

CANoe

Produktinformation

Inhaltsverzeichnis

1	Einführung CANoe	5
1.1	Bussysteme und Protokolle.....	6
1.2	Produktkonzept und -varianten.....	6
1.3	Lieferumfang	6
1.4	Systemvoraussetzungen	7
1.5	Zusätzliche Nutzungsszenarien	7
1.5.1	CANoe unter „EULA“	7
1.5.2	CANoe unter "ELA"	7
1.6	Weiterführende Informationen.....	7
2	Funktionen	8
2.1	Spezielle Funktionen	9
2.2	Datenbasisunterstützung	10
3	Analyse.....	10
3.1	Messaufbau	11
3.2	Trace-Fenster	12
3.3	Grafik-Fenster	13
3.4	Scope-Fenster	14
3.5	Daten-Fenster	15
3.6	Statistik-Fenster.....	15
3.7	State Tracker.....	16
3.8	Write-Fenster	17
3.9	Video-Fenster	18
3.10	GPS-Fenster	19
3.11	Trigger und Filter.....	19
3.12	Logging/Replay	20
4	Stimulation/Simulation.....	20
4.1	Variablen und Generatoren.....	20
4.1.1	Interaktiver Generator	21
4.1.2	Signalgenerator.....	22
4.2	Startwerte setzen.....	23
4.3	Symbol-Mapping	23
4.4	Interaction Layer, Netzwerk-Management, Transportprotokolle	23
4.4.1	OEM-spezifische Erweiterungen	24
4.4.2	Interaction-Layer-Konfiguration.....	24
4.5	MATLAB/Simulink.....	25
4.5.1	Weitere Funktionen der CANoe MATLAB-/Simulink-Integration	25
4.5.2	Weiterführende Informationen.....	26
5	Test.....	27
5.1	Test von Steuergeräten und Netzwerken	27
5.2	CANoe RT/VN8900 und CAPL-on-Board.....	30
5.3	Vector Tool Plattform.....	30
5.4	CAN-/CAN-FD-Störungen	30
5.5	Weiterführende Informationen.....	31

6	Diagnose	31
6.1	Weiterführende Informationen.....	34
7	Programmierung	34
7.1	CAPL-Anbindung.....	34
7.1.1	C-ähnliche Syntax.....	34
7.1.2	Ereignisorientierte Steuerung.....	35
7.1.3	Symbolischer Zugriff.....	35
7.1.4	Anwendungsspezifische Spracherweiterungen	35
7.2	CAPL Browser	37
7.3	.NET-Programmierung.....	38
7.4	Fehlersuche.....	39
7.4.1	Weiterführende Informationen.....	40
7.5	Visual Sequencer	40
8	Panels	40
9	Hardware-Schnittstellen.....	41
10	Schnittstellen zu anderen Anwendungen	41
10.1	COM-Schnittstelle	41
10.1.1	Weiterführende Informationen.....	42
10.2	FDX	42
10.3	ASAM XIL API	42
10.4	FMI.....	42
11	Option .Scope	42
11.1	Anwendungsgebiete	43
11.2	Die Vorteile im Überblick	43
11.3	Unterstützte Protokolle	43
11.4	Unterstützte Oszilloskop-Hardware.....	44
11.5	Oszilloskop-Software	44
11.5.1	Konfigurationsfunktionen	44
11.5.2	Trigger-Funktionen	44
11.5.3	Analysefunktionen	45
11.5.4	Offline-Funktionen.....	45
12	Option .Sensor.....	45
12.1	Anwendungsgebiete	46
12.2	Unterstützte Sensor-Protokolle	46
12.3	Highlights.....	46
13	Option .XCP	47
13.1	Anwendungsgebiete	47
13.2	Steuergerätezugriff.....	48
13.2.1	Unterstützte Bussysteme und Protokolle.....	48
13.2.2	VX1000 Mess- und Kalibrier-Interface.....	48
13.2.3	Hardware-Debugger-Support	48
13.3	Die Vorteile im Überblick	48
13.4	Funktionen	49
13.4.1	Spezielle Funktionen	49
13.5	Integration in CANoe	49

13.6	Konfiguration.....	50
14	Option .AMD (AUTOSAR Monitoring and Debugging).....	50
14.1	Anwendungsgebiete	51
14.1.1	Konfigurationsprozess für konventionelles Messen	52
14.1.2	ASAP2 Updater	52
14.2	MICROSAR Monitoring Feature	52
14.2.1	Generisches Messen.....	52
15	Funktionserweiterung für spezielle Anwendungen.....	53
15.1	DiVa (Diagnostic Integration and Validation Assistant).....	53
16	Schulungen.....	53

V6.0 04/2018

Gültig für CANoe ab Version 11.0.

In diesem Dokument werden die CANoe Anwendungsgebiete Analyse, Stimulation/Simulation, Test, Diagnose und ihre einzelnen Funktionen aufgeführt. Das Dokument enthält zusätzlich einen kurzen Überblick über die Programmierung in CANoe, Zusatzoptionen und -Programme, Hard- und Software-Schnittstellen.

Produktinformationen und technische Daten zu CANoe und den Optionen .FlexRay, .LIN und .MOST werden in jeweils eigenen Dokumenten bereitgestellt.

1 Einführung CANoe

CANoe ist das vielseitige Werkzeug für die Entwicklung, den Test und die Analyse von ganzen Steuergerätenetzwerken, aber auch von einzelnen Steuergeräten. Es unterstützt Netzwerk-Designer, Entwicklungs- und Testingenieure bei OEMs und Zulieferern im kompletten Entwicklungsprozess – von der Planung bis hin zur Inbetriebnahme kompletter verteilter Systeme oder einzelner Steuergeräte.

Am Anfang des Entwicklungsprozesses werden mit CANoe Simulationsmodelle erstellt, die das Verhalten der Steuergeräte nachbilden. Im weiteren Verlauf der Steuergeräteentwicklung sind diese Modelle durchgängig die Grundlage für die Analyse, den Test und die Integration von Bussystemen und Steuergeräten. So können Problemstellen frühzeitig erkannt und korrigiert werden. Zur Auswertung der Ergebnisse stehen grafisch orientierte sowie textbasierte Analysefenster zur Verfügung.

Zur einfachen und automatisierten Durchführung von Tests enthält CANoe das Test Feature Set. Damit werden sequentielle Testabläufe modelliert, ausgeführt und automatisch ein Testreport erzeugt. Für die Diagnosekommunikation mit dem Steuergerät steht in CANoe zudem das Diagnostic Feature Set zur Verfügung.

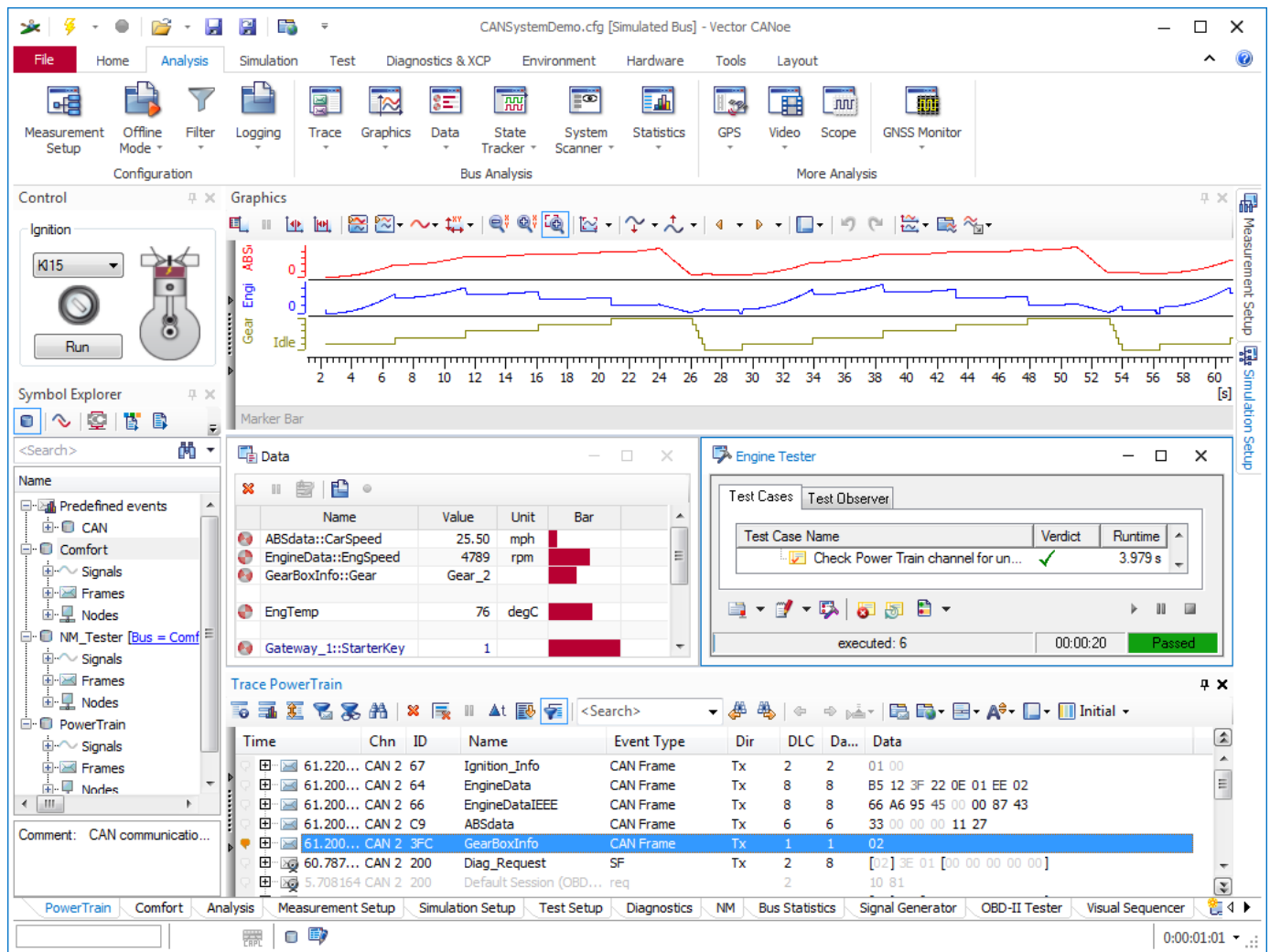


Bild 1: CANoe Bedienoberfläche

1.1 Bussysteme und Protokolle

In CANoe stehen für die unterschiedlichen Bussysteme und CAN-basierten Protokolle verschiedene CANoe Optionen zur Verfügung, die beliebig kombiniert werden können.

CANoe unterstützt die folgenden **Bussysteme**: CAN, CAN FD, LIN, MOST, FlexRay, J1708, Ethernet, K-Line, A429, WLAN und AFDX¹

Die Option .CAN ist Basis für die unterstützten **CAN-basierten Protokolle**: J1939, ISO 11783, CANopen, GMLAN, CANaero.

Weitere auf Anfrage.

Ausführliche Informationen zu den Optionen finden Sie in den jeweiligen Produktinformationen.

1.2 Produktkonzept und -varianten

Für spezielle Einsatzzwecke bei OEMs und Zulieferern gibt es CANoe in folgenden Varianten:

- > Mit vollem Funktionsumfang
- > Als Runtime-Variante (run) mit unveränderbaren Simulationen, den vollen Analysefunktionen und einfachem Zu- und Abschalten der Netzwerkknoten. Diese Variante ist für Anwender gedacht, die schnell und einfach ihr Steuergerät im Zusammenspiel mit einer vorgegebenen Restbussimulation testen wollen.
- > Als Project-Execution-Variante (pex) mit ausschließlich grafischer Bedienoberfläche. Die Testfälle und Ergebnisse werden möglichst einfach gesteuert, ohne dass die zu Grunde liegenden Botschaften speziell ausgewertet werden müssen.

Der CANoe/CANalyzer Kompatibilitätsmodus erlaubt Ihnen das Verwenden beider Programme, z.B. innerhalb eines Projektes oder einer Organisation, indem einheitliche Konfigurationen ausgetauscht werden. Somit wird für jeden Anwendungsfall das passende Programm in der optimalen Variante eingesetzt. Während in der Steuergeräteentwicklung mit der Vollvariante von CANoe gearbeitet wird, prüfen die Systemintegratoren und Testfahrer mit derselben Konfiguration die Buskommunikation in CANalyzer.

1.3 Lieferumfang

Der Lieferumfang hängt von der gewählten Produktvariante ab. Die Vollversion enthält, neben CANoe selbst, folgende Bestandteile:

- > Zahlreiche Beispielkonfigurationen zum Gesamtsystem, zu allen installierten Bussystem-Optionen und zu speziellen Anwendungsfällen wie Test und Diagnose
- > Editoren und Anzeigeprogramme für unterschiedliche Datenbasisformate, für Panels und zur CAPL-Programmierung
- > Installationsanleitungen, Handbücher und Hilfen
- > Transportprotokoll (TP) nach ISO/DIS 15765-2 sowie den Interaction Layer (IL) gemäß Vector Spezifikation

Weitere Module, wie z.B. OEM-spezifische TP oder IL, sind nicht im Standardlieferumfang enthalten, können aber über den Support kostenlos angefragt werden.

¹ AFDX® ist ein eingetragenes Warenzeichen von Airbus

1.4 Systemvoraussetzungen

Komponente	Empfehlung	Minimum
Prozessor	<ul style="list-style-type: none"> > Intel-kompatibel > > 2 GHz > ≥ 2 Kerne 	<ul style="list-style-type: none"> > Intel-kompatibel > 1 GHz > 2 Kerne
Speicher (RAM)	16 GB	4 GB
Festplattenplatz	≥ 20 GB SSD	≥ 3 GB
	Je nach verwendeten Optionen und benötigten Betriebssystemkomponenten	
Bildschirmauflösung	Full HD	1280×1024 Pixel
Grafikkarte	DirectX 9.0c oder höher und Shader Model 1.1 oder höher*	
Betriebssystem	Windows 7 SP1/8.1/10 (Version 1709)	

Tabelle 1: * Wird nur für den State Tracker benötigt

1.5 Zusätzliche Nutzungsszenarien

1.5.1 CANoe unter „EULA“

Ergänzend zu Ziffer 2.1 der „End User Lizenzbestimmungen für Standard-Softwareprodukte von Vector“ gelten folgende Nutzungsszenarien für CANoe als erlaubt; „Eine Automatisierung von CANoe oder der Remote-Zugriff auf CANoe ist mit einer Device-Lizenz erlaubt, wenn CANoe betrieben wird, um mit Vector-Hardware (VN, VT, VX) auf ein reales System zuzugreifen (zum Beispiel an einem Testplatz oder in einer Server-Umgebung).“

1.5.2 CANoe unter „ELA“

Ergänzend zu Ziffer 2.1 und Ziffer 2.2 der „Enterprise-Lizenzbestimmungen für Standard-Softwareprodukte von Vector“ gelten folgende Nutzungsszenarien für CANoe als erlaubt; „Eine Automatisierung von CANoe oder der Remote-Zugriff auf CANoe ist mit einer Device-Lizenz und/oder Named-User License erlaubt, wenn CANoe betrieben wird, um mit Vector-Hardware (VN, VT, VX) auf ein reales System zuzugreifen (zum Beispiel an einem Testplatz oder in einer Server-Umgebung).“

1.6 Weiterführende Informationen

> [Vector Download-Center](#)

> **Demo-Versionen**

Für CANoe stehen im Internet diverse Demo-Versionen zur Verfügung. Sie enthalten zu den verschiedenen Anwendungsgebieten Beispielkonfigurationen sowie eine detaillierte Hilfe, in der alle Funktionen von CANoe beschrieben werden.

> **Application Notes**

In den folgenden Kapiteln wird auf zusätzliche Application Notes verwiesen, die das Thema zu den einzelnen Anwendungsgebieten vertiefen.

> [CANoe Feature Matrix](#)

Weitere Informationen über Varianten, Kanal- und Bussystemunterstützung stehen in der Feature Matrix zur Verfügung.

2 Funktionen

Zu den Grundfunktionen von CANoe gehören:

- > Einsatz von Datenbasen, die das jeweilige Netzwerk beschreiben (z.B. DBC, FIBEX, LDF, NCF, AUTOSAR System Description, MOST Function Catalog)
- > Simulation kompletter Systeme und Restbussimulationen
- > Analyse der Buskommunikation
- > Test kompletter Netze und/oder einzelner Steuergeräte
- > Diagnosekommunikation nach KWP2000 und UDS sowie Einsatz als vollwertiger Diagnosetester
- > Freie Programmierbarkeit durch die Programmiersprache CAPL zur Unterstützung von Simulation, Analyse und Test
- > Erstellen benutzerdefinierter Oberflächen zur Steuerung der Simulation und Tests oder zur Anzeige der Analysedaten
- > Einbinden von zusätzlicher I/O-Hardware und/oder spezieller Test-Hardware (VT System)
- > Intuitive Benutzerschnittstelle mit flexiblem Docking-Konzept und übersichtlichen Menüstrukturen
- > Unterstützung neuer Vector Bus-Hardware
 - > VN1610 (2 Kanäle – CAN)
 - > VN1611 (2 Kanäle – CAN und LIN/K-Line)
 - > VN1630 (4 Kanäle – CAN und LIN/K-Line)
 - > VN1640 (4 Kanäle – CAN und LIN/K-Line)

2.1 Spezielle Funktionen

Zu den Highlights von CANoe gehören:

- > Für kritische, echtzeitrelevante Simulationen und Tests arbeitet CANoe auf zwei PCs verteilt
- > Mit CAPL-on-Board lassen sich CAPL-Knoten direkt auf der Interface-Hardware ausführen
- > Zahlreiche Zusatzmodule sorgen für ein einfaches Anpassen an OEM-spezifische Dienste und Protokolle (Transportprotokolle, Netzwerk-Management, Interaction Layer, ...)
- > Diagnosefunktionen:
 - > Parametrierung der Diagnose mittels Diagnosebeschreibungen als ODX 2.0.1/2.2.0, MDX 2.0/3.0 oder CDD
 - > Definition einfacher Diagnose-Services mit dem Basisdiagnose-Editor
 - > Unterstützung von physikalischer und funktionaler Adressierung
 - > Schnelle und einfache On-Board-Diagnose mit dem eingebauten OBD-II-Tester
 - > Diagnosebeobachter für UDS und KWP2000 auf Basis parametrierbarer Diagnosebeschreibungen
 - > Transportprotokoll-Beobachter für ISO/DIS 15765-2
 - > Unterstützung von DoIP (Diagnose über IP) und HSFZ (High Speed Fahrzeugzugang)
 - > Spezielle Diagnose-CAPL-Funktionen zum Simulieren und Testen von Steuergeräten
- > Das Vector VT System ermöglicht umfassende Steuergerätestests, bei denen neben dem Buszugang auch die I/O-Leitungen verwendet werden
- > Testfälle können mit Anforderungen in gängigen Requirement-Werkzeugen, wie z. B. Telelogic DOORS verknüpft werden.
- > CANoe unterstützt das Einbinden von MATLAB/Simulink-Modellen
- > CANoe kann als Ablaufumgebung für Steuergeräte-Code von AUTOSAR- oder OSEK-OS-Applikationen verwendet werden
- > Zugriff auf steuergeräteinterne Signale über XCP/CCP einschließlich Protokollanalyse für CAN, CAN FD, FlexRay und Ethernet
- > Die Ansteuerung digitaler und analoger I/O-Module sowie Mess-Hardware erlaubt das Verarbeiten realer Signalwerte in Simulationen und Testumgebungen
- > Offene Software-Schnittstellen, wie z.B. Microsoft COM, FDX , FMI oder ASAM XIL API ermöglichen die Integration in bestehende Systemumgebungen.

2.2 Datenbasisunterstützung

CANoe unterstützt Systembeschreibungen auf Basis folgender Formate: DBC (CAN), LDF (LIN), XML (MOST), FIBEX (FlexRay) und AUTOSAR Descriptions (CAN/FlexRay/Ethernet).

CANoe kann folgende Diagnosebeschreibungen verarbeiten: CDD (CANdelaStudio), ODX 2.0.1/2.2.0 in Form von PDX-Dateien und MDX 2.0/3.0.

Die Informationen dieser Datenbasen können in CANoe symbolisch dargestellt und verwendet werden.

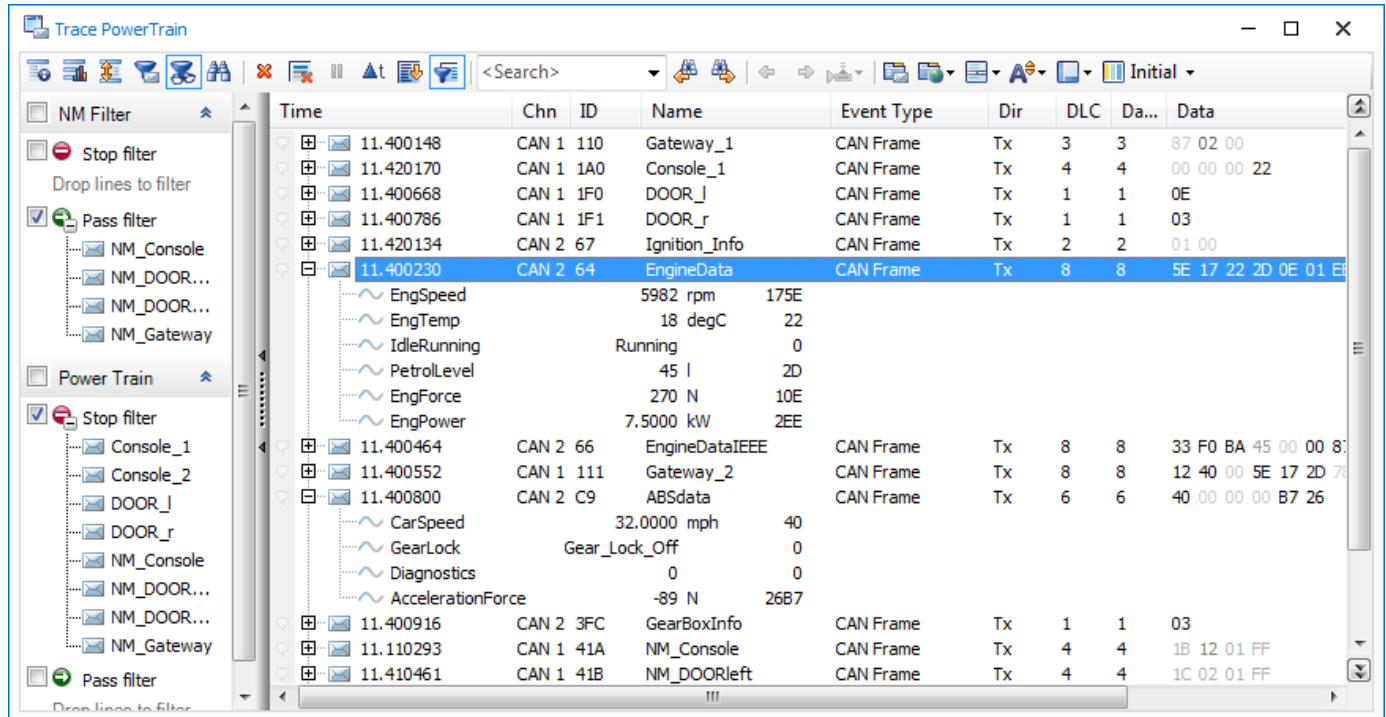


Bild 2: Trace-Fenster mit Analysefiltern und Diagnoseinterpretation

3 Analyse

Grundlage für die Analyse in CANoe ist der Datenfluss von der Datenquelle bis zur Anzeige bzw. Aufzeichnung. Die Daten können dabei zusätzlich verarbeitet werden. Es können z.B. Filter eingebunden werden, die festlegen, welche Daten für die Analyse betrachtet werden sollen und welche nicht.

Highlights

- > Einfaches Konfigurieren der Analysefenster durch Drag & Drop. Es können z.B. Botschaften oder Signale von einem Analysefenster in ein anderes kopiert oder verschoben werden.
- > Für eine vielseitige Analyse können von einem Fenstertyp (z.B. Grafik-Fenster) mehrere Fenster in den Datenfluss eingebunden werden, was ein paralleles Analysieren ermöglicht.
- > Einfaches Starten und Stoppen von Datenaufzeichnungen über die Statusleiste.

CANoe stellt dem Anwender unter anderem die folgenden Fenster und Blöcke zur Verfügung.

3.1 Messaufbau

Im Messaufbau wird der Datenfluss grafisch dargestellt und konfiguriert.

- > **Datenquelle festlegen** (online/offline)
Als Online-Datenquelle dient der simulierte Bus oder der über die Hardware (z.B. CANcardXL) angeschlossene reale Bus. Als Offline-Datenquelle dient eine Datei mit aufgezeichneten Daten.
- > **Analysefenster einfügen**
In den einzelnen Fenstern können die Daten je nach Analyse-Anforderung unterschiedlich dargestellt werden, z.B. als grafische Darstellung von Signalverläufen oder als Anzeige von Signalwerten.
- > **CAPL-Programmknotten einfügen**
Ein CAPL-Programmknotten kann z.B. zum Filtern von Daten oder für verschiedene arithmetische Operationen verwendet werden.
- > **Filter einfügen**
Für eine übersichtliche Darstellung der Daten können Filter eingesetzt werden, die festlegen, welche Daten durchgelassen oder welche explizit gesperrt werden sollen. Filter können während oder nach der Messung aktiv sein, sie können einzelne Signale bis hin zu den Kanälen eines ganzen Bussystems filtern.
- > **Trigger-Bedingungen einfügen**
Trigger-Bedingungen können wie die Filter zur Datenreduzierung eingesetzt werden. Trigger werden gezielt als Reaktion auf Busereignisse konfiguriert und können miteinander verbunden werden.
- > **Daten aufzeichnen**
Für eine Analyse nach der Messung können in einer Logging-Datei Daten aufgezeichnet werden, die später wieder als Offline-Datenquelle eingebunden und abgespielt werden können.

