Diagnostics with CANoe

Setting up a Diagnostic Test System based on DoIP
Agenda

- **Diagnostics in CANoe**
  - DoIP Basics
  - Live-Demo: DoIP test system
## Typical DoIP Testing tasks and challenges

<table>
<thead>
<tr>
<th>Task/Use Case</th>
<th>Challenges</th>
</tr>
</thead>
<tbody>
<tr>
<td>➤ Reuse of existing automated diagnostic tests (e.g. from CAN) on Ethernet-based network</td>
<td>➤ No Diagnostic Description available for Ethernet network</td>
</tr>
<tr>
<td></td>
<td>➤ New physical layer</td>
</tr>
<tr>
<td></td>
<td>➤ DoIP Vehicle identification / routing activation necessary</td>
</tr>
<tr>
<td>➤ Test of Ethernet or other network ECU behind DoIP Gateway</td>
<td>➤ No diagnostic description for Gateway available</td>
</tr>
<tr>
<td>➤ Test the Tester</td>
<td>➤ HW SUT is not available ➔ diagnostic ECU simulation necessary</td>
</tr>
<tr>
<td>➤ Create automated test for DoIP ECU</td>
<td></td>
</tr>
</tbody>
</table>
# Diagnostics in CANoe

## Overview

<table>
<thead>
<tr>
<th>CANoe</th>
<th>External control</th>
<th>Analysis</th>
<th>Simulation</th>
<th>Diagnostics</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>GUI</td>
<td>Online</td>
<td>Matlab/Simulink ®</td>
<td>Console</td>
<td>DiVa</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Logging</td>
<td>OSEK/ AUTOSAR Emulation</td>
<td>Session control</td>
<td>Test modules</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Offline</td>
<td></td>
<td>Fault memory window</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Test units</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>XCP/CCP</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Extensions (OEM specific, J1939, BAP, CANopen, AUTOSAR, ..)

---

**Test units**

- **VN26xx**
- **VN56xx**
- **VN16xx**
- **VN88/89xx**
- **VN75/76xx**

**I/O**

- **MOST**
- **BroadR-Reach**
- **Ethernet**
- **CAN/LIN/K-Line**
- **FlexRay**
- **AO/DIO**
- **Sensor buses**
Communication via...

... **“built-in diagnostic channel”**:  
- Uses almost all relevant communication parameters/attributes from diagnostic description and/or “Diagnostics/ISO TP” configuration dialog  
- Used by all interactive diagnostic windows in CANoe/CANalyzer:  
  - Diagnostics Console Window  
  - Fault Memory Window  
  - Session Control Window  
  - Diagnostic Parameters Window  
- No CAPL code necessary, but also usable with CAPL

... **CAPL Callback Interface (CCI)**:
- *(Almost)* every parameter/attribute relevant for communication can be modified via CAPL for fault injection  
- CAPL code necessary  
- Can **not** be used by interactive diagnostic windows, i.e. changes of attributes/parameters via CAPL do **not** have an influence on the communication via these windows (exception: variant selection)
Built-in diagnostics channel up to CANoe V9.0 SPx
Diagnostics in CANoe

Built-in diagnostics channel **since CANoe 10.0**

**Diagnostic Layer**
- Diagnostic Windows
- Test Modules, Test Units, **Simulation Nodes**
- CANoe/CANalyzer “built-in” Diagnostic Channel
- Transport Protocol DLL
- CAPL Callback Interface (CCI)

**Transport Layer**

**Physical Layer**
CAPL Callback Interface (CCI)

Necessary mainly for following use cases:

- Fault injection on tester or simulation side (especially on TP or lower levels!)
- Gateway implementation

Reference implementations (CAPL include files) for simple simulation/test purposes are available for:

- CAN TP
- LIN TP
- Flexray TP (AUTOSAR & ISO)
- DoIP
- K-Line
Agenda

Diagnostics in CANoe

- **DoIP Basics**
  
  Live-Demo: DoIP test system
Vehicle and external network

According to ISO 13400-2:2012(E)
DoIP Basics

Nested header structure

Ethernet frame:
- Eth Hdr
- IP Hdr
- TCP/UDP Hdr
- DoIP
- Eth CRC
- Max. 1518 Bytes (without VLAN-Tag)

DoIP message:
- Protocol Version
- Inv. Protocol Version
- Payload Type
- Payload Length
- DoIP Payload
- DoIP Source Address
- DoIP Target Address
- User Data
- e.g. UDS request

DoIP payload (e.g. Diagnostic Message):
- Bytes:
  - DoIP Source Address: 2 Bytes
  - DoIP Target Address: 2 Bytes
  - User Data: 0..4294967295

"Logical" DoIP addresses are needed for addressing ECUs in "conventional" networks like CAN, LIN, FlexRay, MOST etc.

