown' on bus CAN 0**
   - **Stimulation of the input parameters**
   - **Waited for 100 ms**
Diagnostics over FlexRay

Overview

- Support of Fault Memory Window and Diagnostics Console
- Send and receive diagnostics requests and responses in simulations
- Support of TFS functionality, XML test patterns
- Diagnostic Observer for Trace window. Support of diagnostic parameter in Data and Graphics window
- CAPL callback interface for ECU simulations, TP layer access for tests (e.g. DiVa)
- Trace window FlexRay TP Observer for AUTOSAR FlexRay TP, BMW TP & ISO 10681-2 TP
Interactively browse & access available diagnostic services

Cyclical update of fault memory contents / logging of DTCs for external processing

One separate window per real / simulated network node

No code development necessary

→ CDD/ODX file for ECU / network sufficient
Diagnostics over FlexRay

Diagnostic Observer & Symbolic Output

- Pure observer for monitoring diagnostic traffic on existing networks → no adverse influence on network
- Fallback option to KWP2000 generic interpretation → starting without specific diagnostic definition possible
- Symbolic presentation of traffic according to diagnostic description files → “speaking” service names & “meaningful” parameters
- Output to Trace, Graphics & Data windows
## Add-on Packages

### Availability of Add-in Concept for OEM Modules on FlexRay

<table>
<thead>
<tr>
<th>FlexRay Interaction Layer</th>
<th>OEM</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>AudiFrIL</td>
<td>Audi</td>
<td>Available</td>
</tr>
<tr>
<td>BMWFrIL</td>
<td>BMW</td>
<td>Available</td>
</tr>
<tr>
<td>DAGFrIL</td>
<td>Daimler</td>
<td>Beta</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Network Management</th>
<th>OEM</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUTOSAR NM with VAG specific modifications</td>
<td>Audi</td>
<td>Available</td>
</tr>
<tr>
<td>AUTOSAR NM</td>
<td>Generic</td>
<td>Beta</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Transport Protocol</th>
<th>OEM</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUTOSAR FlexRay TP</td>
<td>Generic</td>
<td>Available</td>
</tr>
<tr>
<td>BMW FlexRay TP</td>
<td>BMW</td>
<td>Available</td>
</tr>
<tr>
<td>ISO 10681-2 TP</td>
<td>Generic</td>
<td>Available</td>
</tr>
</tbody>
</table>

Note: Further variants can be supported on request.
FlexRay Interaction Layer considers OEM specific definitions for

- Cyclic sending of frames/PDUs with application specific cycle periods
- Automatic calculation of message CRCs and message counters
- Modification of signal values and/or timings according to global system states (clamp15/IGN, ...)

![Diagram showing FlexRay Interaction Layer and related components](image)
Thank you for your attention.

For detailed information about Vector and our products please have a look at:

www.flexray-solutions.com

Author:
Carsten Böke
Vector Informatik GmbH
Ingersheimer Str. 24
70499 Stuttgart