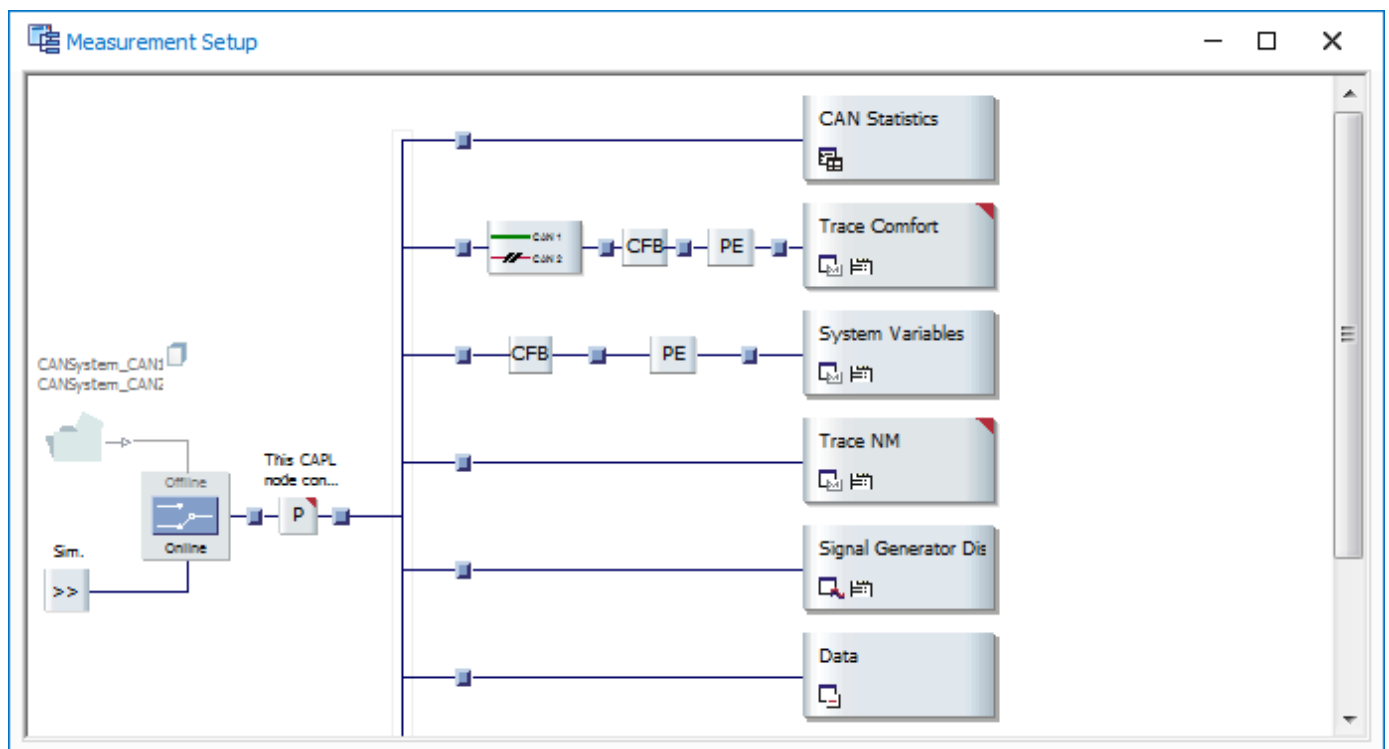


Bild 3: Messaufbau mit Online-Datenquelle, CAPL-Programmblock und verschiedenen Filtern

3.2 Trace-Fenster

Im Trace-Fenster werden Busaktivitäten, wie z.B. das Senden von Botschaften oder Error Frames, aufgelistet. Für jede Botschaft können einzelne Signalwerte dargestellt werden. Für die Analyse der Daten stehen unter anderen die folgenden Funktionen zur Verfügung:

> Filter einfügen

Im Trace-Fenster gibt es verschiedene Arten von Filtern. Mit ihnen kann die angezeigte Datenmenge reduziert werden, es können sogar Daten aus dem Datenstrom gelöscht werden.

> Unveränderte Daten ausblenden

Um die Übersichtlichkeit zu erhöhen, können unveränderte Daten langsam verblassen oder ganz aus der Ansicht entfernt werden.

> Ereignisse einfärben

Wichtige Ereignisse und Botschaften können farblich hervorgehoben werden.

> Marker setzen

Für das Kennzeichnen und das schnelle Wiederfinden von Ereignissen können Marker gesetzt werden. Der Marker wird einem Ereignis und damit auch seinem Zeitstempel zugeordnet. Die gesetzten Marker können auch in anderen Analysefenstern angezeigt werden.

> Statistiken anzeigen

Über verschiedene Ansichten können z.B. Details der Botschaften/Signale einschließlich ihrer Werte detailliert angezeigt werden oder Differenzen der Zeitstempel oder der Signalwerte berechnet werden.

> Daten aufzeichnen

Der Inhalt des Trace-Fensters kann teilweise oder komplett exportiert werden. Bereits exportierte Dateien können nachträglich in ein anderes Format konvertiert werden, z.B. um denselben Datensatz in verschiedenen Programmen weiter zu verarbeiten.

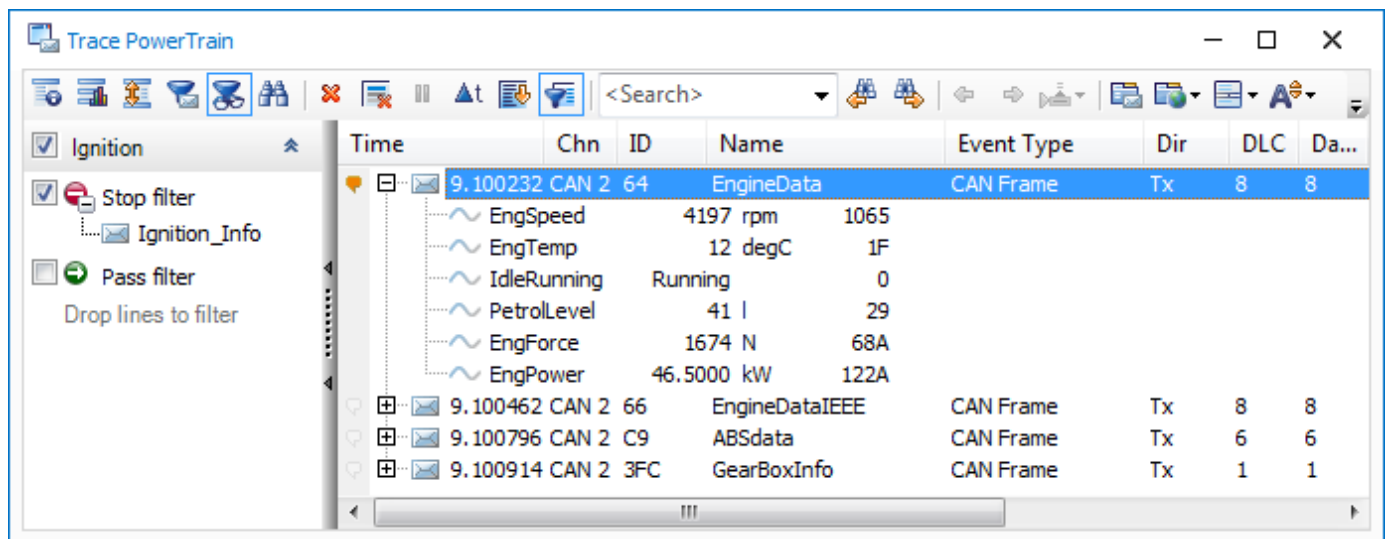


Bild 4: Trace-Fenster mit aktivem Sperrfilter und gesetztem Marker

3.3 Grafik-Fenster

Im Grafik-Fenster werden Werte von Signalen, Umgebungsdaten und Diagnoseparametern grafisch als Kurve dargestellt. Zum Messen und Auswerten dieser Kurven stehen unter anderen die folgenden Funktionen zur Verfügung:

> **Messmarken/Differenzmarken anzeigen**

Mit Hilfe von Mess- oder den Differenzmarken können absolute oder relative Messwertanalysen durchgeführt werden. Die Messmarke kann mit der Anzeige des Trace-Fensters synchronisiert werden.

> **Marker setzen**

Für das Kennzeichnen und das schnelle Wiederfinden von Ereignissen können Marker gesetzt werden. Der Marker wird einem Ereignis und damit auch seinem Zeitstempel zugeordnet. Die gesetzten Marker können auch in anderen Analysefenstern angezeigt werden.

> **Messspalten anzeigen**

In der Legende können für jedes Signal globale oder lokale Minima und Maxima oder Y-Differenzen zwischen Signalen gleichen Typs angezeigt werden.

> **X/Y-Modus**

Jedes in der Signalliste konfigurierte Signal kann per rechtem Mausklick auch als x-Achste konfiguriert werden.

> **Statistiken anzeigen**

Für ausgewählte oder alle Signale des Grafik-Fensters können Statistikdaten, wie Minimum, Maximum, Durchschnittswert und Standardabweichung, zusammengestellt werden.

> **Daten aufzeichnen**

Signale des Grafik-Fensters können automatisch oder manuell während der Messung aufgezeichnet werden. Die Signale werden dazu aus den Botschaften extrahiert und binär in signalbasierten MDF-Dateien abgespeichert.

Im Grafik-Fenster kann der gesamte Signalverlauf oder nur der sichtbare Ausschnitt in eine Datei gespeichert werden.

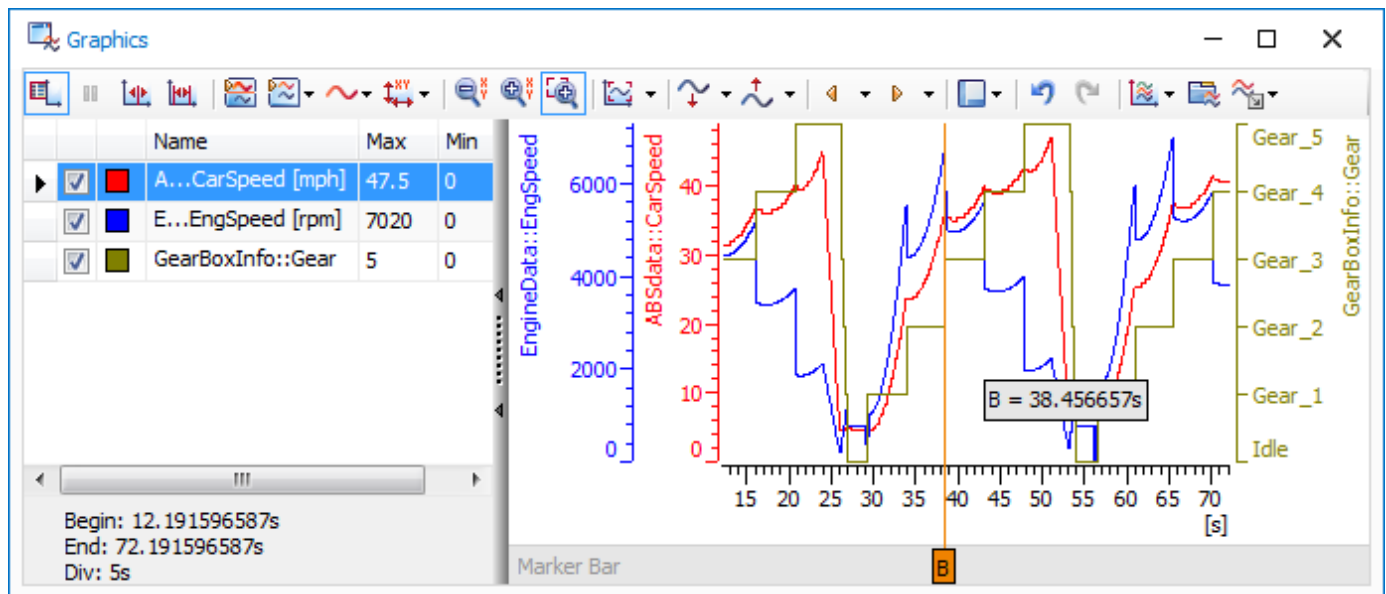


Bild 5: Grafik-Fenster mit gesetztem Marker

3.4 Scope-Fenster

Das Scope-Fenster stellt Buspegelmessungen grafisch dar und dient zur Analyse von Protokollfehlern (siehe auch Option „Scope, Kapitel 11“).

> Trigger setzen

Im Scope-Fenster kann manuell, über CAPL oder über vorkonfigurierte Bedingungen getriggert werden. Es können beliebig viele Trigger-Bedingungen angelegt werden, wobei die einzelnen Trigger-Bedingungen über eine ODER-Beziehung verknüpft sind.

> Messwerte analysieren

Das Diagramm stellt die Messwerte und die logische Interpretation grafisch dar.

> Signale vergleichen

Um Daten zu vergleichen, kann zwischen verschiedenen Möglichkeiten gewählt werden. So können unterschiedliche zeitliche Abschnitte derselben Datenerfassung oder zeitliche Abschnitte unterschiedlicher Datenerfassungen verglichen werden.

> Daten aufzeichnen

Zur späteren Analyse können erfasste Daten exportiert und importiert werden.

> CAPL-Steuerung in Testmodulen

Das Scope-Fenster kann aus einem CAPL-Testmodul gesteuert werden. So kann z.B. mit CAPL getriggert oder aber auf ein Scope-Event gewartet werden.

> Messmarken

Mit Messmarkern können an definierten Stellen Zeit, Spannung sowie deren Differenzen gemessen werden. Die Messmarker zeigen über Tooltips die logische Wertigkeit (z.B. dominant/rezessiv) der Bits an. Alle physikalischen Messwerte werden in einer separaten Legende angezeigt.

> Globale Marker

Im Scope-Fenster können Marker zum Kennzeichnen signifikanter Stellen einer Messung gesetzt werden. Jeder Marker hat einen Namen und einen definierten Zeitstempel. Die Marker können mit anderen Analysefenstern synchronisiert werden.

> Augendiagramm

Erweiterte Analysemöglichkeiten, z.B. für Bitüberlagerung oder Darstellung des „idealen Bits“.

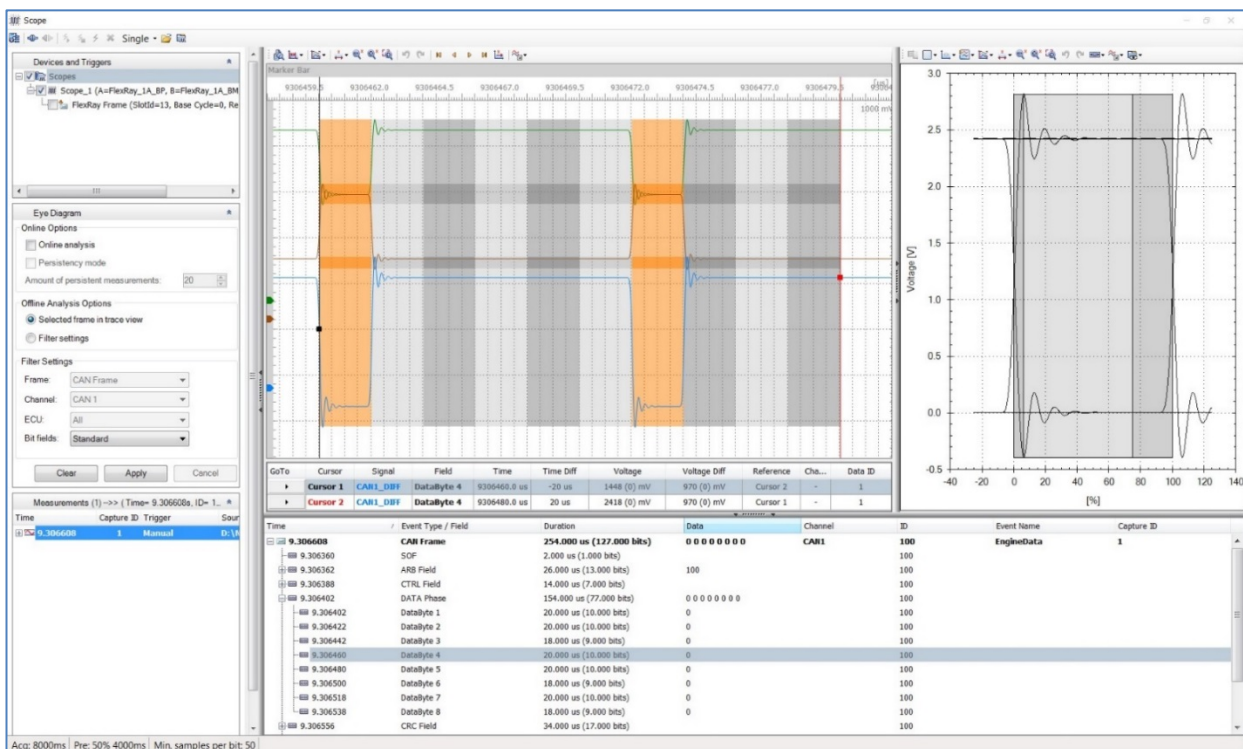


Bild 6: Scope-Fenster mit Augendiagramm

3.5 Daten-Fenster

Mit dem Daten-Fenster können Werte von Signalen, Systemvariablen und Diagnoseparametern in verschiedenen Darstellungsarten angezeigt werden.

> Werte anzeigen

Die Anzeige der Daten kann als Roh- oder symbolische Werte erfolgen. Weitere Anzeigevarianten sind die wissenschaftliche Darstellung sowie das Anzeigen von globalen und lokalen Min-/Max-Werten.

> Daten aufzeichnen

Signale können während der Messung aufgezeichnet und in das binäre Format MDF abgespeichert werden.

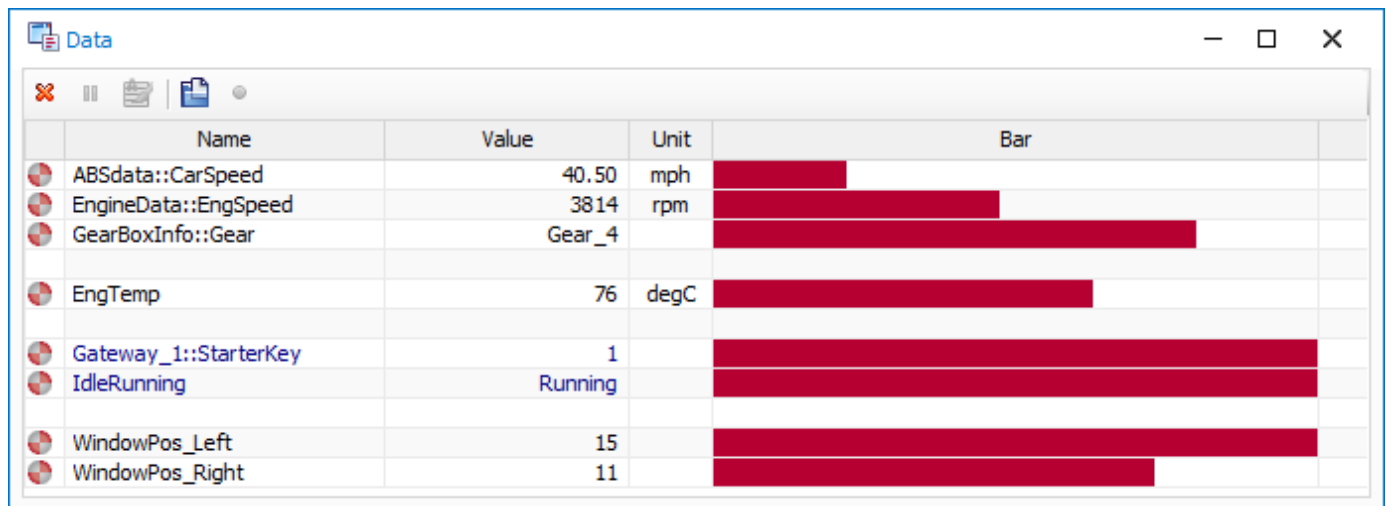


Bild 7: Daten-Fenster mit verschiedenen Darstellungsarten der eingehenden Werte

3.6 Statistik-Fenster

Das Statistik-Fenster zeigt statistische Informationen der Busaktivitäten (CAN, LIN, FlexRay) während einer Messung an. Dazu gehören z.B. die Buslast auf Knoten- und Frame-Ebene, Burst-Zähler/-Dauer, Zähler/Raten für Frames und Fehler und Controller-Zustände.

> Statistikdaten einzelner Kanäle anzeigen

Die Anzeige der statistischen Daten kann auf einen bestimmten Kanal begrenzt oder aber für alle verfügbaren Kanäle konfiguriert werden.

> Aktualisierungsintervall einstellen

Das Intervall zur Aktualisierung der Anzeige kann verändert werden.

> Statistik pausieren

Während einer Messung kann die Anzeige der Statistikdaten angehalten werden.

Statistic	Current / Last	Min	Max	Avg
Busload [%]	2.29	0.00	2.34	2.23
Engine	1.58	0.00	1.64	1.55
Gateway	0.70	0.00	0.71	0.68
DiagRequest...	0.00	0.00	0.00	0.00
Ignition_Info (...)	0.67	0.00	0.67	0.65
NM_Gateway...	0.03	0.00	0.03	0.03
NM_Tester	0.00	0.00	0.00	0.00
Unknown sender	0.00	0.00	0.00	0.00
Unknown events	0.00	0.00	0.00	0.00
Min. Send Dist. [ms]	0.000	n/a	n/a	n/a
Burst Time [ms]	0.922	0.300	0.958	0.851
Bursts [total]	1406	n/a	n/a	n/a
Frames per Burst	5	2	5	4
Std. Data [fr/s]	133	0	134	129

Bild 8: CAN-Statistik-Fenster mit Statistikdaten für einen Kanal (CAN 2).

Bestimmte CAN-/LIN-/FlexRay-Statistiken können in Analysefenstern, wie z.B. dem Grafik-Fenster, oder aber auch in Programmknoten über automatisch definierte Statistik-Systemvariablen ausgewertet werden. Diese Systemvariablen stehen für jeden konfigurierten Netzwerkkanal zur Verfügung und werden unabhängig vom Statistik-Fenster aktualisiert.

3.7 State Tracker

Mit dem State Tracker können Zustände, Zustandsübergänge, CAN-/CAN-FD-Frames, Signale und Diagnoseparameter analysiert und zeitliche Abhängigkeiten visualisiert werden. Der State Tracker eignet sich besonders zu Anzeige von digitalen Ein- und Ausgängen sowie von Statusinformationen, wie Klemmenstatus oder Netzwerk-Management-Zustände.

> Fehler suchen

Anhand der Analyse des zeitlichen Verlaufs von Zuständen, Signalen und Zustandsübergängen können Fehler gesucht und Funktionen überwacht werden.

> Informationen analysieren

Unterschiedliche Informationen wie die Zustände steuergeräteinterner Kommunikation, Bussignale oder Steuergeräte-I/Os können zusammen analysiert werden.

> AUTOSAR Runnables überwachen

Monitoring von Runnable-Zuständen, die über XCP ausgelesen wurden.

> Trigger setzen

Für das Auslösen einer Messung können Trigger-Bedingungen frei definiert werden.

> Marker setzen

Für das Kennzeichnen von Messzeitpunkten können Marker gesetzt werden. Die Zeit zwischen den gesetzten Markern kann gemessen werden.

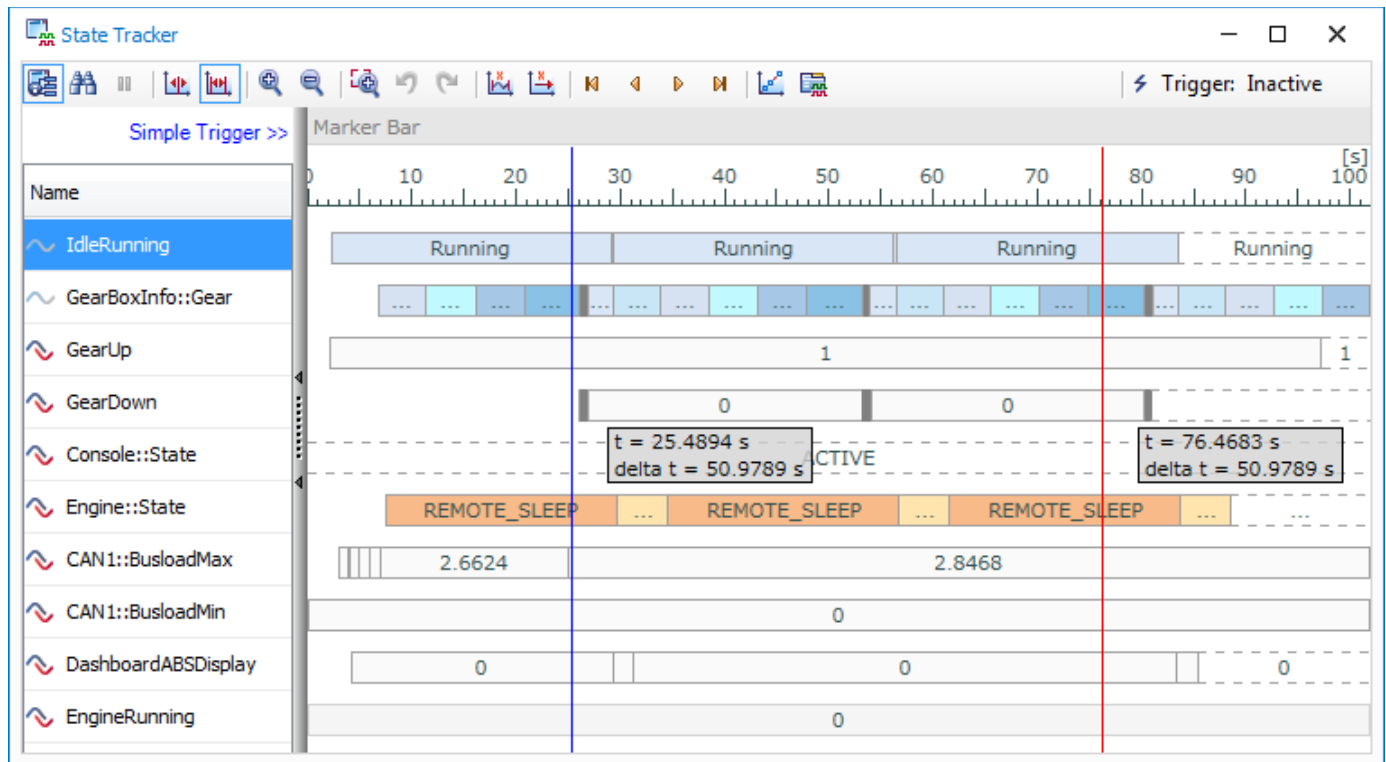


Bild 9: State Tracker zur Analyse von Steuergerätezuständen

3.8 Write-Fenster

Im Write-Fenster werden Systemmeldungen und anwenderspezifische Ausgaben aus CAPL-Programmen angezeigt.

> Ausgabe konfigurieren

Das Write-Fenster bietet verschiedene Ansichten, um die Meldungen je nach Quelle zu filtern.

> Ausgabe protokollieren

Die Ausgabe des Write-Fensters kann entweder in eine Datei gespeichert oder als Text in die Zwischenablage kopiert und von dort in andere Windows-Anwendungen übernommen werden.

> Status anzeigen

Die Statusanzeige des Write-Fensters informiert über neue, ungelesene Warn- und Fehlermeldungen im Write-Fenster.