Contains diagnostic request or response, e.g. 0x10 0x03 = extended Diagnostic Session
## DoIP Header Structure (example)

<table>
<thead>
<tr>
<th>Item</th>
<th>Byte</th>
<th>Example</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Protocol Version</td>
<td>0</td>
<td>0x02</td>
<td>DoIP Version 2 (ISO 13400-2:2012)</td>
</tr>
<tr>
<td>Payload Type</td>
<td>2</td>
<td>0x00</td>
<td>Vehicle Identification Request with VIN</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>0x03</td>
<td></td>
</tr>
<tr>
<td>Payload Length</td>
<td>4</td>
<td>0x00</td>
<td>Payload Length = 17 (0x11)</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>0x00</td>
<td></td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>0x00</td>
<td></td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>0x11</td>
<td></td>
</tr>
<tr>
<td>DoIP Payload</td>
<td>8...24</td>
<td>...</td>
<td>Vehicle Identification Number (VIN)</td>
</tr>
</tbody>
</table>
Connection and Vehicle Discovery

Physical Connect & (optional) address assignment via Auto-IP or DHCP

Vehicle announcement (3x)

[Tester did not receive the initial vehicle announcement]

Vehicle identification request

Vehicle identification response

[Tester is already reachable]

Vehicle identification request

Vehicle identification response

According to ISO 13400-2:2012(E)
Case 1: Without parameters

DoIP Basics
Vehicle Identification Sequence (1/2)

UDP: Vehicle identification request ()
UDP: Vehicle identification response (VIN1, EID1\(^1\)), ...)
UDP: Vehicle identification response (VIN2, EID2, ...)

1) EID = Entity Identification, typically contains the MAC address of the DoIP entity (here: of the ECU / Gateway)
Case 2: With Parameters, e.g. VIN

DoIP Basics

Vehicle Identification Sequence (2/2)

- UDP: Vehicle identification request (VIN1)
- UDP: Vehicle identification response (VIN1, EID1, ...)
- Discard request if VIN does not match
DoIP Session Example

According to ISO 13400-2:2012(E)

- Connection and vehicle discovery
- Open TCP socket
- Routing activation request
- Routing activation response
- Diagnostic message [Request]
- Diagnostic message positive ack.
- Diagnostic message [Response]
- Diagnostic request
- Diagnostic response
- Initialize non-DoIP network, if necessary
- Loop
- Close TCP socket

According to ISO 13400-2:2012(E)
Routing Activation Request

TCP: Routing activation request (SA\(^1\), Activation Type)

TCP: Routing activation response (SA, TA\(^2\), Routing Activation Response Code)

1) SA = Source Address, here equal to Tester Log. Address
2) TA = Target Address, here equal to ECU Log. Address
Diagnostic Messages

TCP: Diagnostic Message (SA, TA, UD)

TCP: Diagnostic Message Positive Acknowledgement (SA, TA, ACK Code, ...)  

TCP: Diagnostic Message (TA, SA, UD)

1) UD = User Data, contains the Diagnostic Request/Response

2) Only sent by a gateway after receiving, processing and copying the message correctly into the destination network transmission buffer
Agenda

Diagnostics in CANoe
DoIP Basics

- **Live-Demo: DoIP test system**
Further documentation: Application notes

**CANoe and CANalyzer as diagnostic tools**
- (AN-IND-1-001_CANoe_CANalyzer_as_Diagnostic_Tools.pdf)

**CAPL Callback Interface in CANoe**
- (AN-IND-1-012_CAPL_Callback_Interface.pdf)

**Diagnostics via CANoe Gateways**
- (AN-IND-1-004_Diagnostics_via_Gateway_in_CANoe.pdf)
Diagnostics with CANoe

Contact: sales@vector.com