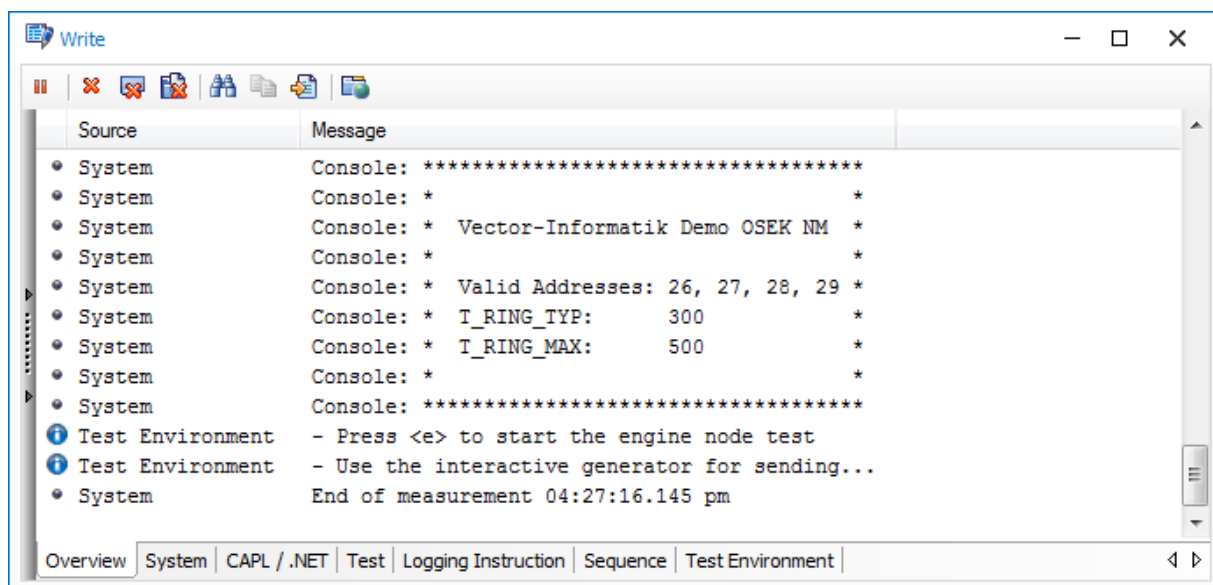


Bild 10: Write-Fenster mit Systemmeldungen und CAPL-Ausgaben

3.9 Video-Fenster

Mit dem Video-Fenster können Video-Dateien aufgezeichnet und wiedergeben werden.

Abhängig von der aktuellen Systemkonfiguration und den bereits auf dem System installierten DirectShow-Komponenten werden im Video-Fenster die meisten gängigen Video-Formate unterstützt. So z.B. nicht komprimierte AVI, komprimierte AVI (mit entsprechend installierten Codec, z.B. DivX), MPG, MPEG, WMV und MP4.

> Online-Messung & Aufzeichnung

Mit Hilfe eines Video-Fensters können Video-Dateien von einer konfigurierten Video-Quelle (z.B. einer Kamera) aufgezeichnet werden. Nach der Messung zeigt das Video-Fenster die aufgezeichnete Video-Datei an, die abgespielt und auf der navigiert werden kann.

> Datei-Import

Außerhalb der Messung ist es möglich, Video-Dateien in ein Video-Fenster zu importieren.

> Offline-Analyse

Wird eine Video-Datei als Offline-Quelle in der Konfiguration des Video-Fensters definiert, dann wird diese bei der Offline-Analyse angezeigt. Typischerweise wird als Offline-Quelle die Video-Datei definiert, die während der Online-Messung aufgezeichnet wurde.

> Fenster-Synchronisation

Video-Dateien können mit anderen Analysefenstern synchronisiert werden.

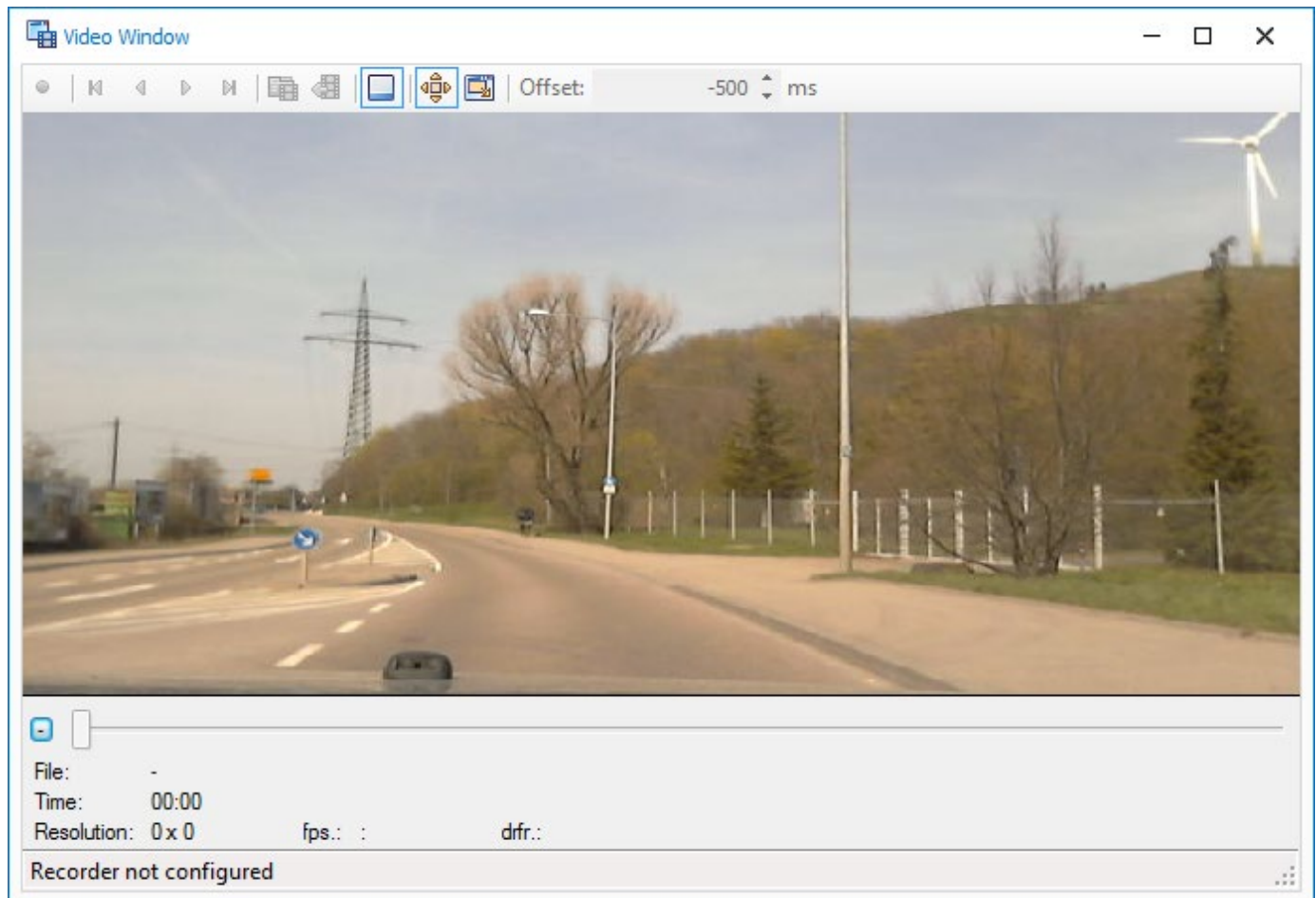


Bild 11: Video-Fenster

3.10 GPS-Fenster

Mit dem GPS-Fenster können GPS-Informationen in CANoe integriert werden. Das GPS-Fenster ist im Basisumfang von CANoe enthalten.

- > **Daten anzeigen**
GPS-Daten können in den Analysefenstern angezeigt werden.
- > **Offline-Analyse**
GPS-Daten können für eine spätere Analyse aufgezeichnet und abgespielt werden.
- > **Fenster-Synchronisation**
GPS-Daten können mit anderen Analysefenstern synchronisiert werden.
- > **Karte anzeigen**

Im GPS-Fenster werden die aktuelle Fahrzeugposition und die zurückgelegte Wegstrecke auf einer elektronischen Landkarte dargestellt. Bei der Interpretation aufgezeichneter Messdaten können damit geografische Gegebenheiten berücksichtigt werden.

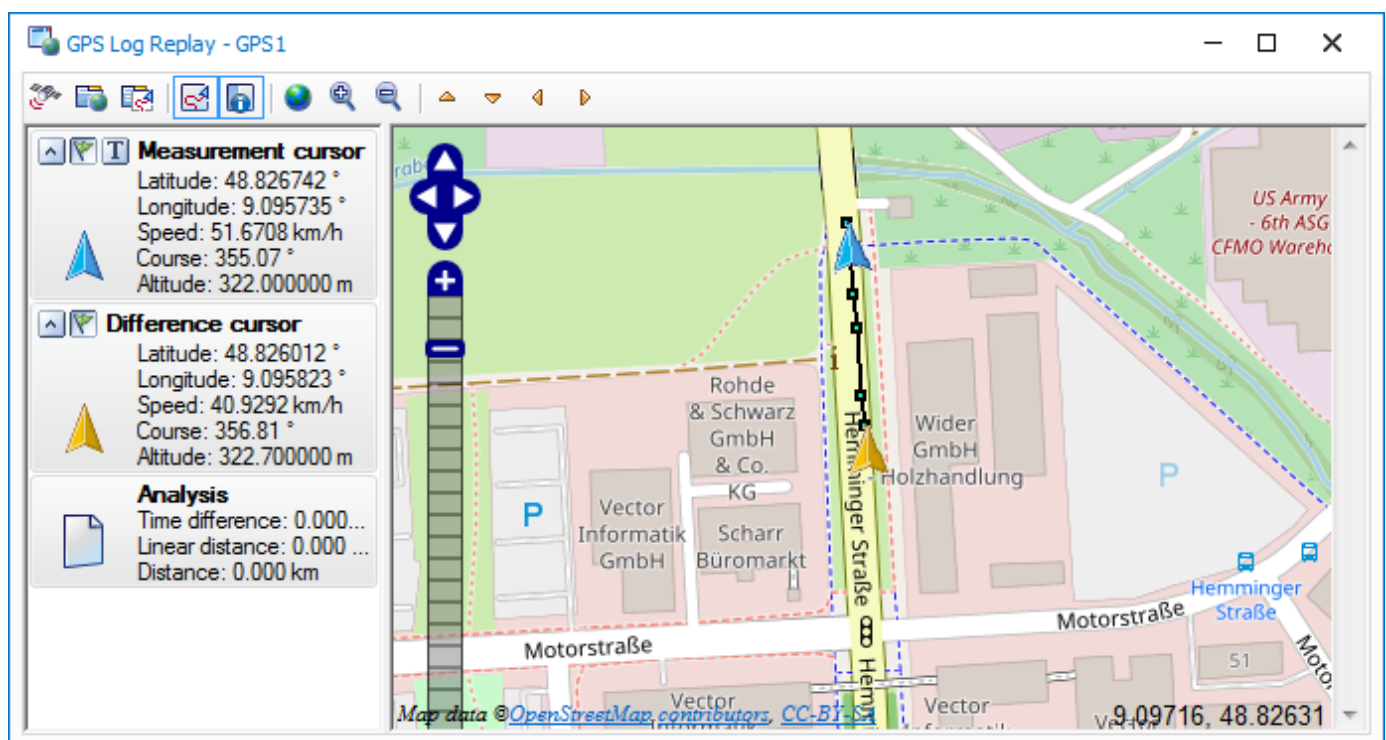


Bild 12: GPS-Fenster

3.11 Trigger und Filter

Trigger und Filter können gezielt auf Busereignisse reagieren und dienen zur Reduktion der angezeigten bzw. aufgezeichneten Daten. Beispiele für Trigger-Bedingungen sind Fehlerzustände, Botschaften, Signale oder auch Signalveränderungen (Flanken). Durch Gruppenbildung und Verknüpfung mit logischen Operatoren können komplexe Systemzustände getriggert werden.

- > **Filter im Messaufbau**
Im Messaufbau stehen verschiedene Filter zur Verfügung, mit denen festgelegt werden kann, welche Daten gezielt zu den jeweiligen Analysefenstern durchgelassen bzw. welche Daten explizit gesperrt werden sollen. Alle Filter können als Sperr- und Durchlassfilter eingesetzt werden.
- > **Trigger im Messaufbau**
Über verschiedene Trigger-Bedingungen kann im Messaufbau das Aufzeichnen der Daten in eine Logging-Datei beeinflusst werden.

> Filter im Trace-Fenster

Im Trace-Fenster können während und nach der Messung über verschiedene Filter Daten zur Analyse reduziert werden. Sie können z.B. auf vordefinierte Filter zugreifen, auf einzelne Signale und Signalwerte filtern oder aber auf verschiedene Spalten-Filter setzen.

> Filter in der Hardware

Über die Akzeptanzfilterung steuern die CAN Controller welche empfangenen Botschaften an CANoe durchgereicht werden.

3.12 Logging/Replay

Für eine Analyse nach der Messung können in CANoe Daten aufgezeichnet und später wieder abgespielt werden.

> Replay

Mit dem Replay-Block können in einer Logging-Datei aufgezeichnete Messabläufe wiedergegeben werden. Die in der Logging-Datei enthaltenen Botschaften werden in den Datenfluss eingeleitet.

> Logging

Mit dem Logging-Block kann der Busverkehr in den Formaten BLF und ASCII aufgezeichnet werden. Die aufgezeichneten Daten können dann im Offline-Modus oder mit einem Replay-Block wieder abgespielt werden.

4 Stimulation/Simulation

Bei der Entwicklung von verteilten Kommunikationssystemen mit CANoe wird die Restbussimulation inklusive der grafischen Bedien- und Anzeigeoberfläche automatisch mit Hilfe der Datenbasisinformationen erstellt. Das Kommunikationsverhalten dieser Systeme kann vollständig simuliert und analysiert werden. Im Zuge des weiteren Entwicklungsprozesses können innerhalb dieser Simulation einzelne Knoten durch reale Steuergeräte ersetzt werden. Der Zulieferer erhält durch diese Restbus- und Umgebungssimulation eine Entwicklungs- und Testumgebung sowohl für das Gesamtsystem als auch für einzelne Steuergeräte und Module.

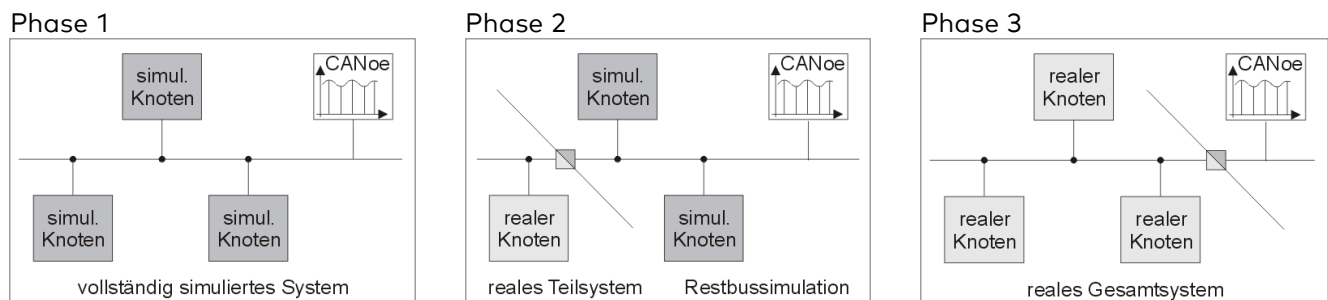


Bild 13: Der Entwicklungsprozess mit CANoe von der Netzwerksimulation bis zum realen Gesamtsystem

Highlight

- > Einfaches Starten und Stoppen von Simulationen und Stimulationen über die Statusleiste.

4.1 Variablen und Generatoren

Systemvariablen stehen für alle Simulations- und Analyseblöcke, Panels und für die I/O-Hardware-Integration zur Verfügung. Sie dienen systemweit dem Austausch von Konfigurationsparametern, Messgrößen oder zur Anbindung externer Programme über die COM-Schnittstelle.

Für die Stimulation der Restbussimulation oder einer angeschlossenen I/O-Hardware können Signale, Umgebungs- oder Systemvariablen direkt mit Signalgeneratoren verknüpft werden. So ist es leicht möglich, Signalkurven wie Rampen oder Sinusverläufe in das System einzuspielen. Es ist zudem möglich, aufgezeichnete Signalverläufe aus Logging-Dateien zu extrahieren und als Generatortyp zu verwenden.

4.1.1 Interaktiver Generator

Mit dem Interaktiven Generator (IG) können Botschaften gesendet sowie die zugehörigen Signalwerte gesetzt werden. Damit kann auf einfache Weise eine Restbussimulation erstellt werden.

> Botschaften definieren

Botschaften können in einer Sendeliste manuell oder über eine Datenbasis konfiguriert werden. Die Eigenschaften der Botschaften können angepasst werden.

> Botschaften senden

Die Botschaften, die in der Sendeliste konfiguriert sind, können zyklisch über eine bestimmte Schaltfläche oder über einen beliebigen Tastendruck versendet werden.

> Signalwerte ändern

Im Interaktiven Generator können die Rohdaten jeder Botschaft in der Signalliste geändert werden. Sind für eine Botschaft Signale definiert, können mit Hilfe des integrierten Signalgenerators Signalverläufe (Signalkurven) definiert werden. Rohdaten und Signalwerte werden dann einfach mit der entsprechenden Botschaft auf den Bus gesendet, um z.B. die Reaktion eines Steuergerätes zu prüfen.

> Schicht-7-Protokolle

Je nach Bussystem unterstützt der Interaktive Generator einfache Schicht-7-Protokolle (z.B. J1939 oder GMLAN) und das Versenden von Multiplexing-Botschaften.

Row	Send	Trigger	Name	ID	Channel	Type	DLC
1	▶	Manual		123	CAN 2	CAN Data	8
2	▶	Manual	PowerTrain::ABSdata	C9	CAN 2	CAN Data	6
3	▶	Manual, on key: a	PowerTrain::EngineData	64	CAN 2	CAN Data	8
4	▶	Manual, on key: s	PowerTrain::EngineStatus	65	CAN 2	CAN Data	1
5	▶	Periodic: 50 ms	PowerTrain::GearBoxInfo	3FC	CAN 2	CAN Data	1
6	▶	Manual	PowerTrain::Ignition_Info	67	CAN 2	CAN Data	2

Name	Generator Co...	Generator Type	Raw Value	Raw Step	Phys Value	Phys Step	Unit	Start Bit	Length
EcoMode		User Defined	0	1	0	1		6	2
ShiftRequest		Toggle	0	1	Shift_Req...	1		3	1

Bild 14: Interaktiver Generator mit konfigurierten Botschaften und ihren Signalen

4.1.2 Signalgenerator

Mit dem Signalgenerator kann ein Verlauf für Signale und Variablen (Sinus, Rampe, Puls, Werteliste, ...) definiert werden.

> Werte senden

Das Senden der entsprechenden Botschaften wird hier nach einem definierten Sendemodell übernommen. Bei LIN und FlexRay übernimmt ein Schedule-Table das Senden der Botschaften. Bei CAN übernimmt das Senden der Botschaften der Interaction Layer (IL) in Verbindung mit Funktions-Bibliotheken (DLLs).

Der Signalgenerator kann während der Messung gestartet und gestoppt werden.

> Werteverläufe definieren

Es gibt zwei Arten, um Signalgeneratoren zu definieren. Ein Verlauf kann entweder für ein einzelnes Signal/Variable definiert werden oder aber aus einer Logging-Datei geladen werden.

> Unterstützung in Panels

Um Steuergeräte mit Hilfe von Panels automatisch über einen längeren Zeitraum zu stimulieren, können den Signalen/Variablen auf den Panels Signalgeneratoren zugewiesen werden. Die zugewiesenen Signalgeneratoren werden im Panel visualisiert und eventuelle Fehler angezeigt.

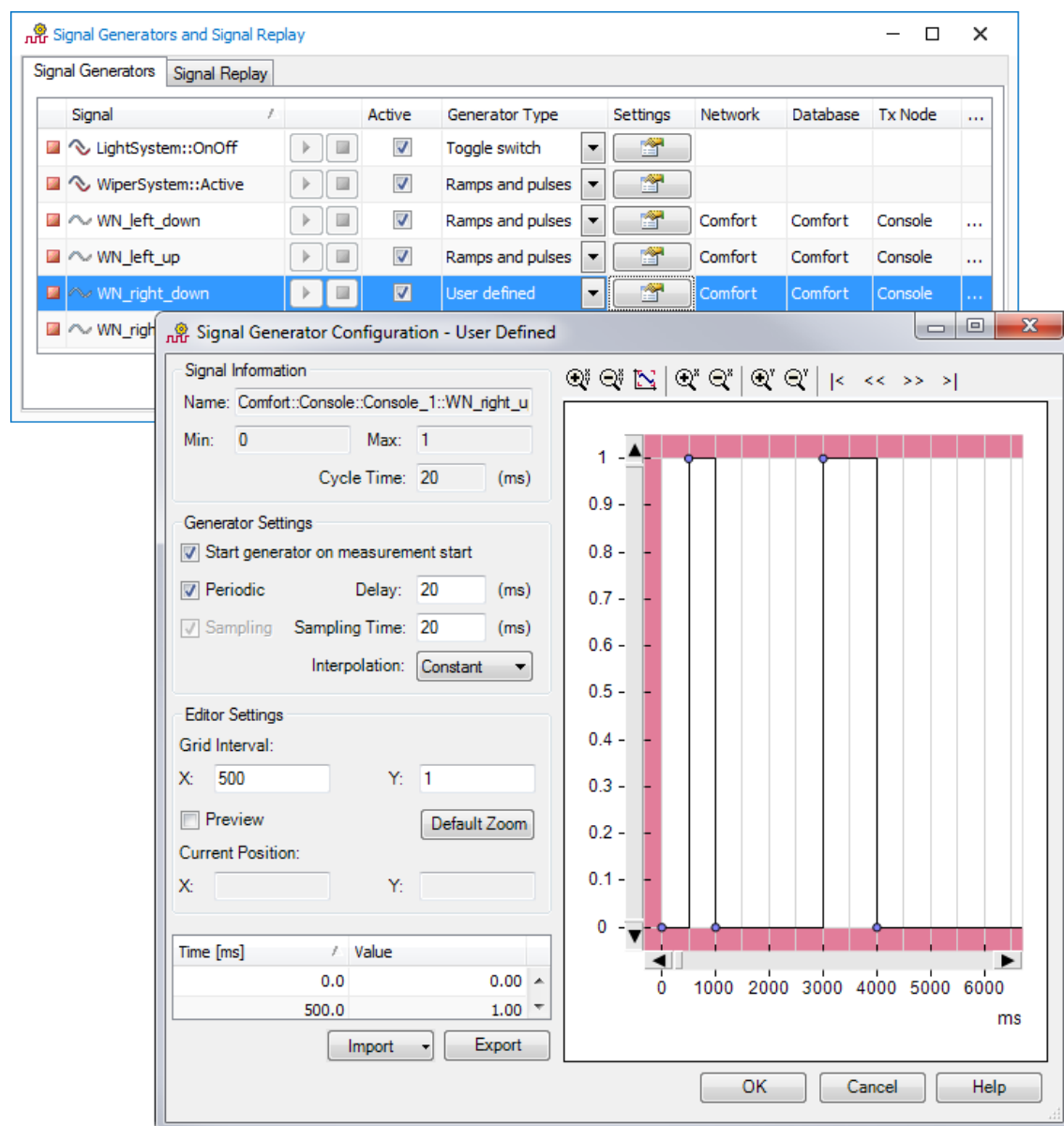


Bild 15: Signalgenerator mit benutzerdefiniertem Signalverlauf

4.2 Startwerte setzen

Im Startwerte-Fenster können für Systemvariablen, Umgebungsvariablen und Signale Werte vorgegeben werden, die bei Messungsstart gesetzt werden.

Die Liste der Startwerte kann in eine Datei exportiert bzw. aus einer Datei geladen werden. Damit kann z.B. eine Simulation bequem mit verschiedenen Sätzen von Startwerten parametrisiert werden.

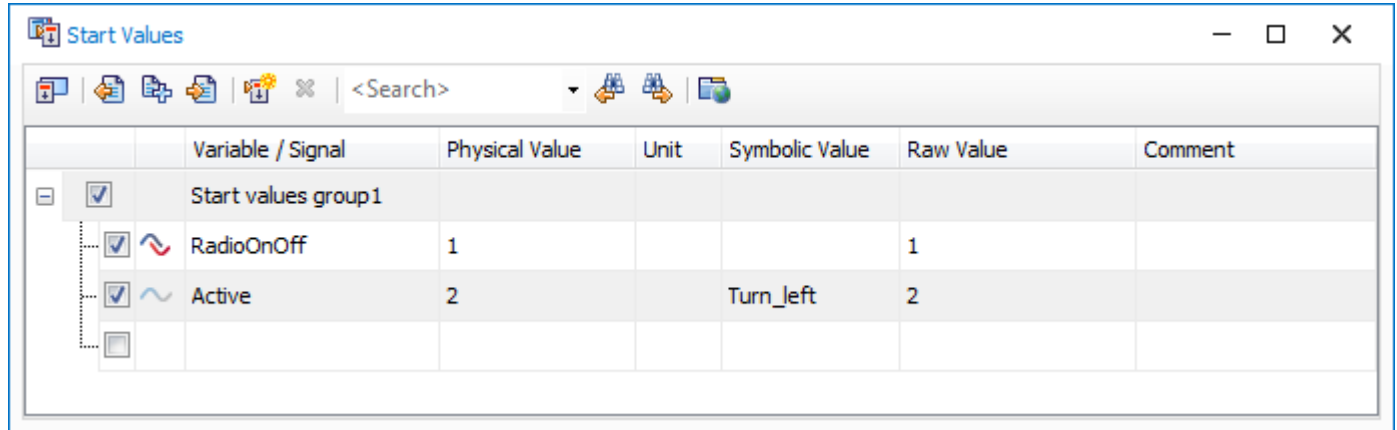


Bild 16: Startwerte-Fenster

4.3 Symbol-Mapping

Mit dem Symbol-Mapping-Dialog können Symbole (Systemvariablen, Umgebungsvariablen und Signale) untereinander verknüpft werden.

Ändert sich zur Laufzeit der Wert der Quell-Variable, wird der Wert der Ziel-Variable automatisch gesetzt. Optional können Sie eine lineare Umrechnungsformel angeben.

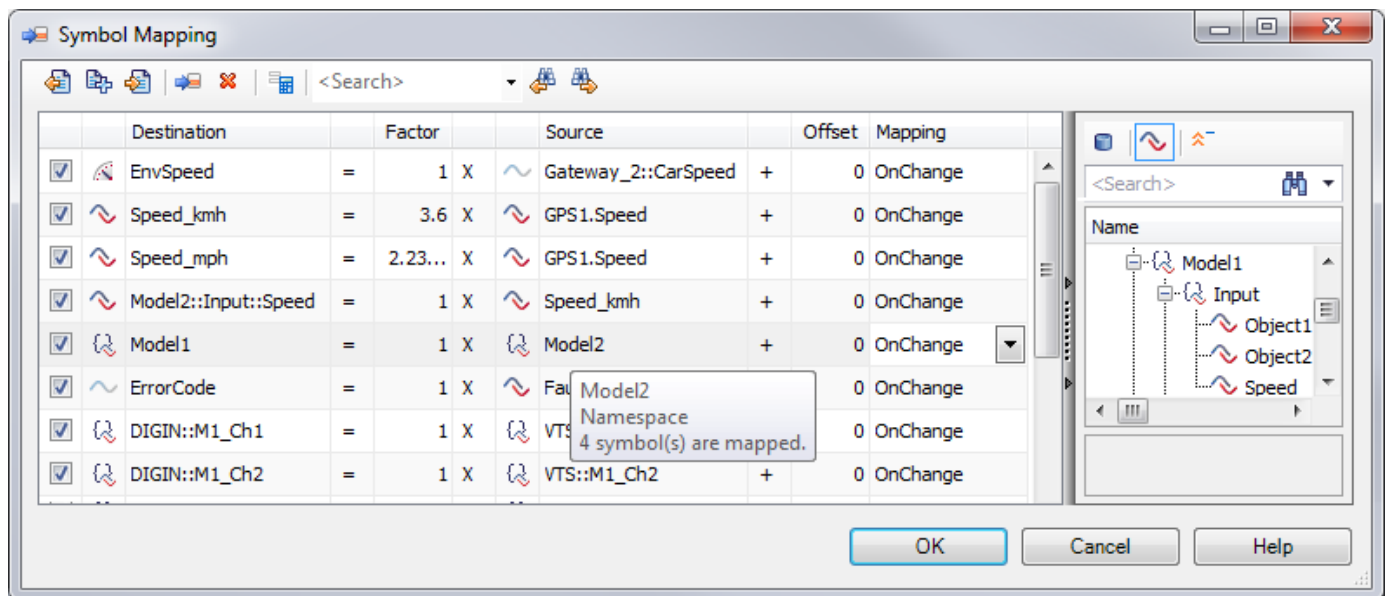


Bild 17: Symbol-Mapping-Dialog

4.4 Interaction Layer, Netzwerk-Management, Transportprotokolle

Beim Erstellen der Restbussimulationen bietet CANoe einen umfangreichen Satz an Protokollbibliotheken. Diese implementieren zum Beispiel die Netzwerk-Management-Funktion nach einem bestimmten OEM- oder AUTOSAR-Standard. Auch der Einsatz standardisierter Transportprotokolle erleichtert den Aufbau von Simulationen oder Testanwendungen, da diese Dienste bereits komplett integriert sind. Über zusätzliche Möglichkeiten können zudem Fehlerfälle reproduzierbar eingespielt und so der entsprechende Steuergeräte-Stack geprüft werden. Die ebenfalls erhältlichen Interaction Layer abstrahieren das Sendeverhalten der Simulationsknoten auf einen Signal-Layer. So können in allen Anwendungen Signale

einfach gesetzt werden und CANoe sorgt dann dafür, dass diese, entsprechend der jeweiligen Sendelogik des OEM, auf den Bus geschickt werden.

CANoe unterstützt die folgenden Protokollstandards:

- > Diagnoseprotokolle: KWP2000 und UDS (ISO 14229)
- > Netzwerk-Management (NM): AUTOSAR, OSEK-NM
- > Transportprotokolle (TP): ISO/DIS 15765-2, CDMT (J1939), BAM (J1939) sowie TPs für MOST, LIN und FlexRay
- > Interaction Layer (IL): Vector IL

4.4.1 OEM-spezifische Erweiterungen

Zusätzlich unterstützt CANoe spezifische TP-, NM- und IL-Erweiterungen für derzeit über 20 OEMs, z.B. BMW, Daimler, VAG, Audi, Ford, Toyota, Fiat, Porsche, Renault.

Mit diesen Erweiterungen ist es möglich, auf einfache Weise eine komplette Restbussimulation einschließlich NM und TP zu erzeugen. Ein manuelles Schreiben von Code – etwa für das Sendemodell – entfällt. Als Basis dient hierbei die korrekt parametrisierte Netzwerkbeschreibungsdatei, in der das komplette Sendeverhalten konfiguriert sowie festgelegt wird, welche Knoten z.B. am NM teilnehmen. Der Model Generation Wizard kann nun aus einer solchen Beschreibungsdatei eine komplette Restbussimulation für CANoe generieren.

Wenden Sie sich bitte an Ihren Vector Vertrieb für weiterführende Informationen.

4.4.2 Interaction-Layer-Konfiguration

Im Konfigurationsdialog des Interaction Layer (IL) können Sendarten übersichtlich angezeigt und bearbeitet werden.

Der Dialog wertet die Sendartenbeschreibung eines Interaction Layers (Vector Interaction Layer, OEM Interaction Layer) aus der Datenbasis aus und zeigt diese an. Dabei werden die OEM-spezifischen Varianten berücksichtigt.

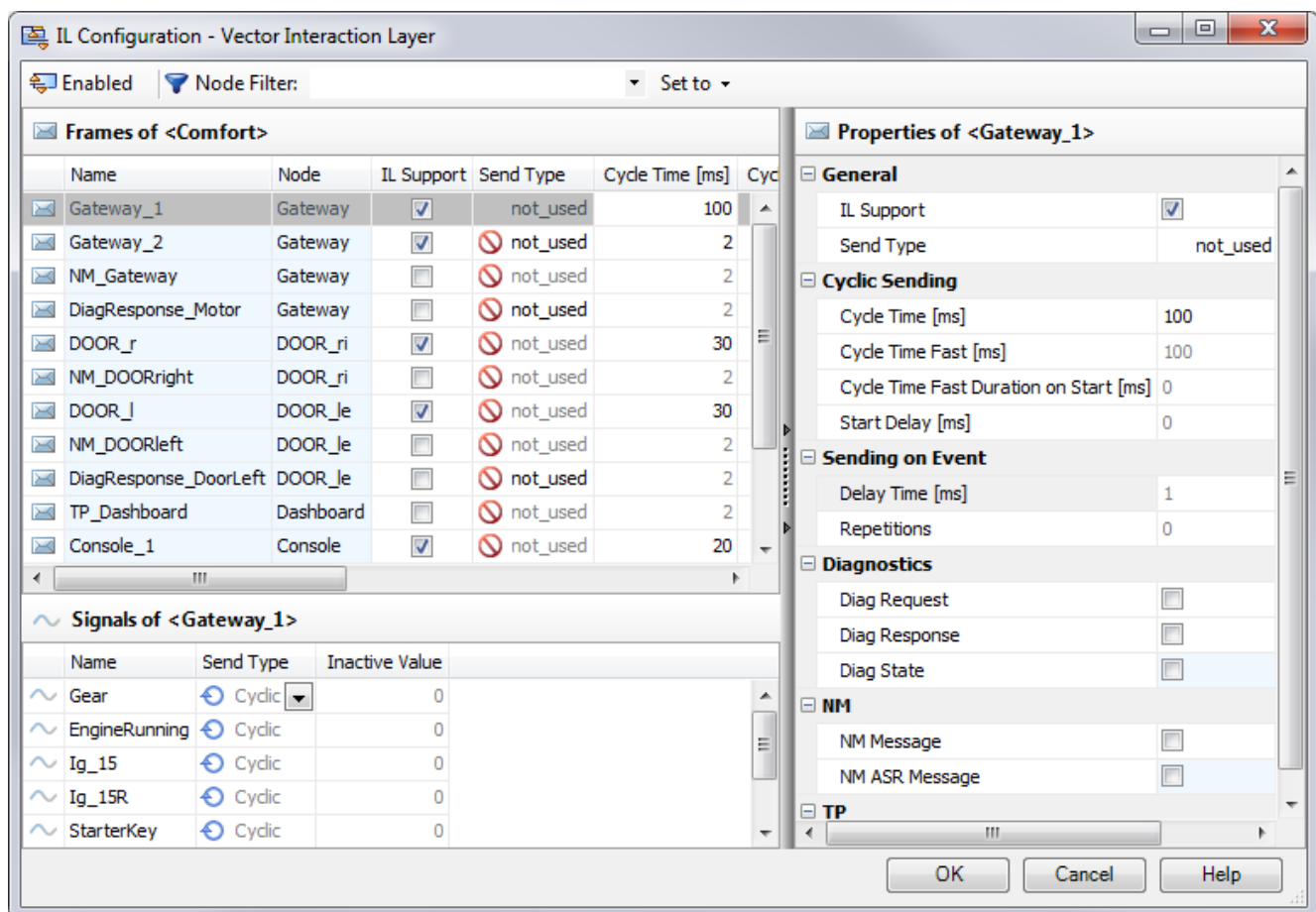


Bild 18: Konfigurationsdialog Interaction Layer

4.5 MATLAB/Simulink

Die CANoe MATLAB-/Simulink-Integration wird von Entwicklungsingenieuren für das Funktions- und Applikations-Prototyping, das Integrieren komplexer MATLAB-Modelle in CANoe Tests und Simulationen sowie für das Entwickeln von Regelungsalgorithmen in Echtzeitanwendungen eingesetzt. Dabei kommunizieren CANoe und die Simulink-Modelle direkt über Signale und Systemvariablen.

Die CANoe MATLAB-/Simulink-Integration unterstützt dabei drei unterschiedliche Ausführungsmodi

- > Im **HIL- oder Online-Modus** wird aus dem Simulink-Modell Code generiert, der als DLL an einen simulierten Knoten in CANoe hinzugefügt wird. Das Modell wird mit CANoe Echtzeit gerechnet. Für nachträgliche Änderungen an den Modellparametern ohne erneutes Kompilieren können automatisch generierte Systemvariablen verwendet werden.
- > Im **Offline-Modus** sind beide Programme gekoppelt. Simulink stellt die Zeitbasis dar und CANoe befindet sich im Slave-Modus. Das komplette System arbeitet simuliert. Ein Zugriff auf reale Hardware ist nicht möglich.
- > Der **Synchronized-Modus** arbeitet ähnlich wie der Offline-Modus. Allerdings stellt beim Synchronized-Modus CANoe die Zeitbasis, die von der angeschlossenen Hardware abgeleitet ist. Somit ist der Zugriff auf reale Hardware bei diesem Modus möglich. Eine Einschränkung ist, dass das Simulink-Modell schneller als Echtzeit gerechnet werden muss, da bei diesem Modus die Simulink-Simulation verlangsamt wird, um die Gesamtsimulation an die CANoe Simulationszeit anzupassen.

4.5.1 Weitere Funktionen der CANoe MATLAB-/Simulink-Integration

- > Der **Model Viewer** zeigt die eingebundenen Simulink-Modelle und Stateflow-Diagramme an und ermöglicht so einen genauen Einblick in die Modellstruktur ohne MATLAB. Eine Navigation durch die Modellstruktur ist möglich.

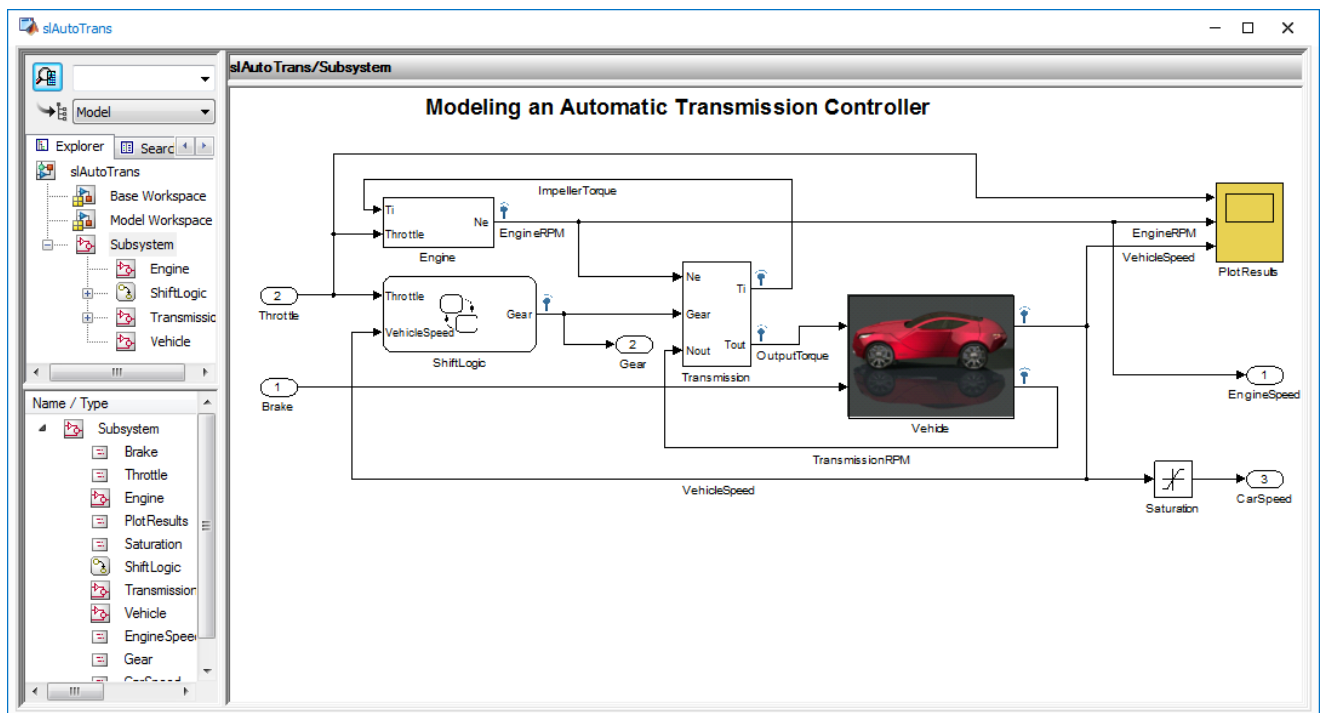


Bild 19: Model Viewer

- > Für eine Modifizierung der Modellparameter kann das Modell über **XCP on CAN** oder **XCP on Ethernet** kalibriert werden während das Modell autark im Standalone-Modus läuft. Hierzu wird bei der Code-Generierung eine A2L-Datei erzeugt.

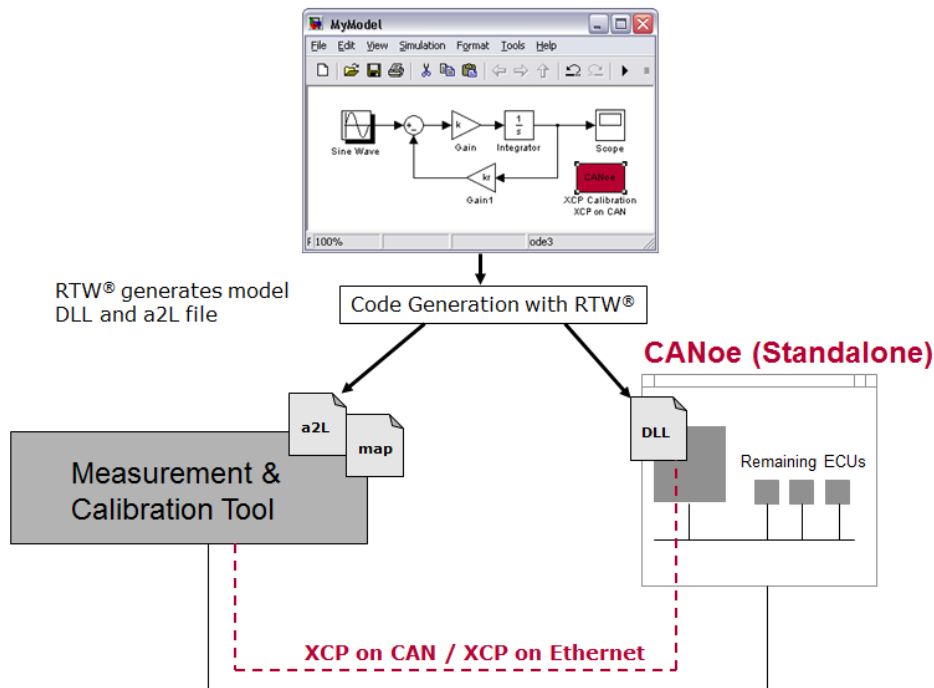


Bild 20: Workflow-Kalibrierung

- > Aufruf von CAPL-Funktionen aus einem Simulink-Modell
- > Ein Simulink-Subsystem mit einer CAPL-Funktion triggern
- > In Simulink auf Änderungen von Umgebungsvariablen reagieren
- > Verwendung von Simulink-Modellen im Analysezeitpunkt von CANoe. Hier können die speziellen Möglichkeiten zur Datenanalyse von Simulink verwendet werden.

4.5.2 Weiterführende Informationen

In der Application Note **AN-IND-1-007_Using_MATLAB_with_CANoe** wird die Verwendung von MATLAB/Simulink zusammen mit CANoe beschrieben. Sie beinhaltet die Grundsätze der CANoe/MATLAB-Schnittstelle und bietet einen Überblick über die verschiedenen Anwendungsfälle.

5 Test

5.1 Test von Steuergeräten und Netzwerken

Eine der Hauptanwendungen von CANoe ist der Test von Steuergeräten und Netzwerken. Dabei werden beispielsweise einzelne Entwicklungsschritte verifiziert, Prototypen geprüft oder Regressions- und Konformitätstests durchgeführt. CANoe bedient das **System under Test** hierbei an allen Schnittstellen. So wird eine möglichst vollständige Testabdeckung gewährleistet.

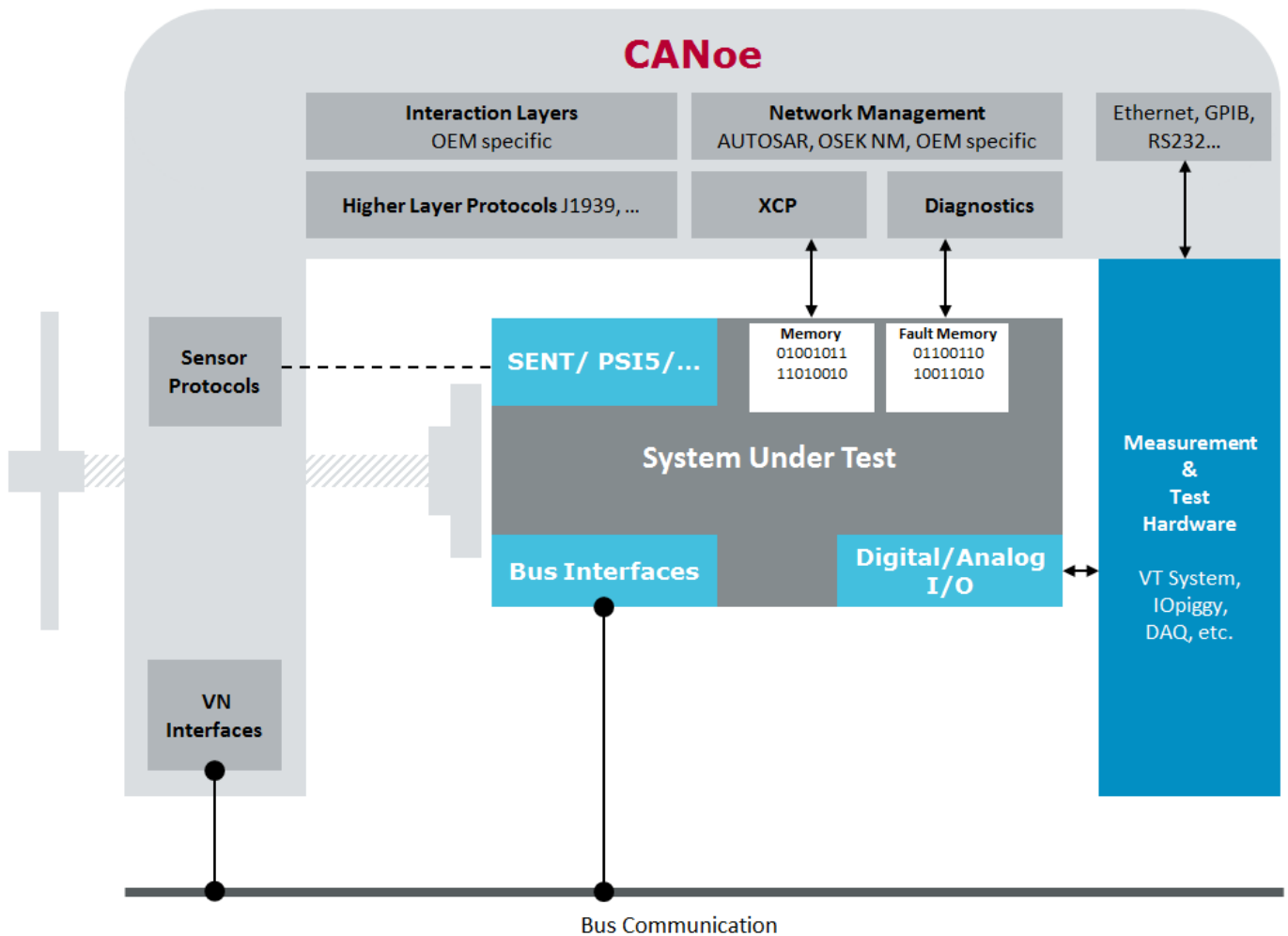


Bild 21: Testen mit vollständigem Zugang zum Steuergerät

Damit Ihre Testaufgaben einfach und flexibel umgesetzt werden können, besteht das Test Feature Set aus folgenden Bestandteilen:

- > Das Umsetzen sequentieller Testabläufe für CANoe erfolgt in **Testmodulen/Test Units**, die in Testgruppen und Testfälle untergliedert sind. Die einzelnen Module lassen sich während einer Messung jederzeit ausführen.
- > Test Units bestehen aus einer Menge an Dateien, welche die Testimplementierung beinhalten (z.B. CAPL-Dateien, C#-Dateien, Testtabellen, Testdiagramme und Parameterdateien). Innerhalb einer Test Unit können beispielsweise die Tests für eine bestimmte Funktionalität des Steuergeräts implementiert sein. Test Units werden in sogenannten Testkonfigurationen verwaltet. Eine Testkonfiguration kann 1...n Test Units enthalten, die sequentiell nacheinander ausgeführt werden. Eine CANoe Konfiguration wiederum kann beliebig viele Testkonfigurationen enthalten, die auch parallel ausgeführt werden können. Ausführbare Test Units können mit vTESTstudio erstellt werden. vTESTstudio ist ein separates Produkt.

- > Testmodule können in CAPL, XML oder .NET beschrieben werden.
In XML-Modulen werden Tests aus vordefinierten Test Pattern zusammengesetzt und einfach über Eingangs- und Ausgangsvektoren parametrisiert. CAPL- und .NET-Testmodule hingegen werden programmiert und weisen dadurch eine sehr flexible Testablaufsteuerung auf. Das Entwickeln der .NET-Testmodule erfolgt komfortabel in C# oder VB.NET. Je nach Anforderung lassen sich die unterschiedlichen Beschreibungsformen kombinieren.
Testmodule werden in Testumgebungen verwaltet. Die Testumgebungen enthalten Testmodule sowie weitere Funktionsblöcke für die Testdurchführung und werden unabhängig von der Systemkonfiguration gespeichert. Somit lassen sie sich leicht in unterschiedlichen Projekten verwenden.
Hinweis: Die Beschreibungsform der Test Units wird derzeit priorisiert weiterentwickelt. Neue Projekte sollten darum nach Möglichkeit Test Units statt Testmodule verwenden, um von den künftigen Weiterentwicklungen auch optimal profitieren zu können.
- > Parallel zur Testausführung können weitere Systemzustände, wie das Einhalten von Zykluszeiten einzelner Botschaften, überprüft werden. Diese Nebenbedingungen werden automatisch in die Testbewertung aufgenommen.
- > Die **Test Service Library** enthält eine Sammlung von vorbereiteten Prüffunktionen, die den Aufbau von Tests erleichtern. Sie werden in den Testmodulen verwendet und über die Datenbasis parametrisiert. So können zum Beispiel Zykluszeiten von Botschaften, die Reaktionszeit eines Steuergeräts auf den Empfang einer Botschaft bis zum Versenden der Antwortbotschaft oder die Gültigkeit von Signalwerten und Diagnoseparametern überwacht werden. Für das Beurteilen der Qualität der getesteten Steuergeräte werden verschiedene Statistikwerte der Tests ausgegeben, unter anderem die Anzahl der gemeldeten Abweichungen im Testzeitraum.
- > Beim Ausführen eines Testmoduls bzw. einer Test Unit wird ein umfangreicher **Testreport** erzeugt. Neben den Namen der durchlaufenen Testfälle und den einzelnen Testergebnissen können beispielweise auch benutzerdefinierte Angaben oder automatische Screenshots unterschiedlicher Analysefenster protokolliert werden. Folgende verschiedene Ausgabeformate für den Testreport stehen zur Verfügung:
 - > **VTESTREPORT** (empfohlen)
CANoe schreibt die Ergebnisse in eine effiziente Ablage, die später leicht umfassende und schnelle Analysen erlaubt. Die hierbei erzeugten VTESTREPORT-Dateien können mit dem CANoe Test Report Viewer visualisiert und analysiert werden. Der CANoe Test Report Viewer liegt der CANoe Installation bei, und kann zusätzlich aus dem Vector Downloadcenter bezogen und kostenfrei genutzt werden.
 - > **XML**
CANoe schreibt die Ergebnisse in eine flexibel XML-Datei, die weiterverarbeitet werden kann. Über ein XSLT-Stylesheet wird das Ausgabeformat für den Testreport in eine lesbare Datei überführt.
- > Die direkte Ansteuerung von I/O-Hardware in CANoe ermöglicht es, neben der Buskommunikation auch analoge und digitale Steuergeräteschnittstellen zu verwenden. Neben Standard-I/O-Komponenten bietet hier das Vector **VT System** ein modulares Hardware-System, um Steuergeräte im Automotive-Umfeld umfassend zu testen.

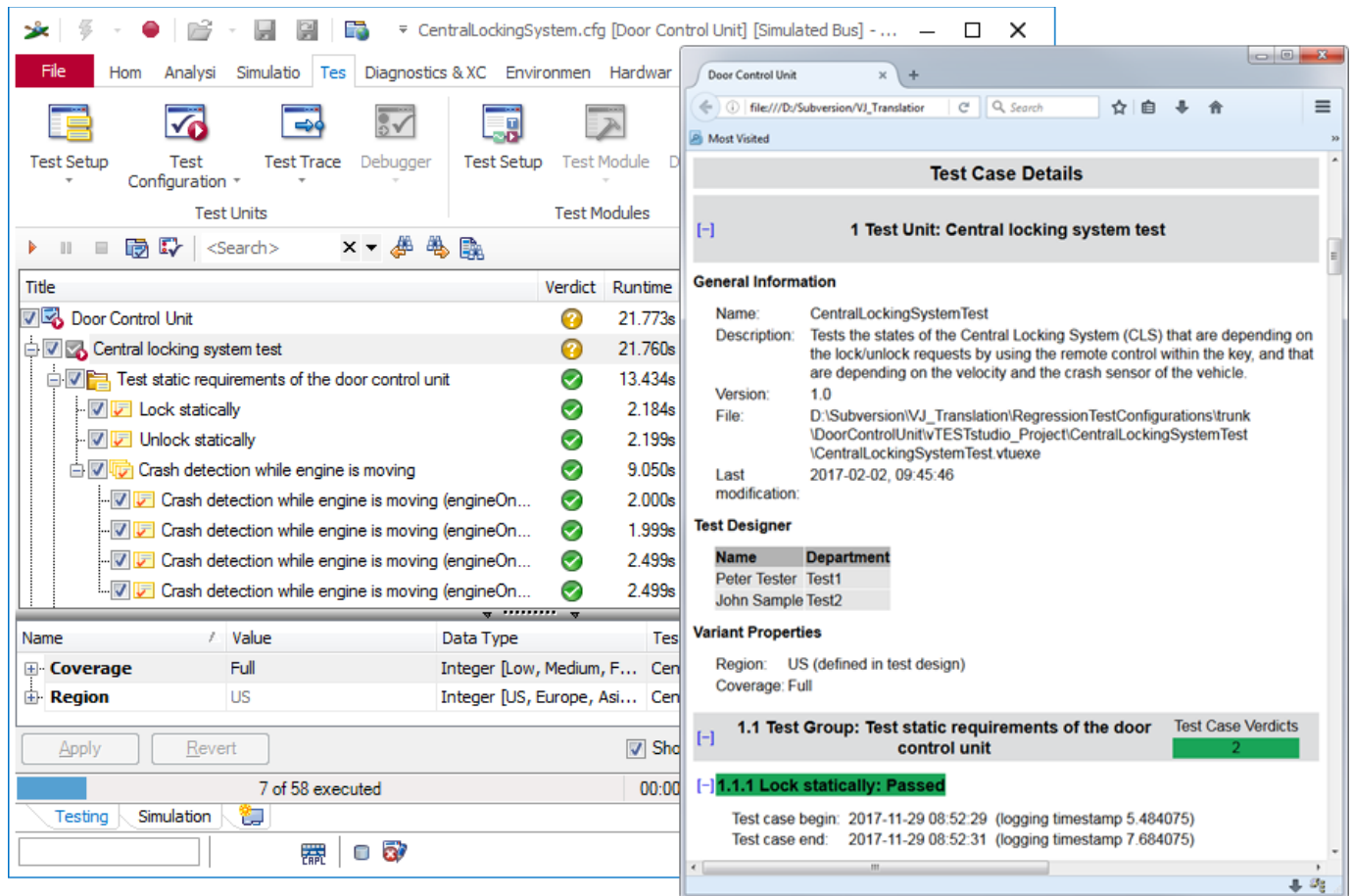


Bild 22: Testablauf für eine Zentralverriegelung mit zugehörigem Testreport in HTML

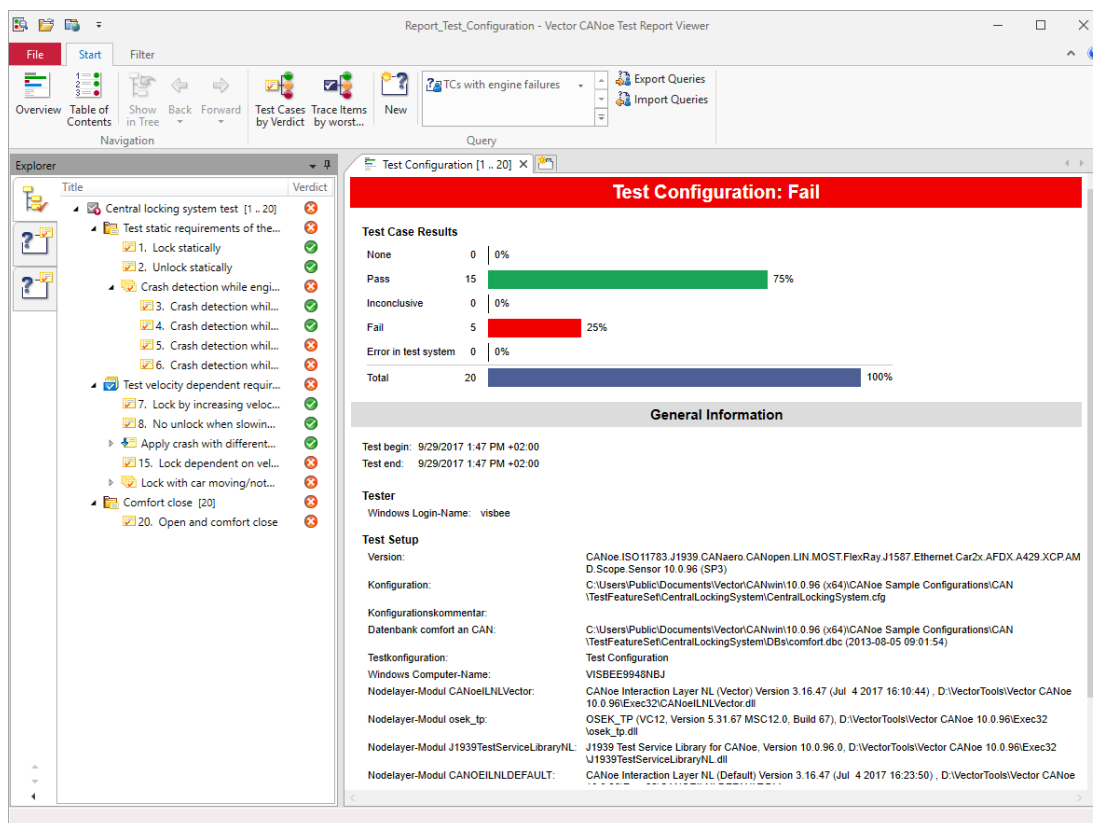


Bild 23: Bedienoberfläche des neuen CANoe Test Report Viewer

5.2 CANoe RT/VN8900 und CAPL-on-Board

CANoe bietet verschiedene Möglichkeiten, echtzeitrelevante Simulations- und Testfunktionen auf einer dedizierten Plattform, also getrennt von der grafischen Oberfläche, auszuführen. Damit wird die Gesamtleistung des Systems bei Bedarf einfach vergrößert und es sind kürzere Latenzzeiten und genauere Timer möglich.

- > Bei CANoe RT erfolgen das Konfigurieren der Simulations- und Testumgebung und das Auswerten auf einem Standard-Computer, während die Simulation und der Test auf einem dedizierten Computer unter Windows Embedded ablaufen.
- > Das Netzwerk-Interface VN8900 beinhaltet eine Prozessorplattform, auf der automatisch die Simulation und der Test ausgeführt werden.
- > Mit dem VN8900 ist auch ein Standalone-Betrieb möglich, d.h. eine CANoe Konfiguration kann auf das Gerät geladen werden und ohne weitere Anbindung an einen GUI-Computer betrieben werden.
- > CAPL-on-Board verbessert die Echtzeitfähigkeit von Vector USB-Hardware-Interfaces z.B. VN1630/40, indem CAPL-Knoten und Visuelle Sequenzen direkt auf der Hardware ausgeführt werden.

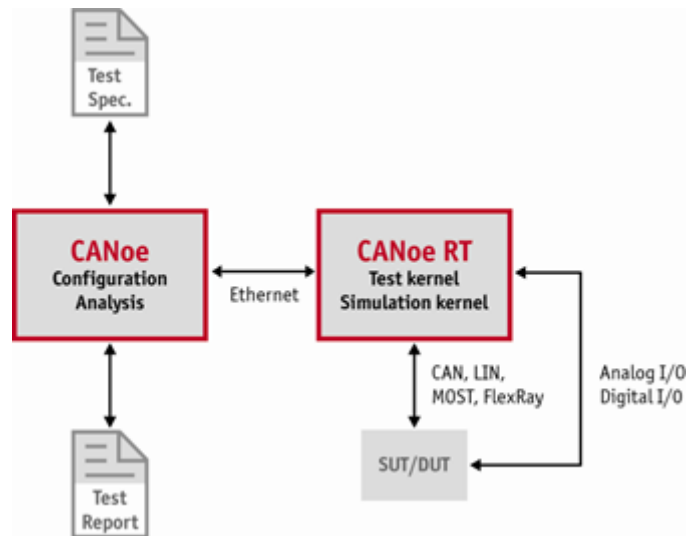


Bild 24: Testsystem auf Basis von CANoe RT und Test Feature Set mit Integration unterschiedlicher Bussysteme sowie digitalen und analogen I/Os

5.3 Vector Tool Plattform

Die Vector Tool Plattform (VTP) bietet verschiedene Dienste an, die in CANoe verwendet werden können. Derzeit besteht VTP aus den beiden Komponenten Smart Device Access (SDA) und Extended Real Time (ERT). Nicht alle Geräte unterstützen dabei automatisch alle Komponenten.

- > Mit der Extended Real Time wird die Latenz und der Determinismus von CANoe verbessert. Um dies zu erreichen, wird das Gerät logisch in zwei Bereiche aufgeteilt. Ein neuer Bereich stellt die Extended Real Time bereit, in der vordefinierte Funktionen unter Echtzeitbedingungen ausgeführt werden können. Mit ERT können für einige Messwerte zusätzliche Transferzykluszeiten von 200µs und 500µs eingestellt werden, um höhere Abtastraten zu erreichen.
- > Der Plattform Manager ist ein Zusatzwerkzeug mit dem alle Geräte der Vector Tool Plattform (VN8800, VN8900, VT6000, CANoe RT Rack) konfiguriert und gesteuert werden können.

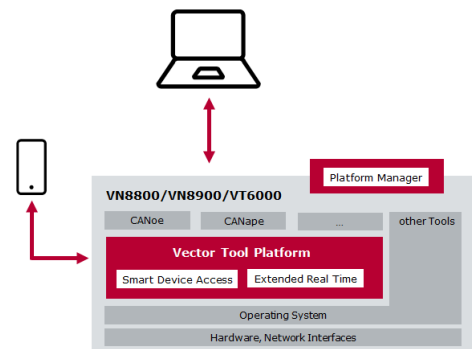


Bild 25: Komponenten der Vector Tool Plattform

5.4 CAN-/CAN-FD-Störungen

Mit dem Hardware Interface VH6501 kann der CAN-/CAN-FD-Bus gestört werden. Für das Auslösen der Störung bietet das Interface verschiedene Trigger-Möglichkeiten für CAN und CAN FD. Es können beliebige Störsequenzen erzeugt werden, z.B. um das Steuergerät in den BUS-OFF-Zustand zu versetzen.

Das VH6501 kann durch eine CAPL API konfiguriert werden, die direkt in CANoe integriert ist. Mit dieser API können die verschiedenen Trigger und Sequenzen konfiguriert werden. Die Trigger-Position für eine Störsequenz kann bei einem Frame Trigger beliebig festgelegt werden.

5.5 Weiterführende Informationen

In der Application Note **AN-IND-1-002_Testing_with_CANoe** werden die grundlegenden Konzepte des Test Feature Sets von CANoe beschrieben.

6 Diagnose

CANoe wird bei der Diagnoseentwicklung in Steuergeräten eingesetzt. Dabei unterstützt CANoe sowohl bei der Implementierung als auch dem Testen der Diagnosefunktionalität, indem es den Zugriff auf die Diagnoseschnittstelle von Steuergeräten bereitstellt. Folgende Konzepte und Funktionen stehen dabei unter anderem zur Verfügung:

- > Unterstützung aller wichtigen Diagnosebeschreibungsformate für KWP2000 und UDS (ISO 14229):
 - > ODX 2.0.1/2.2.0 in Form von PDX-Dateien
 - > MDX 2.0/3.0/4.0
 - > CANdelaStudio (CDD)
- > Anpassungsmöglichkeit der wichtigsten Diagnose-Kommunikationsparameter der Diagnosebeschreibung (Transport- und Diagnoseschicht) mit dem Diagnose/ISO-TP-Konfigurationsdialog
- > Basisdiagnose-Editor zur schnellen Definition einfacher Diagnose-Services, falls keine Diagnosebeschreibung verfügbar ist (für CAN, LIN, FlexRay, Ethernet und K-Line)
- > Interaktiver Diagnosetester mit Diagnose-Konsole, Fehlerspeicher-Fenster und Diagnose-Session-Steuerung mit konfigurierbarer Security-DLL
- > Vorkonfigurierter OBD-II-Tester mit zugehöriger Diagnose-Konsole und Fehlerspeicher-Fenster
- > Unterstützung mehrerer Adressierungsschemas (z.B. normal, extended, normal fixed und mixed) und Adressierungstypen (funktional/physikalisch)
- > Analyse der Diagnosekommunikation auf Service- und Parameterebene (d.h. symbolische Darstellung auf Basis der Diagnosebeschreibung) im Trace-, Daten- und Grafik-Fenster sowie im State Tracker
- > Anzeige von Protokollfehlern im Trace-Fenster
- > Optionale Verwendung von Panels zur Anzeige von Diagnoseparametern und Stimulation von Steuergeräten mittels Diagnose-Requests
- > Simulation der Diagnosefunktionalität von Steuergeräten
- > Spezifikations-/Integrations-/Regressionstests auf Basis des Test Feature Set (Test Units sowie CAPL- und XML-Testmodule) oder mit CANoe .DiVa
- > Zugriffsmöglichkeiten auf alle Schichten der Diagnosekommunikation (CAN-Botschaften, Transportprotokoll und Diagnose-Services) für Gut- und Schlechtfalltests
- > Unterstützung aller wichtigen Netzwerktypen im Automobilbereich (CAN, LIN, FlexRay, Ethernet und K-Line)
- > Gateways zu anderen Netzwerken (z.B. MOST) können durch CAPL-Simulationsknoten oder CAPL DLLs realisiert werden
- > Unterstützung von DoIP (Diagnose über IP), HSFZ (High Speed Fahrzeugzugang) und DoSoAd (Diagnose über AUTOSAR Socket Adapter)
- > Aufzeichnen und Abspielen von Diagnosesequenzen mittels Makros

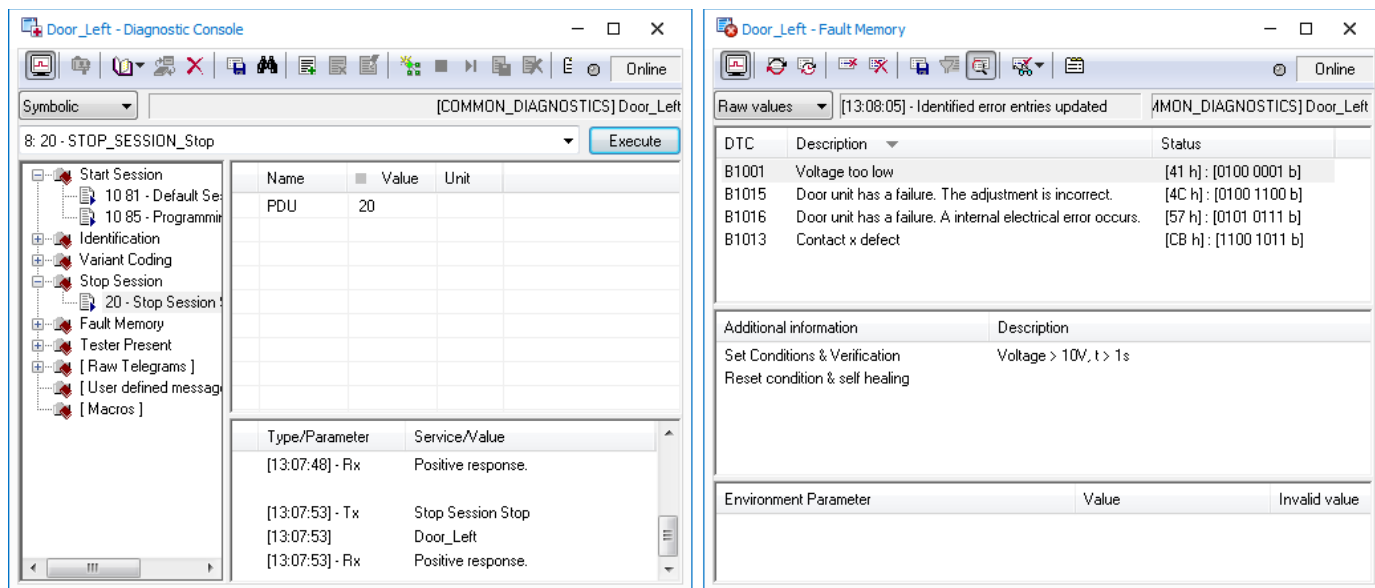


Bild 26: Diagnose-Konsole und Fehlerspeicher-Fenster

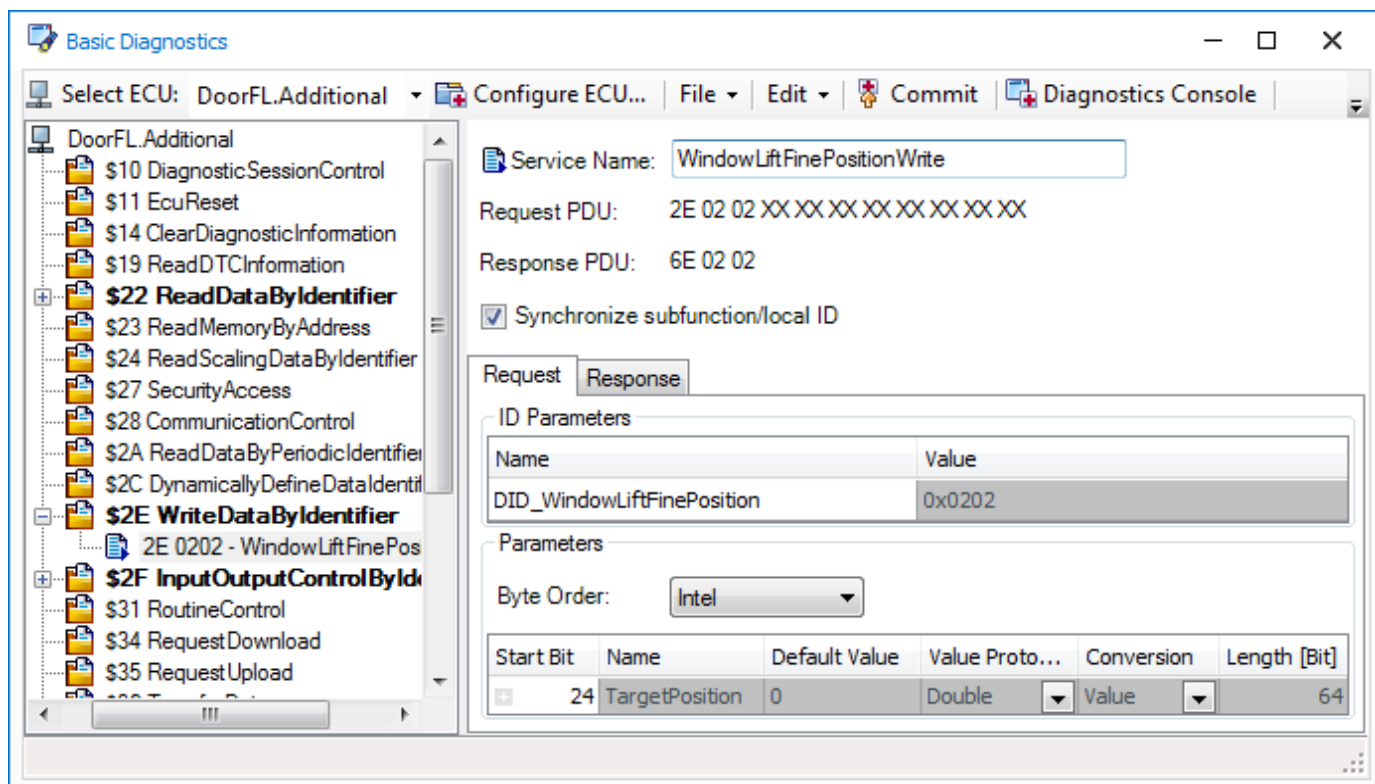


Bild 27: Basisdiagnose-Editor

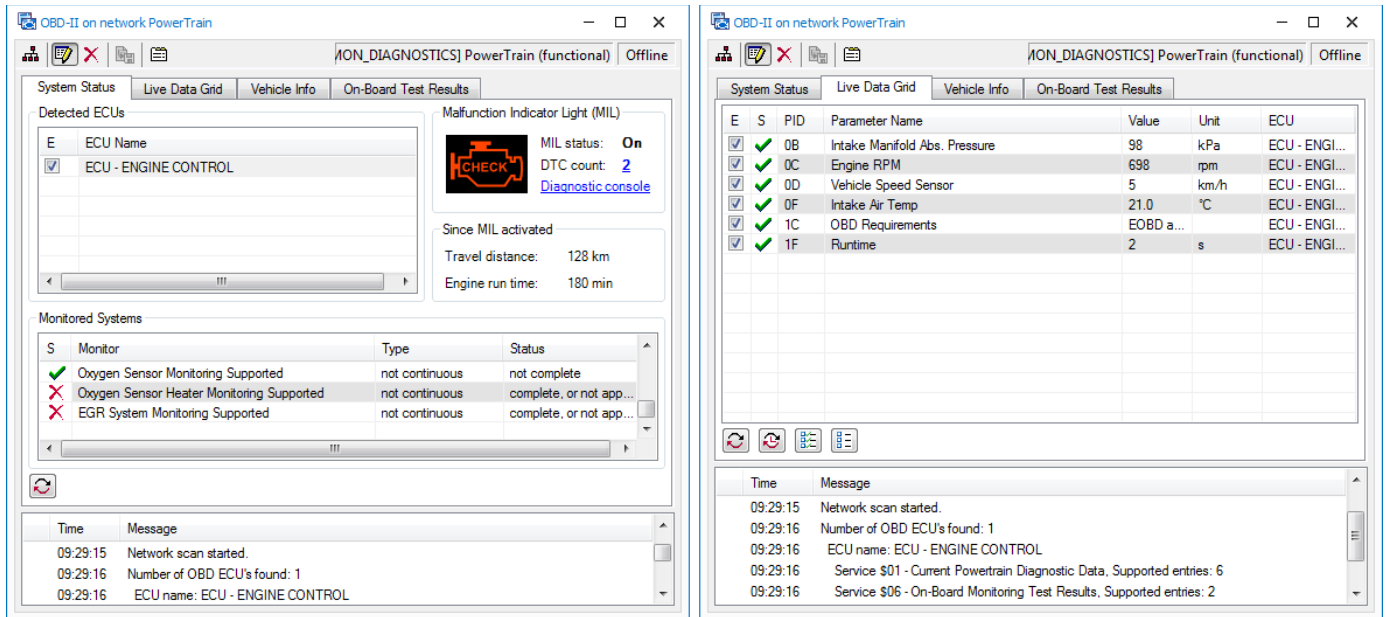


Bild 28: OBD-II-Fenster

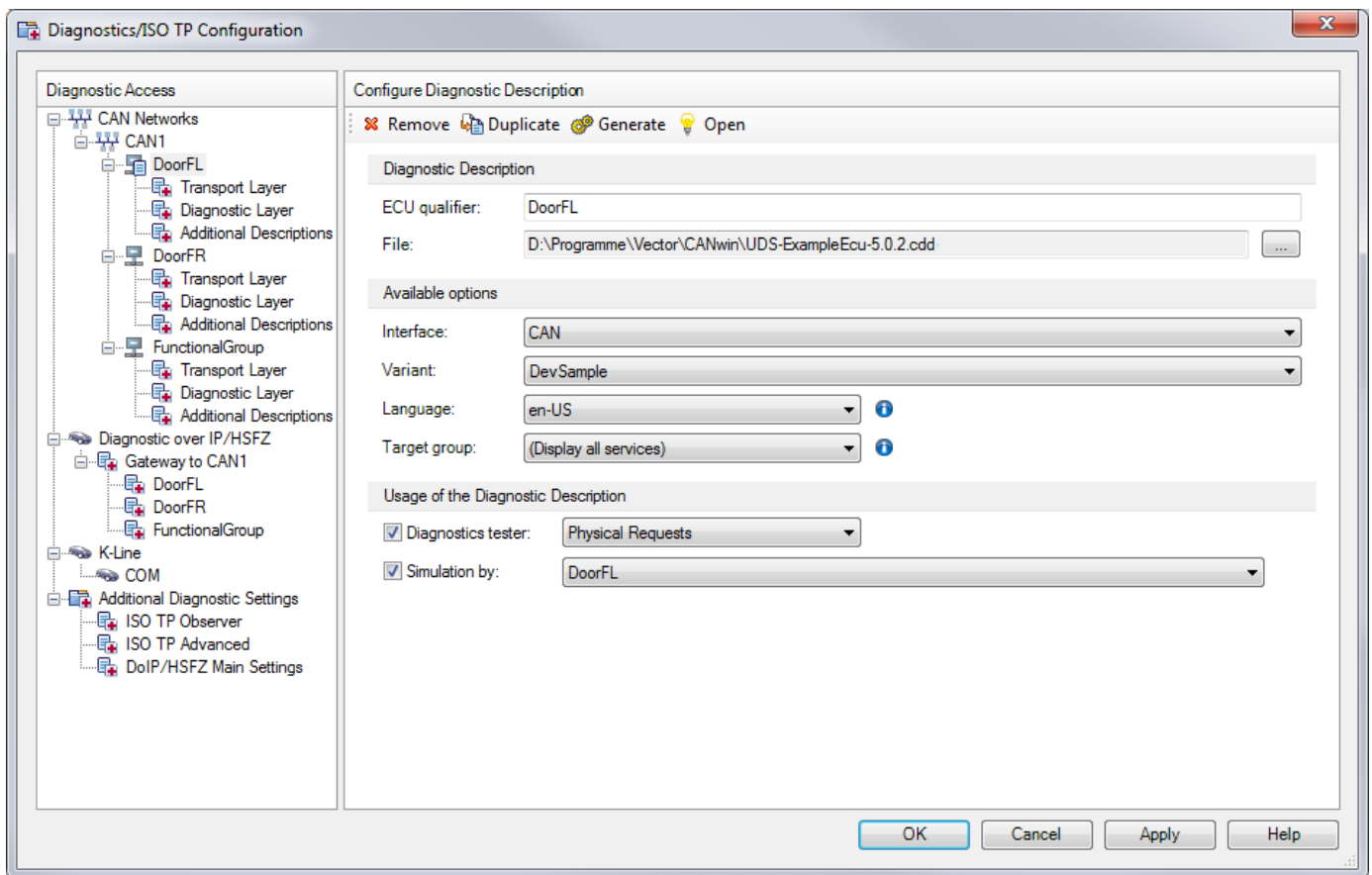


Bild 29: Diagnose-/ISO-TP-Konfigurationsdialog

14.116307	27 01	Seed Level #1 Request::req	req	<tester>	Door	2	27 01
14.117771	67 01	Seed Level #1 Request::pos	pos	Door	<tester>	4	67 01 48 0A
Variant: "CommonDiagnostics"							
SID-PR		0x67	67				
Type		0x01	01				
SecuritySeed		0x480A	48 0A				
0000:		67 01 48 0A	g.H.				
14.142271	27 02	Key Level #1 Send::req	req	<tester>	Door	4	27 02 B7 F5
Variant: "CommonDiagnostics"							
SID-RQ		0x27	27				
Type		0x02	02				
SecurityKey		0xB7F5	B7 F5				
0000:		27 02 B7 F5	'...				
14.143759	67 02	Key Level #1 Send::pos	pos	Door	<tester>	2	67 02
25.552319	19 02	Fault Memory Read (all identified)::req	req	<tester>	Door	3	19 02 00

Bild 30: Darstellung der Diagnosekommunikation im Trace-Fenster

6.1 Weiterführende Informationen

Die Application Note **AN-IND-1-001_CANoe_CANalyzer_as_Diagnostic_Tools** beschreibt eine allgemeine Einführung in die Arbeit mit Diagnose in CANoe/CANalyzer. Es werden grundlegende technische Aspekte und Möglichkeiten mit dem Diagnostic Feature Set aufgeführt. Das Dokument ergänzt die Hilfe von CANoe und kann als Tutorial genutzt werden.

Die Application Note **AN-IND-1-004_Diagnostics_via_Gateway_in_CANoe** beschreibt das Konzept für ein Diagnose-Gateway zwischen CAN und jedem anderen Bussystem oder Transportprotokoll, um das Diagnostic Feature Set von CANoe verfügbar zu machen, wenn ein direkter Zugriff nicht möglich ist.

7 Programmierung

7.1 CAPL-Anbindung

Mit der Programmiersprache CAPL (Communication Access Programming Language) ist der Funktionsumfang von CANoe stark erweiterbar. Die besonderen Merkmale von CAPL sind:

- > Schnell erlernbar durch Anlehnung an die Programmiersprache C
- > Arbeitet vollständig ereignisorientiert. Die Ansteuerung wird von CANoe übernommen.
- > Unterstützt symbolischen Zugriff auf alle Datenbasisinformationen wie z.B. Botschaften und Signale. Signalwerte können direkt in ihrer physikalischen Größe verwendet werden.
- > Der Sprachumfang ist um spezielle Funktionen für eine schnelle Umsetzung der Problemstellung der verschiedenen Einsatzszenarien (Simulation, Test, Diagnose und Analyse verschiedener Bussysteme) erweitert
- > Flexible Erweiterung durch externe Bibliotheken

7.1.1 C-ähnliche Syntax

Es sind die üblichen skalaren Datentypen sowie Arrays davon vorhanden (1, 2, 4 und 8 Byte lange ganzzahlige Typen, sowie ein 8 Byte langer Fließkommatyp). Zuweisungen, Rechenoperatoren sowie Schleifensteuerung entsprechen der C-Syntax.

```
myFunction {
    int counter;

    for ( counter = 0; counter < 8; counter++ ) {
        doSomethingWithCounter ( counter );
    }
}
```

7.1.2 Ereignisorientierte Steuerung

CAPL ist eine ereignisgesteuerte Programmiersprache. Im Gegensatz zu C stehen in CAPL spezielle vordefinierte Event Handler (Ereignisprozeduren) zur Verfügung, die immer dann ausgeführt werden, wenn ein bestimmtes Ereignis eintritt – entweder zeitgesteuert oder durch die Hardware bzw. CANoe-intern ausgelöst.

Hier nur einige Beispiele für diese Event Handler:

Event Handler	Ereignis
On timer Sekundentakt	zeitgesteuert
On message ESPStatus	Botschaftsein- oder -ausgang
On signal update	Signalwert wird erneut beschrieben
On sysvar	Systemvariable wird verändert
On diagRequest	Diagnoseanfrage
On FError	FlexRay-Busfehler erkannt

7.1.3 Symbolischer Zugriff

Der Zugriff auf Signalwerte erfolgt in der Regel mit physikalischen Werten, unabhängig von der Skalierung der Nachrichtenübermittlung. Diese wird in der Datenbasis eingestellt und von dort übernommen.

> Physikalischer Zugriff auf Signalwerte:

```
// Definition of the representation in the database
$EnergyMgmt::BatteryVoltage = 14.1;
```

> Zugriff auf Rohwert eines Signals:

```
// 8 to 18 Volt with 12bit resolution, without range check
$EnergyMgmt::BatteryVoltage.raw = (14.1 - 8) / (18 - 8) * 4096;
```

> Zugriff auf Botschaftsbasis:

```
// Most significant bytes Motorola, with 12bit only the lower 4 bits are used
msg.byte(0) = (msg.byte(0) & 0xF0) | (byte)((14.1 - 8) / (18 - 8) * 4096 / 256) & 0xF;
// Least significant Byte
msg.byte(1) = (byte)((14.1 - 8) / (18 - 8) * 4096) & 0xFF;
output(msg);
```

7.1.4 Anwendungsspezifische Spracherweiterungen

Für alle Anwendungsgebiete von CANoe gibt es zahlreiche Funktionen, die speziell auf die alltäglichen Problemstellungen dieser Themen zugeschnitten sind.

> **Simulation**

Eine komplette Restbussimulation kann mit Hilfe von CAPL erstellt werden. Hierbei wird dem Entwickler Routinearbeit abgenommen. Signale, Botschaften und das Timing-Verhalten der Busse werden in der Datenbasis definiert (z.B. in DBC-, LDF- oder FIBEX-Dateien), die schon oft zur Verfügung gestellt und von zentraler Stelle gewartet und aktualisiert werden. Mit den mitgelieferten Modellierungsbibliotheken (DLLs) können die Datenbasisinformationen genutzt werden, ohne ein Zeile Code zu schreiben. So kann das Senden - zyklisch oder auch nur zu bestimmten Ereignissen - gemäß der Datenbasisvorgaben komplett von CANoe übernommen werden, der Entwickler muss sich lediglich um die eigentliche Funktion, also den Inhalt der Signale, kümmern.

> **Test**

Auch für die Programmierung von automatisierten Tests bietet CAPL komfortable Steuerungsmöglichkeiten, die sowohl die Testdurchführung wie auch die Auswertung unterstützen. Mit wenigen Zeilen Code steht ein Grundgerüst, das den Testablauf und das automatisierte Reporting verwendet. So liefert ein CAPL-Testknoten mit einem kleinen Programm schon mit dem Standardreport eine übersichtliche Zusammenfassung des Testablaufs

```
testcase MyTestCase()
{
    TestCaseTitle("myTC", "My Test Case");
    TestCaseDescription("A test of mine");

    TestCaseComment("first take a short break ");
    TestWaitForTimeout(200);
    If ( MyTestExecution () > 0 )
        TestStepPass("myTC successful");
    else
        TestStepFail("myTC failed");
}

long MyTestExecution ()
{
    /* Own code */
    return 1;
}

MyTest() {
    MyTestCase();
    TestSetVerdictModule(TestGetVerdictLastTestCase());
}
```

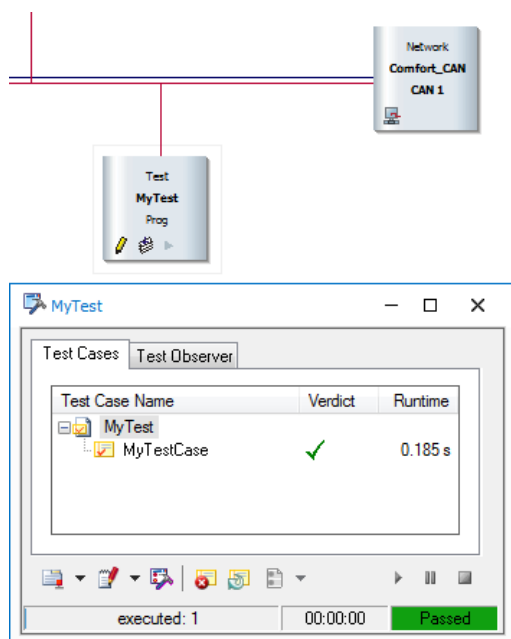


Bild 31: Testknoten mit Testausführungsdialog und Testreport

Test Overview			
Test begin: 2012-09-10 13:49:26 (logging timestamp 3.694054)			
Test end: 2012-09-10 13:49:26 (logging timestamp 3.894054)			
Statistics			
Overall number of test cases	1		
Executed test cases	1	100% of all test cases	
Not executed test cases	0	0% of all test cases	
Test cases passed	1	100% of executed test cases	
Test cases failed	0	0% of executed test cases	
Test Case Results			
meinTC Mein Testcase pass			
Test Module Information			
Test Case Details			
1. Test Case meinTC: Mein Testcase: Passed			
Ein Test von mir			
Test case begin: 2012-09-10 13:49:26 (logging timestamp 3.694054)			
Test case end: 2012-09-10 13:49:26 (logging timestamp 3.894054)			
Test Case Sequence			
Timestamp	Test Step	Description	Result
3.694054		erst eine Kurze Pause	-
3.894054	Resume reason	Elapsed time~200ms (max~200ms)	-
3.894054		MeinTC erfolgreich	pass
End of Report			

> Diagnose

Auch für Anwendungsfälle im Bereich Diagnose kann mit CAPL einfach und effizient ein Programm erstellt werden. Hier eine vereinfachte Implementierung einer Antwort auf eine Diagnoseanfrage:

```
on diagRequest SerialNumber_Read
{
    diagResponse this resp;

    // Set the parameters in the response.
    DiagSetParameter( resp, "SerialNumber", 70499);

    DiagSendResponse( resp);
}
```

> Analyse

CAPL kann auch in der Analyse von Messergebnissen - online und offline - eingesetzt werden. So kann als einfache Aufgabe das Auftreten eines bestimmten Ereignisses gezählt oder auch mit den Inhalten von Signalen Berechnungen durchgeführt werden.

```
On message Brake {
    long TempCounter = 0;
    $BRECounter++;
    // Weighing the average
    TempCounter = $BRECounter;
    if ($BRECounter > 1000)
        TempCounter = 1000;
    @AveragePressure = @AveragePressure * TempCounter + $Brake::Pressure;
    @AveragePressure = @AveragePressure / (TempCounter + 1);
    output ( this );
}
```

7.2 CAPL Browser

Der CAPL Browser ist mehr als ein Editor für CAPL-Programme. Er bietet die Funktionen einer modernen Entwicklungsumgebung, wie:

- > Code-Ergänzung und Syntaxprüfung während des Schreibens
- > Konfigurierbares Syntax-Highlighting
- > Syntaxsensitive Einrückung
- > Einklappbare Funktionsblöcke und Funktionsreferenz in einer Baumansicht zur schnelleren Navigation
- > Suchen und Ersetzen in einzelnen oder mehreren Dateien
- > Hilfe mit Referenzen auf Funktionen
- > Aufruf des Compilers mit direkt anwählbaren Quelltextzeilen im Fehlerfall
- > Hierarchische Funktionsliste mit Suchfunktion zur direkten Übernahme in den Quelltext

Darüber hinaus stehen im CAPL Browser die Objekte aus der CANoe Datenbasis zur Verfügung. Diese werden ebenfalls in einer Baumansicht dargestellt. Im sogenannten **Symbol Explorer** kann auf die folgenden Aspekte der Datenbasis zugegriffen werden:

- > Netzwerksymbole, wie Knoten, Botschaften und Signale
- > Umgebungsdaten, das sind datenbasisspezifische Umgebungsvariablen und die CANoe-weit verwendeten Systemvariablen
- > Alle Diagnosesymbole, wie Requests, Responses und Fehlerspeicher

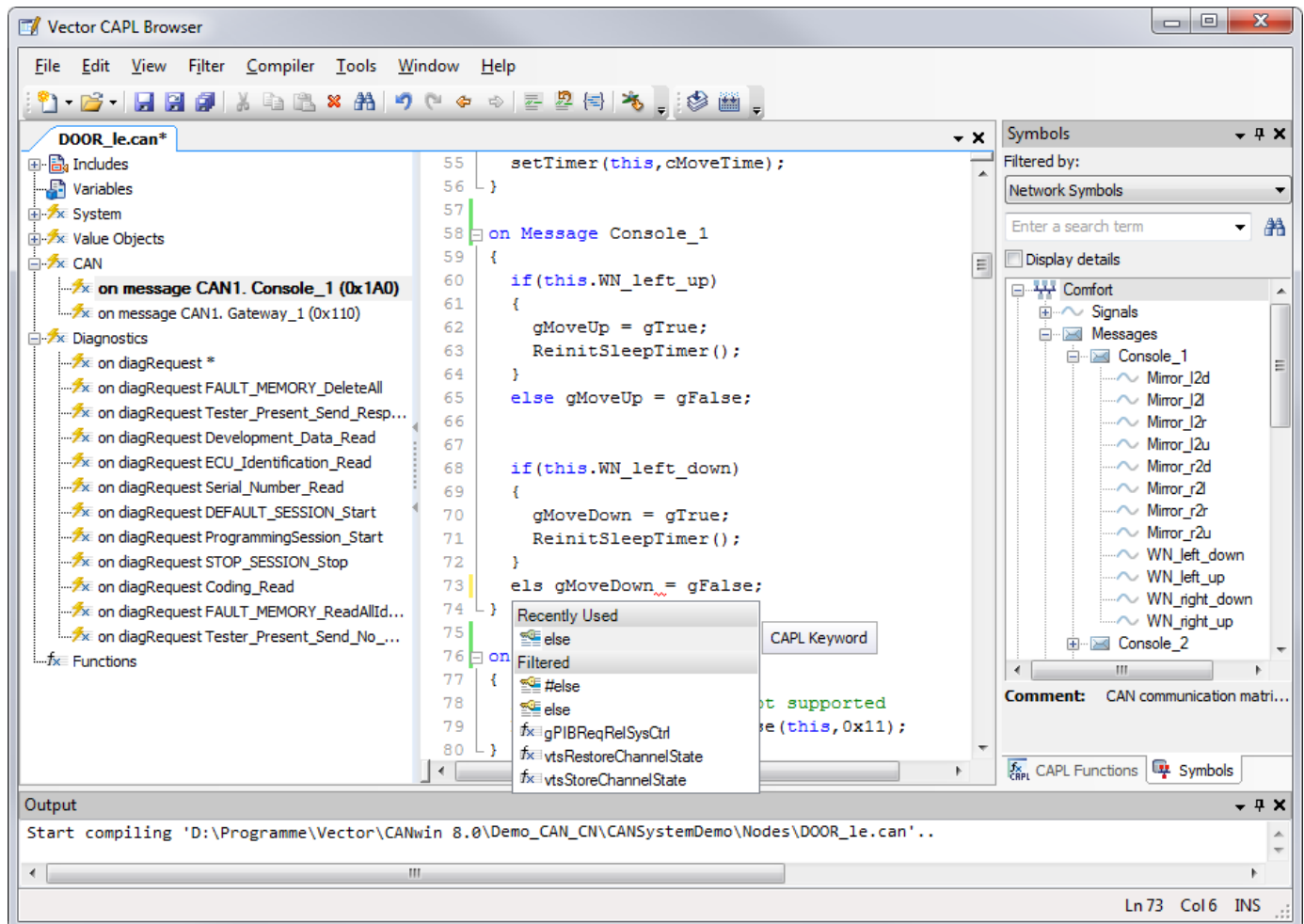


Bild 32: CAPL Browser mit geöffnetem CAPL-Programm, seinen enthaltenen Ereignisprozeduren und in der Datenbasis enthaltenen Netzwerksymbolen

7.3 .NET-Programmierung

In CANoe können an verschiedenen Stellen .NET-Programmiersprachen benutzt werden:

- > für die Programmierung simulierter Netzwerkknoten
- > für die Programmierung von Testmodulen, Testfällen und Testbibliotheken
- > für die Programmierung sogenannter Snippets

CANoe bietet für die .NET-Programmierung ein spezielles API an, das die Sprachen gezielt erweitert (also Embedded Domain Specific Languages erstellt). Als Programmiersprachen werden C# (von Vector empfohlene Programmiersprache), Visual Basic .NET oder J# direkt unterstützt:

- > **Visual Studio als Editor benutzen**
.NET-Programme können z.B. mit Visual Studio 2005, 2008 oder 2010 als Entwicklungsumgebung komfortabel editiert werden. Die Express Editions von Visual Studio sind kostenlos verfügbar.

> Zugriff auf Signale, Umgebungs- und Systemvariablen

Für jedes Signal und jede Umgebungs- oder Systemvariable steht in .NET eine Klasse zur Verfügung, mit der auf den Wert in der CANoe Laufzeitumgebung zugegriffen werden kann.

Beispiel:

```
double value = EnvSpeedEntry.Value;
```

> Zugriff auf CAN-Botschaften

Für jede in der Datenbasis definierte CAN-Botschaft steht in .NET eine Klasse zur Verfügung. Es können Instanzen dieser Klassen angelegt, Signalwerte gesetzt und mit der Methode **Send** der Frame auf den Bus gelegt werden. Außerdem kann mit dem Attribut [OnCANFrame] auf den Empfang von CAN-Frames reagiert werden.

Wird keine Datenbasis verwendet, kann die allgemeine Klasse CANFrame direkt benutzt werden oder von dieser Klasse eigene Klassen abgeleitet werden. Mit dem Attribut [Signal] werden Signale definiert.

> Zugriff auf Diagnose

Sofern eine oder mehrere Diagnosebeschreibungen konfiguriert sind, können in Testmodulen und Snippets über eine .NET-Bibliothek Diagnose-Requests gesendet und Diagnose-Responses empfangen werden. Es ist möglich, Parameter in Requests zu setzen und aus Responses auszulesen.

Die Bibliothek Vector.Diagnostics wird von mehreren weiteren Vector Anwendungen unterstützt, d.h. Diagnoseabläufe lassen sich sehr einfach sowohl in CANoe, als auch in CANape, CANdito oder Indigo wiederverwenden.

> Ereignisprozeduren

Um auf Ereignisse in CANoe zu reagieren, werden Methoden mit speziellen Attributen versehen. Die Methoden werden dann aufgerufen, wenn das Ereignis eintritt (genau wie in CAPL).

7.4 Fehlersuche

Für die Fehlersuche in CAPL- und .NET-Programmen steht der Vector Debugger zur Verfügung. Mit ihm können im Quelltext der Programme Haltepunkte eingefügt und Werte von Variablen geprüft werden.

Im simulierten Modus ist ein Debuggen aller CAPL- und .NET-Programme möglich, da die Simulation zu diesem Zweck angehalten wird. Bei der Verwendung von realer Hardware ist ein Debuggen nur in Programmen von Testmodulen möglich, da die von der Hardware gesendeten Ereignisse weiterhin ausgewertet werden müssen.

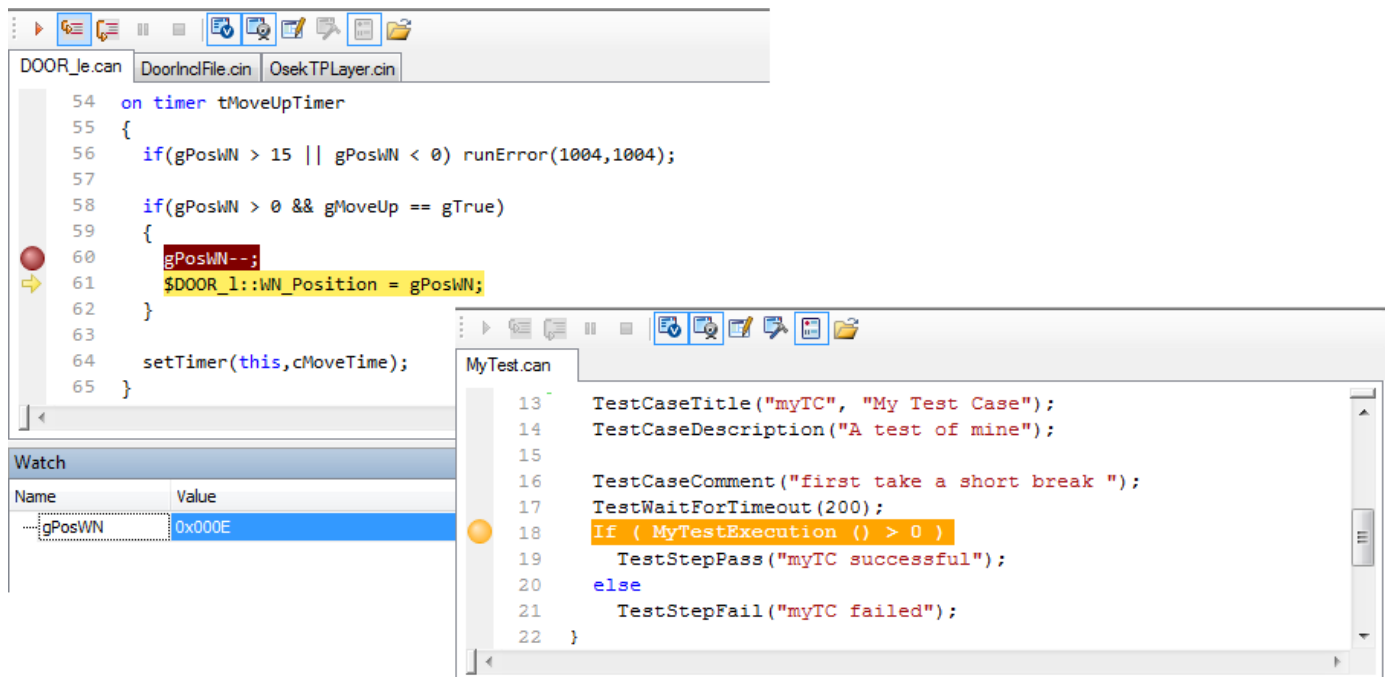


Bild 33: Debugger mit Haltepunkten

7.4.1 Weiterführende Informationen

In der Application Note **AN-IND-1-011_Using_CANoe_NET_API** werden das .NET-API-Konzept und seine Verwendung in CANoe beschrieben. Dabei wird vorausgesetzt, dass der Leser mit dem .NET-Framework und der Application Note **AN-IND-1-002_Testing_with_CANoe** vertraut ist.

7.5 Visual Sequencer

Ablauf-Sequenzen werden schnell und ohne Programmierung grafisch konfiguriert. Innerhalb solcher Sequenzen können z.B. Variablen und Signale gesetzt werden. Auch Frames und Diagnosebefehle lassen sich senden. Zudem besteht die Möglichkeit, auf bestimmte Ereignisse zu warten, Werte zu prüfen oder mit Kontrollstrukturen (Repeat...until) Wiederholungen zu definieren. Somit eignen sich diese Sequenzen bestens zum einfachen Testen von heterogenen Systemen oder zur Stimulation von Steuergeräten.

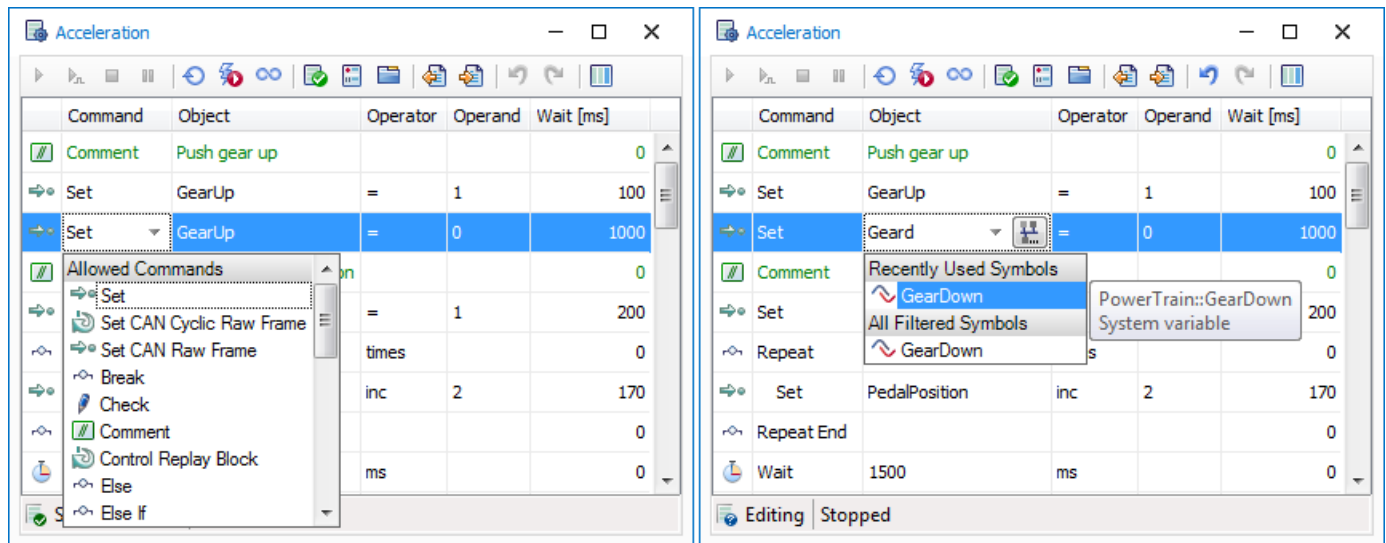


Bild 34: Visual Sequencer zum Erstellen von Test- und Stimulations-Sequenzen. Einfache Auswahl der Befehle und der Datenbasisobjekte mit "Auto-Complete"-Unterstützung sowie detaillierter Anzeige der Datenbasisinformationen

8 Panels

Panels sind grafische Elemente, über die mit sogenannten Steuerelementen (Controls), wie z.B. Schieberegler oder Zeigerinstrumenten, Signal- und Variablenwerte verändert und angezeigt werden können. In CANoe stehen verschiedene Arten von Panels zur Verfügung.

> Signal-Panel

Ein Signal-Panel bietet eine einfache Möglichkeit, Signalwerte zur Messungszeit zu verändern. Bei einem Signal-Panel wird zwischen Knoten- und Netzwerk-Panels unterschieden. Bei der Verwendung eines Knoten-Panels werden automatisch die Tx-Signale des entsprechenden Knotens konfiguriert. Bei der Verwendung von Netzwerk-Panels werden automatisch die Tx-Signale des gesamten Netzwerkes konfiguriert.

> Symbol-Panel

Mit dem Symbol-Panel können Werte von Signalen und Variablen direkt während der Simulation angezeigt und verändert werden.

> Benutzerdefinierte Panels

Mit Hilfe benutzerdefinierter Panels können Oberflächen für verschiedene Einsatzbereiche erstellt werden. Panels werden beispielsweise auch zur Steuerung der Simulations- und Testumgebung oder zur Anzeige der Analysedaten aus CAPL-Programmen verwendet.

Mit dem Panel Designer können solche Panels komfortabel erstellt werden. Beispielsweise werden Symbole einfach per Drag & Drop mit einem Steuerelement verknüpft. Das Konfigurieren der einzelnen Panels und Steuerelemente erfolgt über das ständig geöffnete Eigenschaften-Fenster, und eine ganze Reihe nützlicher Ausrichtefunktionen sorgen für ein optimales Layout.

> Frei programmierbare ActiveX- und .NET-Panels

Diese Panels können in Programmiersprachen wie Visual Basic 6.0, Visual Basic.NET oder auch C# realisiert und in CANoe eingebunden werden.

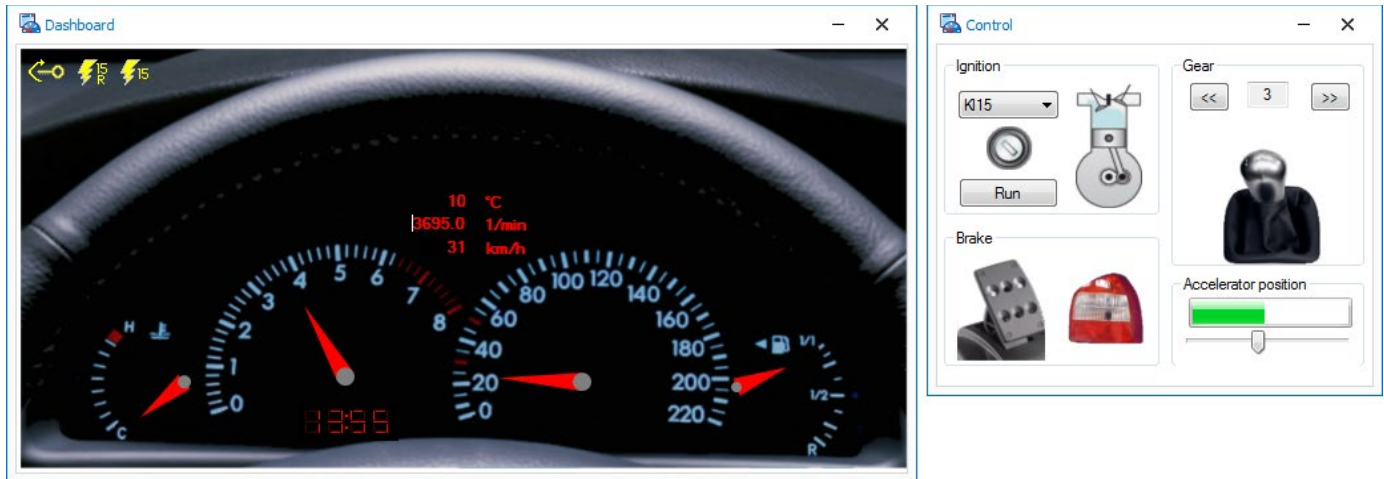


Bild 35: Benutzerdefinierte Panels zur Anzeige von Signal- und Variablenwerten

9 Hardware-Schnittstellen

CANoe unterstützt alle bei Vector erhältlichen Hardware-Schnittstellen. Dank der großen Auswahl unterschiedlicher Computer-Interfaces (PCMCIA, USB 2.0, PCI, PCI-Express, PXI) und der Bus-Transceiver ist für jeden Anwendungsfall der optimale Buszugang möglich.

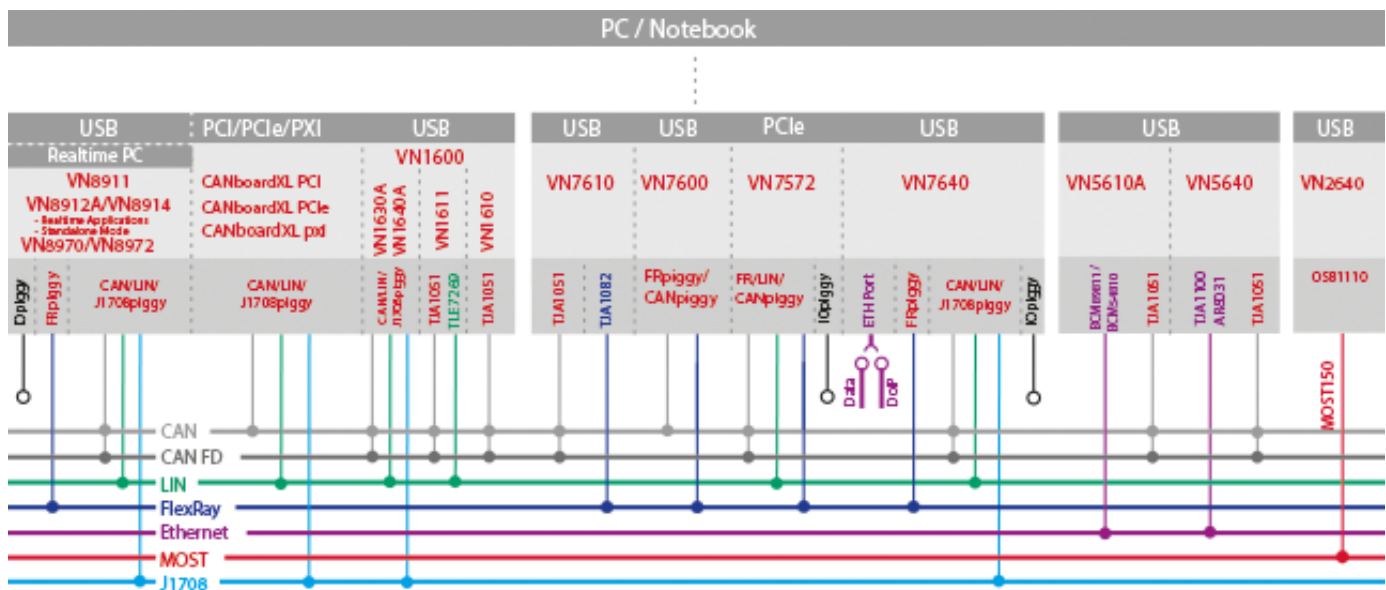


Bild 36: Übersicht über die Vector Hardware

10 Schnittstellen zu anderen Anwendungen

Mit CANoe kann auf den von CANape bereitgestellten ASAM-MCD3-Server – und damit über XCP und CCP – auf Parameterwerte in vorhandenen Steuergeräten zugegriffen werden. Eine direkte XCP-Integration in CANoe bietet die Option .XCP (siehe Kapitel 12).

10.1 COM-Schnittstelle

Der integrierte COM Server (Component Object Model) erlaubt das Steuern des Messablaufs durch externe Anwendungen und einen bequemen Datenaustausch mit Standardsoftware, etwa zur Messdatenanalyse und zur weitergehenden Auswertung des beobachteten Busverkehrs. Häufig eingesetzte Programmier-/Script-Sprachen sind Visual Basic oder Visual Basic for Applications. Aber auch C++/C# kommen häufig zum Einsatz. Die Funktionalität, die CANoe über die COM-Schnittstelle anbietet, umfasst u.a. folgende Bereiche:

- > Steuerung der Simulation, Start und Stopp der Messung

- > Laden von bestehenden Konfigurationen, Erzeugen von neuen Konfigurationen, Hinzufügen von Datenbasen und Blöcken in den Simulationsaufbau
- > Steuerung der automatisierten Tests, Starten der Testausführung, Hinzufügen von Testmodulen
- > Zugriff auf Signale und Systemvariablen, Zugriff auf CAPL-Funktionen, Kompilieren von CAPL-Knoten

Visual-Basic-Script-Beispiel zum Start der Messung:

```
set app = createobject( "canoe.application")
set measurement = app.measurement
measurement.start
set app = nothing
```

Visual-Basic-Script-Beispiel zum Öffnen einer Konfiguration:

```
set app = createobject( "canoe.application")
app.open "D:\PathToMyConfig\myconfig.cfg"
set app = nothing
```

10.1.1 Weiterführende Informationen

In der Application Note **AN-AND-1-117_CANalyzer_CANoe_as_a_COM_server** wird eine allgemeine Einführung in die COM-Server-Funktionalität von CANoe/CANalyzer beschrieben. Es werden die grundlegenden technischen Aspekte und Möglichkeiten aufgeführt und diese anhand von Microsoft Visual-Basic-Beispielen erklärt.

10.2 FDX

CANoe FDX (Fast Data eXchange) ist ein Protokoll, mit dem Daten einfach, schnell und zeitnah zwischen CANoe und anderen Systemen über eine Ethernet-Verbindung ausgetauscht werden können. Das Protokoll erlaubt anderen Systemen den lesenden und schreibenden Zugriff auf Systemvariablen, Umgebungsvariablen und Bussignale von CANoe. Darüber hinaus ist es möglich, Steuerbefehle (z.B. zum Starten und Stoppen der Messung) per FDX an CANoe zu senden oder Statusinformationen zu empfangen.

Bei dem anderen System kann es sich zum Beispiel um ein HIL-System in einem Prüfstand handeln oder um einen Computer, der Daten von CANoe anzeigen soll. Das Protokoll setzt auf dem weit verbreiteten Standardprotokoll UDP (auf Basis von IPv4) auf.

10.3 ASAM XIL API

ASAM XIL standardisiert eine API zur Kommunikation zwischen Test-Automatisierungswerkzeugen und Test-Ausführungssystemen. Es werden verschiedene Typen von Ausführungssystemen unterstützt, z.B. SIL, MIL oder HIL.

Der Datenaustausch mit CANoe erfolgt über vorbereitete .NET Assemblies und einer C# API.

10.4 FMI

FMI (Functional Mock-up Interface) ist eine tool-unabhängige Schnittstelle zum Austausch von Modellen und zur Kopplung von unterschiedlichen Tools. CANoe unterstützt sowohl den Import von solchen Modellen (FMUs) für Co-Simulation, als auch den Export von FMUs. Es werden die FMI-Versionen 1.0 und 2.0 unterstützt.

11 Option .Scope

Die Option .Scope ist eine integrierte Oszilloskop-Lösung für CANoe, basierend auf einer sehr leistungsfähigen USB-Oszilloskop-Hardware. Diese CANoe Option erscheint als weiteres Analysefenster mit Ansichten für Konfiguration, Buspegel und Protokolldekodierung. Die unterstützte Hardware hat bis zu 4 Eingangskanäle für 2 CAN/CAN FD/FlexRay oder 4 LIN/IO und wird mit Hilfe der Sync-Leitung der Vector Interface Hardware (z.B. VN1630/40, VN8900, VT System) getriggert. Die Option .Scope steht für alle CANoe Varianten außer CANoe pex zur Verfügung und ermöglicht mit der Hilfe des CANoe Test Feature Sets eine vollständige Automatisierung von Physical-Layer-Tests.

11.1 Anwendungsgebiete

Die mächtige Kombination von USB-Oszilloskop und CANoe bietet viele neue Möglichkeiten bei der Analyse von Protokollfehlern und bei der Automatisierung von Physical-Layer-Tests. Besonders bei der Durchführung von Konformitätstests ist das Darstellen der physikalischen Ebene oft unverzichtbar. Mit busspezifischen Trigger-Bedingungen und der CANoe Zeitsynchronisierung wird die Ursache von Protokollfehlern deutlich schneller als mit jedem traditionellen Oszilloskop gefunden.

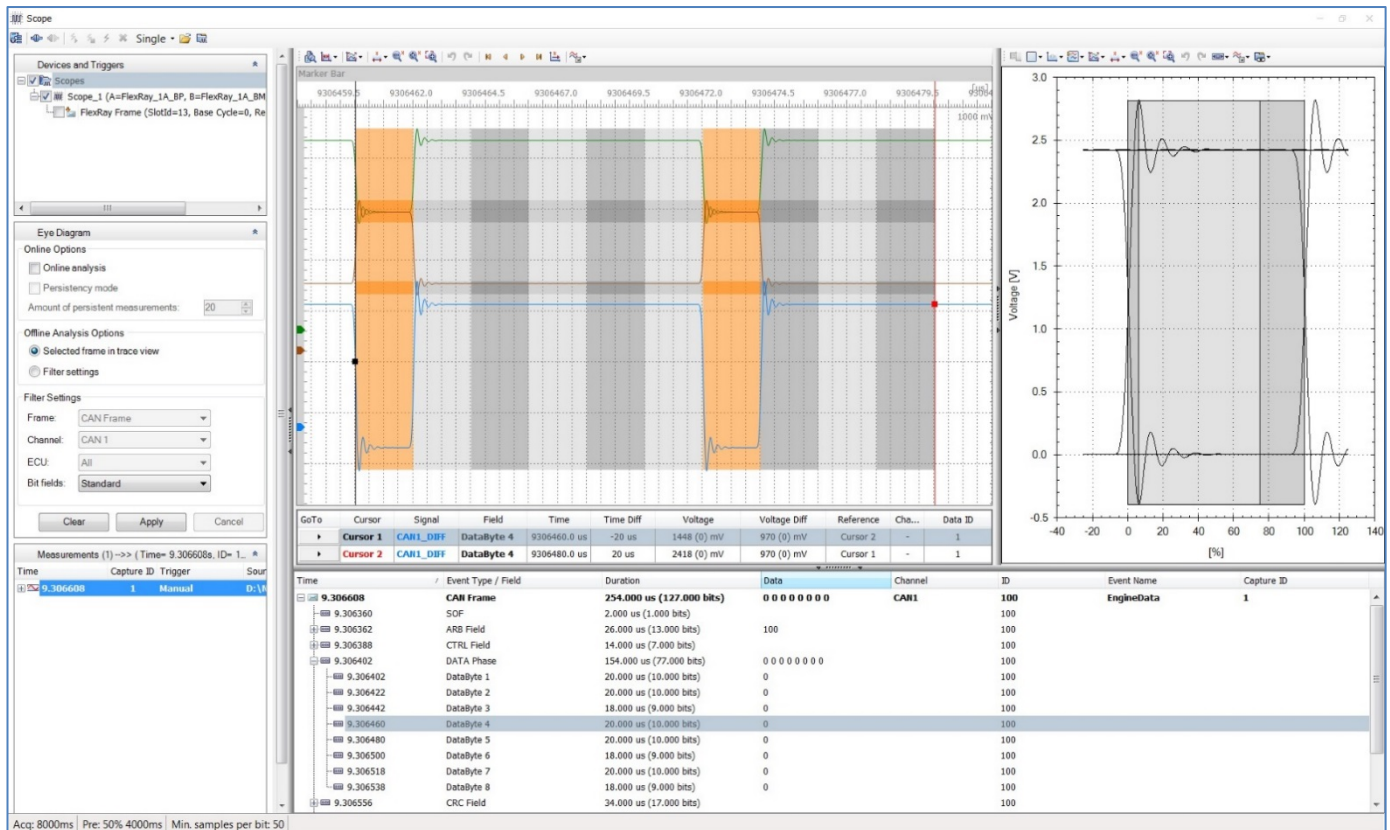


Bild 37: Detaillierte Analyse eines CAN Frames auf physikalischer und logischer Ebene inklusive Augendiagramm

11.2 Die Vorteile im Überblick

- > Extrem kompakte und portable Oszilloskop-Lösung basierend auf einer USB-Oszilloskop-Hardware
- > Bequemer Busanschluss über Vector Scope Bus Probe mit 300 MHz Bandbreite
- > Oszilloskop-Funktionen speziell für die Busanalyse entwickelt, z.B.
 - > Vordefinierte Trigger-Bedingungen für Busfehler und Ereignisse
 - > Synchrones Aufzeichnen von Bus- und I/O-Signalen mit der Zeitbasis von CANoe
 - > Vollständiges Dekodieren von Protokollfehlern (z.B. CAN Error Frames)
 - > Augendiagramm
- > CAPL-Interface für automatische Scope-Tests inklusive Reporterstellung

11.3 Unterstützte Protokolle

- > Bussysteme: CAN, CAN FD, FlexRay und LIN
- > Sensorprotokolle: SENT und PSIS

11.4 Unterstützte Oszilloskop-Hardware

- > Vector PicoScope 5444B-034
 - > USB-Präzisionsoszilloskop mit 200 MHz-Bandbreite und 500 MS/s Abtastrate
 - > 4 Eingangskanäle für Bus-Signale (z.B. 2 CAN/CAN FD/FlexRay oder 4 LIN/IO)
- > Vector Picoscope 5242B-034
 - > USB-Präzisionsoszilloskop mit 60 MHz-Bandbreite und 500 MS/s Abtastrate
 - > 2 Eingangskanäle für Bus-Signale (z.B. 1 CAN/CAN FD/FlexRay oder 2 LIN/IO)
- > Busanschluss via Vector Scope Bus Probe mit D-Sub-Verbindung
- > Vector Scope Y-Trigger Cable für interne und externe Triggerung über die Sync-Line eines Vector Interfaces

11.5 Oszilloskop-Software

Die Option .Scope erscheint in CANoe als neues Analysefenster mit Ansichten für Konfiguration, Messungen, Buspegel und Protokolldekodierung.

11.5.1 Konfigurationsfunktionen

- > Unterschiedliche Busleitungen können an die Eingänge des Oszilloskops angeschlossen werden. Für jedes Bussystem wird eine Standardkonfiguration der Scope-Hardware erstellt.
- > Einfache Konfiguration der Abtastrate (unabhängig von der Bus-Baudrate) in min. Abtastpunkte pro Bit
- > Automatisches Einstellen der Aufzeichnungsdauer abhängig von der Bus-Baudrate
- > Einstellbare Pre-/Post-Trigger-Zeit von 10 % bis 90 %
- > Unterstützt mehrere Scope-Geräte desselben Typs

11.5.2 Trigger-Funktionen

- > Trigger-Modi **Einmalig** und **Wiederholt**
- > Manuelles Triggern über die Bedienoberfläche Symbolleiste
- > Einfache Trigger-Bedingungen:
 - > Trigger auf Frame (ID oder ID-Bereich)
 - > Trigger auf Protokollfehler (z.B. CAN Error Frames)
 - > Alle Trigger-Bedingungen sind als logisches OR kombinierbar
 - > Triggern auf externe Signale mit Flanken- und Pulstrigger (I/O-Trigger)
- > Fortgeschrittene Trigger-Bedingungen in programmierbar in CAPL:
 - > Trigger auf Änderung eines Frame-Signals oder einer Systemvariablen (z.B. I/O)
 - > Trigger auf Test-Fail-Verdikt in Testmodulen
 - > Komplexe Trigger-Bedingungen über AND-/OR-Verknüpfungen definierbar
 - > Definition der Trigger-Bedingungen über die COM-Schnittstelle von CAPL

11.5.3 Analysefunktionen

- > Detaillierte Protokoll-Dekodierung der Buspegel einer Botschaft auf Bit-Ebene, auch für Protokollfehler
- > Anzeige der Zeitstempel und Spannungswerte an den Abtastpunkten
- > Bidirektionale Synchronisation zwischen der Trace-Ansicht (logische Werte des Data-Link-Layers) und dem Diagramm (physikalische Werte). Das Diagramm zeigt die Signal-Codierung der Trace-Ansicht.
- > Zeitsynchronisierte Anzeige von Botschaften mit anderen CANoe Fenstern, z.B. Trace-Fenster, Grafik-Fenster und State Tracker
- > CAPL:
 - > Serielle Bitmaskenanalyse mit definierten Bitmasken für jedes Bit im Analysebereich (CAN, CAN FD und FlexRay)
 - > Messfunktionen für Buspegelspannungen und Zeitdifferenzen (z.B. Bitzeit)
 - > Messung der Flankensteilheit auf Bitebenen
 - > Test Feature Set (TFS) zur Generierung automatisierter und reproduzierbarer Testfälle (Testreports)

11.5.4 Offline-Funktionen

Die Offline-Funktionen des Scope-Fensters stehen Anwendern auch ohne die Lizenz für Option .Scope zur Verfügung, z.B. um Messungen von Kollegen anzuschauen und zu analysieren.

- > Übersicht und einfache Verwaltung von Scope-Messungen
- > Automatische Anzeige von neuen Messdaten
- > Augendiagrammanalyse mit einer Bitmaske über die der Bedienoberfläche
- > Export und Import der gesamten Scope-Messung mit logischen und physikalischen Daten (binäres Format *.csfx)
- > Export von Scope-Messdaten in ASCII (*.csv) oder in MATLAB-Formate (*.mat)
- > Export und Import von Augendiagrammen (*.ceye)
- > Vergleichsmodus unterschiedlicher Scope-Messungen
- > Einfache Screenshot-Erstellung und Bitmap-Export für Testreports
- > Markieren markanter Messpunkte mit globalen Markern

12 Option .Sensor

Die Option .Sensor ermöglicht Ihnen die Sensorkommunikation zu analysieren. Dabei ist es möglich Sensorsignale sowohl auf dem Sensorbus als auch die Verteilung des Sensorsignals in das Fahrzeugnetzwerk zu beobachten. Selbst komplexe Kommunikationsszenarien sind schnell erzeugt und analysiert, da bewährte CANoe Analysekonzepte und eine intuitive Konfiguration zum Einsatz kommen. Mit der Möglichkeit, sowohl das Steuergerät als auch den Sensor zu simulieren, unterstützt die Option .Sensor zudem Entwickler beim Aufbau einfacher bis anspruchsvoller Testumgebungen. Bei der Simulation besteht damit die volle Kontrolle über alle protokollrelevanten Daten. Zusätzlich erleichtern ausgefeilte Fehlererkennungsmechanismen das Debuggen des Systems.

Die physikalische Anbindung an die Sensornetzwerke erfolgt mit dem Hardwaremodul VT2710. Es ist vollständig auf die Funktionalität der Option .Sensor angepasst und ist Teil der modularen Testumgebung VT System von Vector. Vorteilhaft ist der flexible Aufbau des VT2710: Je nach Bedarf sind bis zu vier PSI5- oder SENT-Kanäle über Piggy-Platinen konfigurierbar. Anwender verfügen mit diesem Modul über ein präzises Analysewerkzeug, das genaue Bitraten-Einstellungen und hochgenaue Botschaftszeitstempel erlaubt und sich sowohl im Bedienkonzept wie auch in der Programmierschnittstelle nahtlos in die bestehende VT System Umgebung einfügt. Das Modul ist für weitere Sensorprotokolle vorbereitet.

12.1 Anwendungsgebiete

Entwicklung und Einsatz von Sensoren für:

- > **Powertrain**
Drucksensor, Luftmassensensor, Lambdasonde, ...
- > **Safety**
Beschleunigungssensor, Drehratensensor, Neigungssensor, ...
- > **Komfort**
Regensensor, Temperatursensor, Luftgütesensor ...

12.2 Unterstützte Sensor-Protokolle

- > PSi5 (Peripheral Sensor Interface)
- > SENT (Single Edge Nibble Transmission)

Time	Name	Bus	Channel	Data Source	Type	Dir	Data
0.000354	SyncPulse	PSi5	Channel1	ECU	SyncPulse	Rx	
0.000146	SRSU_L::SensorFrame	PSi5	Channel1	Sensor1::SRSU_L	SensorFrame	Rx	
0.000000	Signal1	PSi5	Channel1	Sensor1::SRSU_L::Signals::Signal1	Signal	Rx	0.0000
0.000354	SyncPulse	PSi5	Channel1	ECU	SyncPulse	Rx	
0.000146	SRSU_L::SensorFrame	PSi5	Channel1	Sensor1::SRSU_L	SensorFrame	Rx	
0.000000	Signal1	PSi5	Channel1	Sensor1::SRSU_L::Signals::Signal1	Signal	Rx	0.0000
0.000354	SyncPulse	PSi5	Channel1	ECU	SyncPulse	Rx	
0.000146	SRSU_L::SensorFrame	PSi5	Channel1	Sensor1::SRSU_L	SensorFrame	Rx	
0.000000	Signal1	PSi5	Channel1	Sensor1::SRSU_L::Signals::Signal1	Signal	Rx	0.0000
0.000353	SyncPulse	PSi5	Channel1	ECU	SyncPulse	Rx	
0.000147	SRSU_L::SensorFrame	PSi5	Channel1	Sensor1::SRSU_L	SensorFrame	Rx	
0.000000	Signal1	PSi5	Channel1	Sensor1::SRSU_L::Signals::Signal1	Signal	Rx	0.0000
0.000353	SyncPulse	PSi5	Channel1	ECU	SyncPulse	Rx	
0.000147	SRSU_L::SensorFrame	PSi5	Channel1	Sensor1::SRSU_L	SensorFrame	Rx	
0.000000	Signal1	PSi5	Channel1	Sensor1::SRSU_L::Signals::Signal1	Signal	Rx	0.0000

Bild 38: PSi5 Trace

12.3 Highlights

- > Intuitive Benutzeroberfläche zum schnellen Konfigurieren der Sensorkanäle
- > Sensorkonfigurationen lassen sich komfortabel in andere CANoe Konfigurationen exportieren
- > CANoe Trace-Fenster für übersichtliche Protokollanalyse
- > Serielles Hardwaremodul VT2710 unterstützt vier PSi5- oder vier SENT-Kanäle

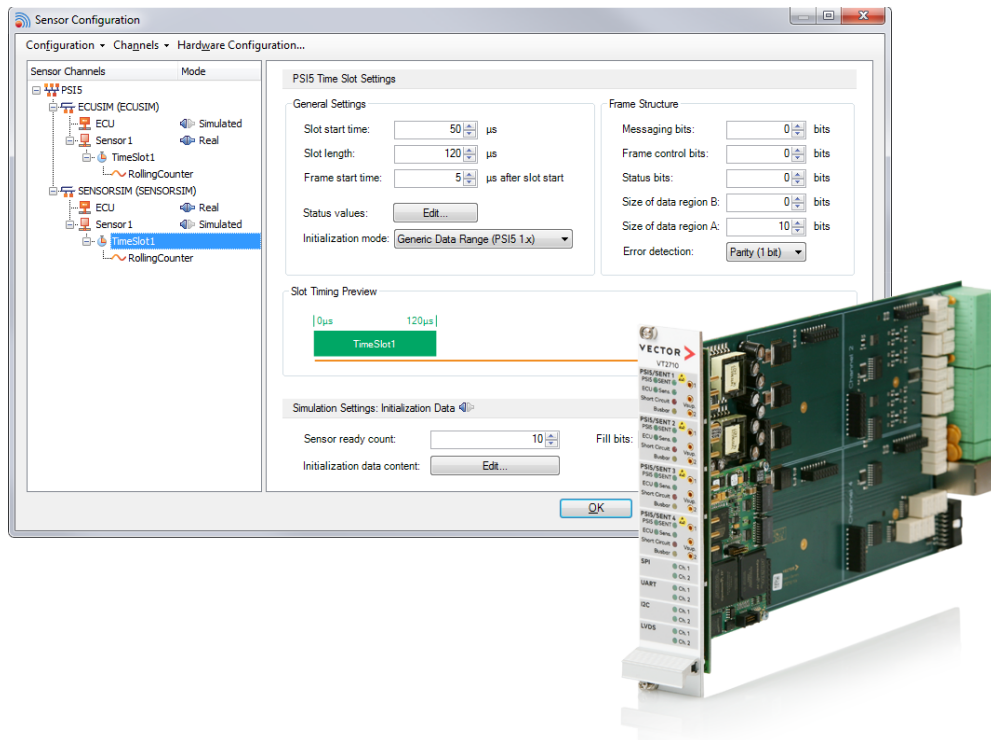


Bild 39: Sensor Modul VT2710 und Konfigurationsdialog der Option .Sensor

13 Option .XCP

Die Option .XCP erweitert CANoe um die Fähigkeit, auf den Steuergerätespeicher zuzugreifen. Der Zugriff erfolgt über das von der ASAM standardisierte XCP- oder CCP-Protokoll und wird komfortabel mit Dateien im A2L-Format konfiguriert.

13.1 Anwendungsgebiete

CANoe bietet mit XCP/CCP den Zugriff auf steuergeräteinterne Werte für Test- und Analyseaufgaben. Im Gegensatz zum reinen Blackbox-Test, bei dem nur die externen Steuergerätesignale stimuliert und gemessen werden, lassen sich über XCP/CCP auch interne Werte verstellen und auswerten. Das Verändern dieser Parameter führt gezielt zu Fehlerzuständen, das daraus resultierende Verhalten eines Steuergeräts kann direkt getestet werden. Der Test unterschiedlicher Varianten einer Steuergerätesoftware ist ebenfalls möglich – das Umschalten erfolgt direkt über XCP. Auch können fehlende Sensorwerte simuliert werden, indem Werte via XCP/CCP in die entsprechenden Speicherstellen geschrieben werden.

Bei der Analyse ermöglicht Ihnen CANoe .XCP, interne Größen des Steuergeräts zeitsynchron zu den Bussignalen zu analysieren. Statusinformationen, wie beispielsweise der Klemmenzustand oder ein Task-Wechsel, können in die Analyse einbezogen werden.

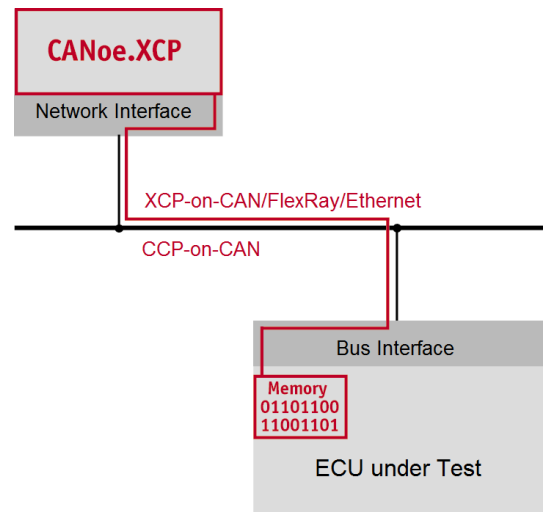


Bild 40: Zugriff auf ein Steuergerät via XCP on CAN/FlexRay/Ethernet sowie CCP-on-CAN

13.2 Steuergerätezugriff

13.2.1 Unterstützte Bussysteme und Protokolle

- > XCP (X Calibration Protocol) über CAN, CAN FD, FlexRay², Ethernet³ und LIN
- > CCP (CAN Calibration Protocol)

13.2.2 VX1000 Mess- und Kalibrier-Interface

- > Messen mit bis zu 30 MB/sec
- > Unterstützte Debug-Schnittstellen: DAP (Device Access Port), Nexus Klasse 2 und 3
- > Keine zusätzliche CPU-Belastung während des Messens über Nexus Klasse 3 und DAP
- > Fahrzeugtauglich
- > Anbindung über XCP on Ethernet
- > Zeitstempelgenauigkeit: 1 µs
- > Verfügbar für folgende Mikrocontroller: Infineon TriCore, Infineon XC 2000, Infineon Aurix, Freescale MPC5xxx, Texas Instruments TMS 570, Renesas V850 E2, weitere auf Anfrage.



Bild 41: Vector VX1000 Produktfamilie

13.2.3 Hardware-Debugger-Support

Über Hardware-Debugger der Firma iSYSTEM ist der Zugriff auf Steuergerätespeicher ohne zusätzliche Software bzw. ohne XCP-Treiber möglich. Es werden keine zusätzlichen Ressourcen beansprucht und das Echtzeitverhalten nicht beeinflusst.

- > Anbindung über XCP on Ethernet Slave in der winIDEA Software 9.12.78 oder neuer
- > Messung über vordefinierte DAQ-Listen (Data AcQuisition)
 - > Bei Änderung eines Speicherbereichs (Mindestabstand 100 µs)
 - > Zyklisch mit 1 ms, 10 ms, 100 ms, 1 s
- > Zeitstempelgenauigkeit: 100 µs
- > Universelle Lösung für die Steuergeräteentwicklung im Labor

13.3 Die Vorteile im Überblick

- > Erweiterte Testmöglichkeiten durch Zugriff auf steuergeräteinterne Werte
- > Zeitsynchrones Analysieren von steuergeräteinternen Werten, Bussignalen und I/O-Signalen
- > XCP/CCP-Protokollinterpretation im Trace-Fenster (Online und Offline)
- > Automatische Konfiguration über ASAM-A2L-Datei

² Erfordert die entsprechende CANoe Option (.FlexRay/.LIN)

³ Die Anzeige der Ethernet Pakete erfordert die CANoe Option .Ethernet. Die übertragenen Daten können auch ohne die Option .Ethernet analysiert werden.

13.4 Funktionen

- > Schreiben und Lesen von skalaren Werten⁴, eindimensionalen Arrays sowie Parametern des Typs Map und Curve
- > Lesen via Polling und DAQ-Listen (Data Acquisition)
- > Dynamische und statische DAQ-Listen
- > Paralleles Testen und Analysieren mehrerer Steuergeräte

13.4.1 Spezielle Funktionen

- > Sicherheitszugang über Seed & Key
- > XCP-Signale stehen als Systemvariablen in allen CANoe Funktionen zur Verfügung
- > Auswertung der per DAQ übertragenen Steuergeräte-Zeitstempel
- > Lauffähig unter CANoe RT (Real Time) und VN8900 im Standalone-Betrieb

13.5 Integration in CANoe

CANoe übernimmt die Rolle des XCP-/CCP-Masters, der über eine A2L-Datei konfiguriert wird. Die XCP-Signale sind als Systemvariablen verfügbar, was den Zugriff via Test Feature Set, CAPL, .NET, DiVa, MATLAB/Simulink und den CANoe Analysefenstern ermöglicht. Die XCP-Signale befinden sich immer unterhalb des Namensraums <XCP>::<Steuergerätename>. „Steuergerätename“ steht hier für den Namen des zu messenden Steuergeräts, der bei der XCP-Konfiguration angegeben wird. Dadurch kann CANoe .XCP parallel mehrere Steuergeräte über XCP ansprechen.

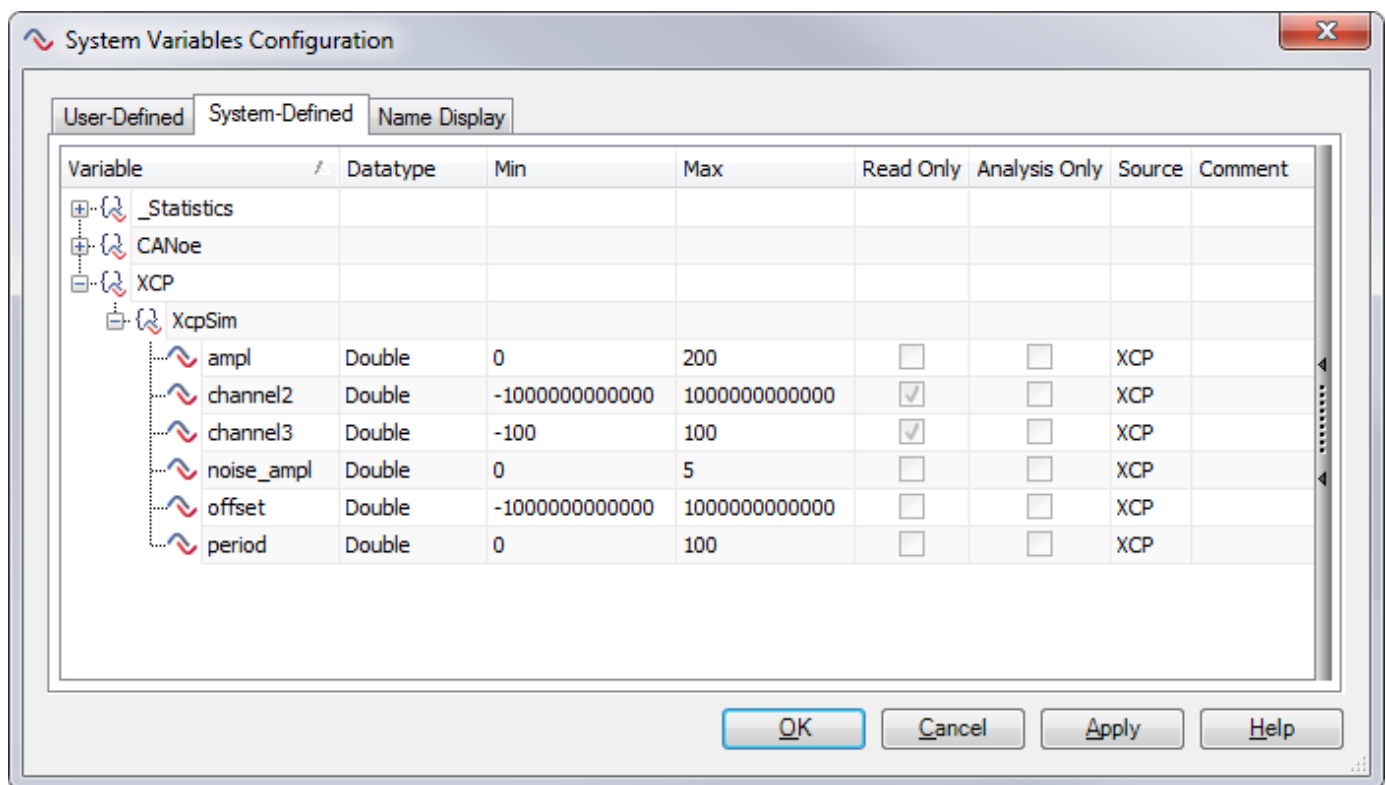


Bild 42: Systemvariablenkonfiguration in CANoe

⁴ 64-Bit-Variablen können nur mit einem Wertebereich von 52 Bit verwendet werden.

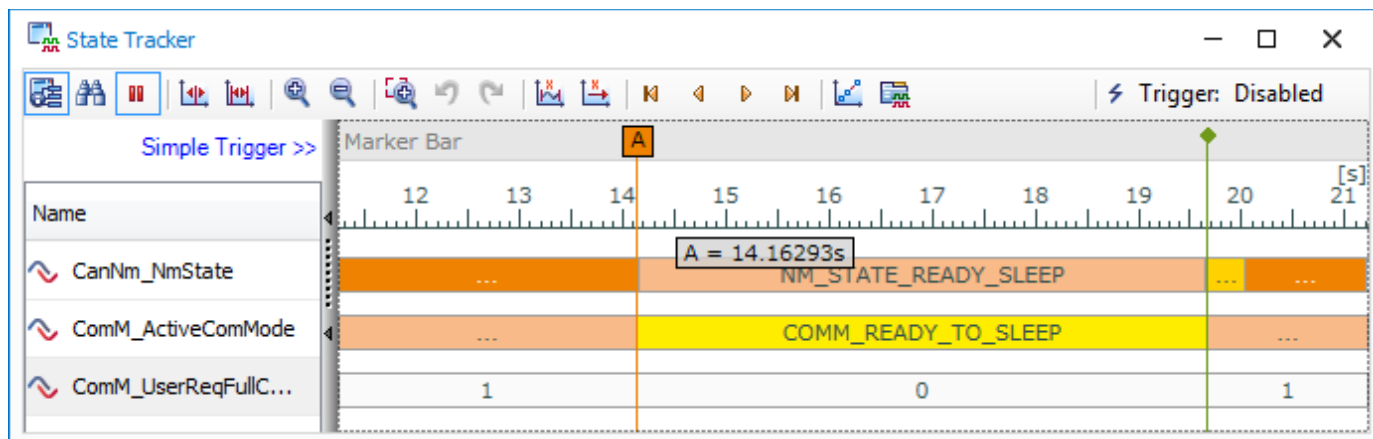


Bild 43: XCP-Systemvariablen im State Tracker

13.6 Konfiguration

Die Konfiguration der Option .XCP erfolgt durch das vom ASAM-Arbeitskreis standardisierte A2L-Dateiformat. In der A2L-Datei sind Informationen über die Kommunikation (Transportschicht) mit dem Steuergerät und die Beschreibungen der mess- und stimulierbaren Variablen enthalten. Es sind auch Informationen über die Werteinterpretation (symbolische Wertetabellen) und Umrechnungsregeln enthalten. So können zum Beispiel direkt symbolisch interpretierte Werte in den Analysefenstern von CANoe angezeigt werden.

14 Option .AMD (AUTOSAR Monitoring and Debugging)

Aktuelle und zukünftige Steuergerätegenerationen enthalten immer mehr und immer komplexere Funktionen. Es werden Funktionen auf verschiedene Steuergeräte aufgeteilt und von unterschiedlichen Zulieferern entwickelt. Durch den AUTOSAR-Standard werden Software-Schichten sowie deren Schnittstellen innerhalb des Steuergeräts spezifiziert. Die Anwendungslogik wird in Software-Komponenten (**SWC**) aufgeteilt, die über die Runtime Environment (**RTE**) miteinander kommunizieren. Die Ansteuerung der Hardware sowie allgemeine Basisdienste wie Netzwerk-Management sind in Basis-Software-Modulen (**BSW-Module**) realisiert. Ein Blick ins Innere des Steuergeräts ist somit notwendig, um zum Beispiel eine fehlerhafte Software-Komponente zu identifizieren oder Teilfunktionen des Steuergeräts zu testen. Dieser „Blick ins Steuergerät“ wird mit der CANoe Option .AMD möglich.

Die Option .AMD enthält den vollen Funktionsumfang der Option .XCP. CANoe erhält somit Zugriff auf den Steuergerätespeicher. Voraussetzung für die Verwendung von CANoe .AMD ist ein Steuergerätezugang über XCP/CCP, über die VX1000 Mess- und Kalibrier-Hardware von Vector oder über den iSYSTEM Hardware-Debugger.

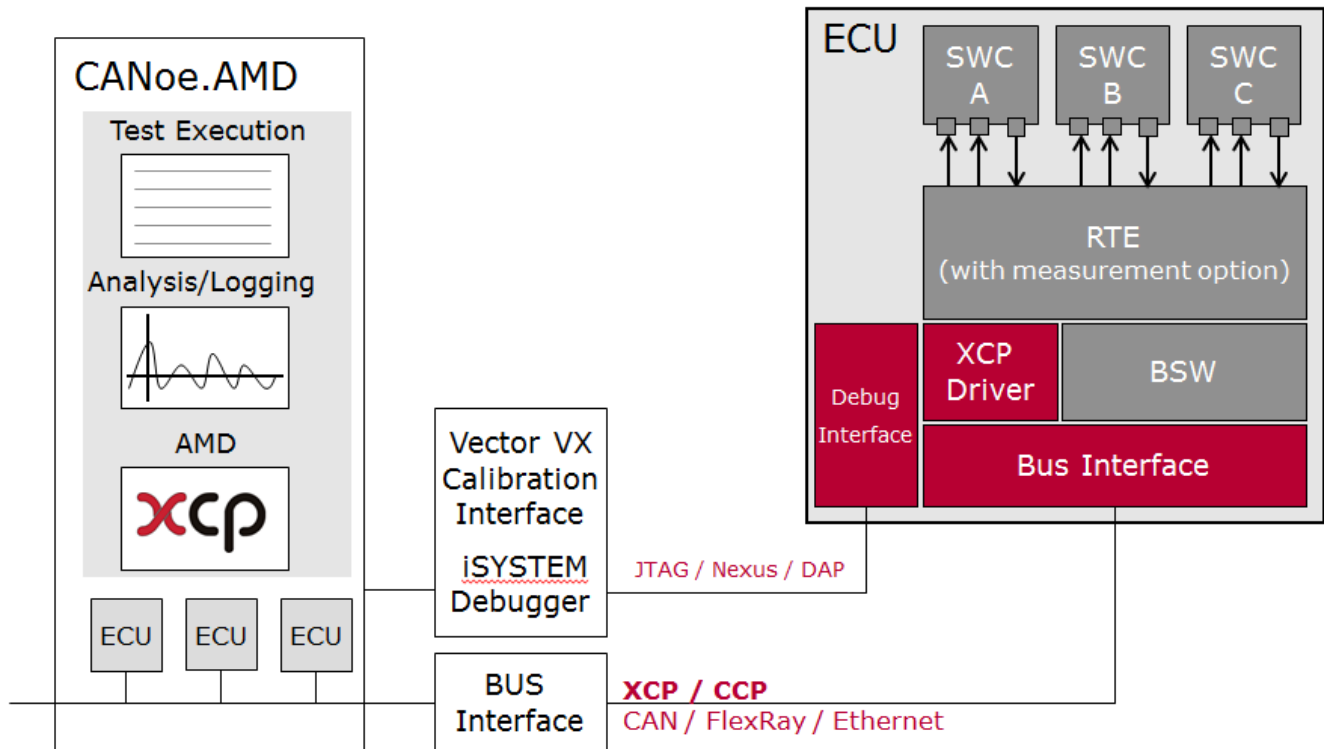


Bild 44: Mögliche Steuergerätezugänge mit CANoe .AMD

14.1 Anwendungsgebiete

- > Fehlersuche in Netzwerken und verteilten Systemen, z.B. das Einschlaf- und Aufwachverhalten des Netzwerk-Managements
- > Fehlersuche in der Basis-Software und in den Software-Komponenten
- > Analyse von verteilten Funktionen mit Zugriff auf mehrere Steuergeräte parallel
- > Automatisiertes Testen von Funktionen mit Zugriff auf den Steuergerätespeicher
- > Prüfung der Integration der Basis-Software und RTE durch Auslesen der entsprechenden Parameter

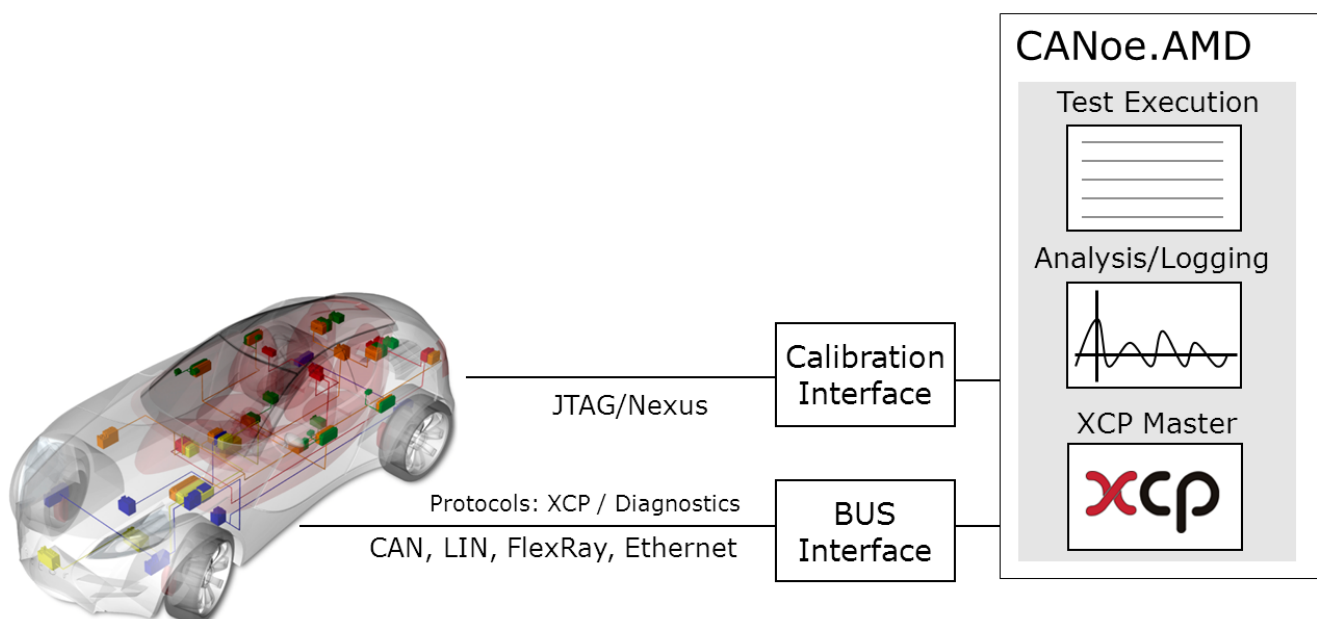


Bild 45: CANoe .AMD, z.B. Einsatz im Fahrzeug

14.1.1 Konfigurationsprozess für konventionelles Messen

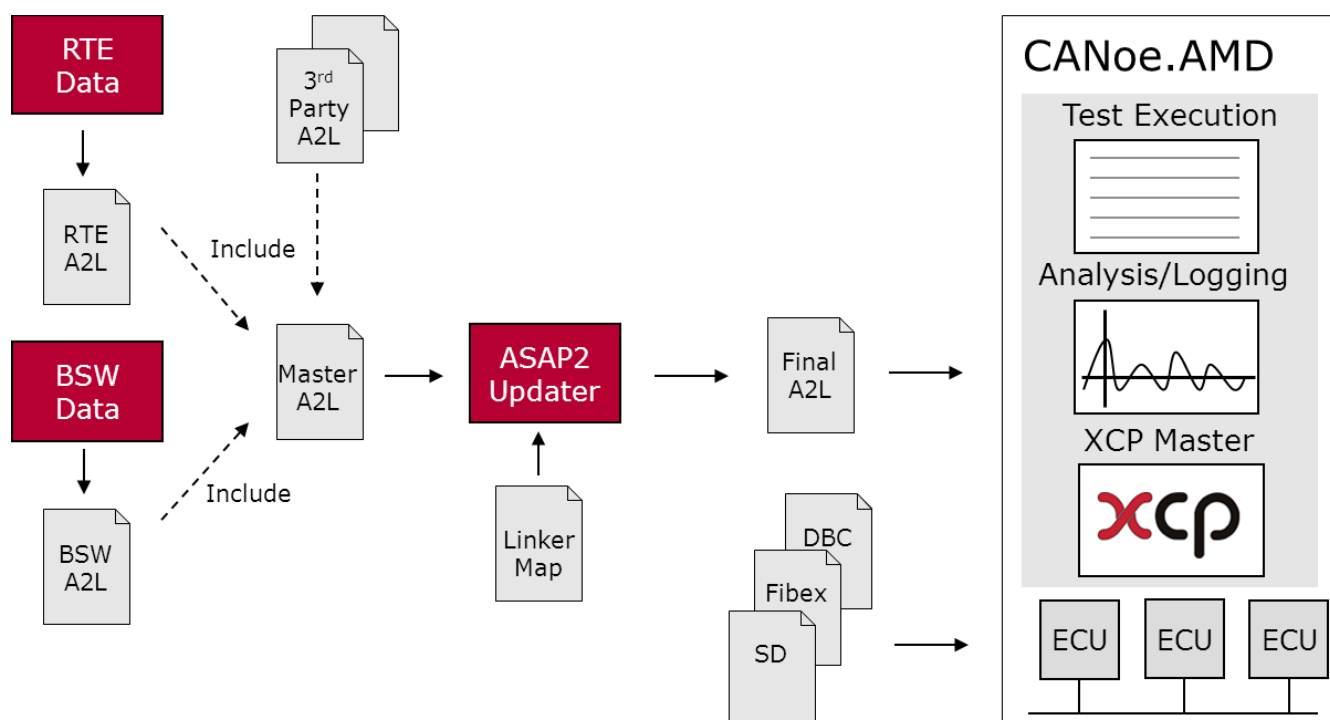


Bild 46: Konfigurationsprozess

Während der Entwicklung eines Steuergeräts mit der Vector AUTOSAR-Lösung (MICROSAR) entstehen verschiedene A2L-Fragmente. Der RTE Generator erzeugt auf Wunsch eine A2L-Datei zum Zugriff auf die Kommunikationsports, die den Software-Komponenten zur Verfügung stehen. Gleichzeitig wird der RTE Code instrumentiert, so dass die Daten synchron zur Anwendung in den XCP-Treiber kopiert werden.

Auch für die Basis-Software-Module der MICROSAR Lösung kann automatisch eine A2L-Datei generiert werden, welche den Zugriff auf die wichtigsten Variablen in der Basis-Software ermöglicht.

Alle A2L-Fragmente werden in einer Master-A2L-Datei inkludiert und somit in einer Datei zusammengefasst. Diese muss nach einem Compiler-Lauf mit den aktuellen Adressen aus der Linker-MAP-Datei aktualisiert werden, um sie in CANoe .AMD verwenden zu können. Die Option .AMD enthält hierfür den ASAP2 Updater.

14.1.2 ASAP2 Updater

Die Aufgabe des ASAP2 Updater ist es, die realen Steuergeräteadressen aus der Linker-MAP-Datei den Variablen in der A2L-Datei zuzuordnen und so eine komplette A2L-Datei zu erstellen. Der ASAP2 Updater kann über die Kommandozeile gestartet und parametrisiert werden oder über ein GUI aus dem Tools-Ordner im Start- Menü von CANoe .AMD aufgerufen werden.

14.2 MICROSAR Monitoring Feature

Beim MICROSAR Monitoring Feature handelt es sich um eine Erweiterung der Vector AUTOSAR-Lösung. Diese Erweiterung ermöglicht es die Basis-Software-Module zu messen. Ein Generator erzeugt basierend auf der Konfiguration der Basis-Software-Module eine A2L-Datei.

14.2.1 Generisches Messen

Das generische Messen ist Teil des MICROSAR Monitoring Features. Beim generischen Messen werden die realen Steuergeräteadressen für ausgewählte Variablen beim Aufbau der XCP-Verbindung vom Steuergerät an CANoe übertragen. Ein Adressen-Update der A2L-Datei vor der Messung ist nicht notwendig. Mit einer einzigen A2L-Datei können Steuergeräte mit unterschiedlichen Software-Ständen gemessen werden. Eine generische A2L-Datei enthält für alle Variablen, die über das generische Messen ausgelesen werden, eine spezielle Kennzeichnung, die von CANoe .AMD ausgelesen wird. So kann das generische Messen mit konventionell gemessenen Variablen in der A2L-Datei kombiniert werden.

Anmerkung:

Der XCP-Slave-Treiber im Steuergerät muss generisches Messen unterstützen. Der Vector MICROSAR⁵-XCP-Professional-Treiber unterstützt das generische Messen.

15 Funktionserweiterung für spezielle Anwendungen

15.1 DiVa (Diagnostic Integration and Validation Assistant)

Die Option .DiVa erweitert CANoe zu einem Werkzeug zum automatischen Generieren und Ausführen von Testfällen für das Implementieren und Integrieren des Diagnoseprotokolls. Die Testfälle werden auf Basis von ODX- oder CANdela Diagnosebeschreibungen erzeugt und gewährleisten eine breite und detaillierte Testabdeckung für die Diagnose-Implementierung eines Steuergerätes.

16 Schulungen

Im Rahmen unseres Schulungsangebotes bieten wir für CANoe verschiedene Schulungen und Workshops in unseren Seminarräumen bei Vector sowie vor Ort bei unseren Kunden an.

Mehr Informationen zu den einzelnen Schulungen und die Termine finden Sie im [Internet](#).

⁵ MICROSAR ist die AUTOSAR-Basis-Software- und RTE-Implementierung von Vector



Mehr Informationen

Besuchen Sie unsere Website für:

- > News
- > Produkte
- > Demo-Software
- > Support
- > Seminare und Workshops
- > Kontaktadressen

www.vector.com