

Vortrag auf der HdT-Konferenz

**vom 13./14.05.2009
in Dresden**

**Einführung in die standardisierte Diagnose-
kommunikation mit UDS on CAN, MVCI und ODX**

**Autor:
Peter Subke – Softing AG**

Abstract

A large number of individual, incompatible solutions for measurement, calibration, and diagnostic tasks make it difficult to exchange data, interconnect components, and operate different systems together. Many vehicle manufacturers and suppliers realized that the best way to overcome this situation would be to standardize interfaces. The Association for Standardisation of Measurement and Calibration Systems (ASAM) provides such standardized interfaces.

The ASAM MCD System and its interfaces ASAM MCD-2D and ASAM MCD-3D are now internationally standardized as ISO 22900 (Modular Vehicle Communication Interface, MVCI) and the respective database format is standardized in ISO 22901 (Open Diagnostic Data Exchange, ODX).

Besides a specification of hardware design requirements in ISO 22900 part one, the MVCI standard specifies two software interfaces: the D-PDU API and the D-Server API.

The standardized technology for test equipment accompanies standardized diagnostic protocols, such as UDS (Unified Diagnostic Services). UDS is standardized in ISO 14229-1, and the implementation on CAN is specified in ISO 15765-3. The combination of ISO 14229-1 and ISO 15765 are complementary forming the diagnostic protocol „UDS on CAN“.

This presentation provides a technical overview about the technology of a diagnostic tester that is based on the ASAM MCD system (MVCI D-Server). The diagnostic protocol is UDS on CAN and is described in ODX. The respective ISO standards are ISO 11898 (CAN), 14229-1 und ISO 15765 (UDS on CAN), ISO 22900 (MVCI) and ISO 22901 (ODX).

Kurzfassung

Proprietäre und inkompatible Mess-, Kalibrier- und Diagnosewerkzeuge erschweren den Austausch von Daten und die Kombination von Komponenten und Systemen.

Fahrzeughersteller und Zulieferer sehen die Lösung dieses Problems in der Standardisierung von Systemen und Schnittstellen (Interfaces).

Der ASAM e.V. (Association for Standardisation of Measurement and Calibration Systems) entwickelt solche standardisierten Interfaces.

Das ASAM MCD System mit seinen standardisierten Interfaces ASAM MCD-2D and ASAM MCD-3D wurden jetzt als internationale Standards veröffentlicht. ISO 22900 spezifiziert das Modular Vehicle Communication Interface (MVCI) mit den zwei Softwareinterfaces D-PDU API und D-Server API.

Für die Diagnosekommunikation zwischen einem PC-basierten Tester und einem oder mehreren Steuergeräten wird heute zunehmend das in ISO 14229-1 international standardisierte Diagnoseprotokoll UDS (Unified Diagnostic Services) verwendet.

Zusammen mit ISO 15765-3 spezifiziert ISO 14229-1 das Diagnoseprotokoll *UDS on CAN*.

Dieser Beitrag liefert einen technischen Überblick über die Technologie PC-basierter Diagnosetester mit einem MVCI D-Server und dem Protokoll UDS on CAN, das im ODX-Format beschrieben wird. Die relevanten Standards sind ISO11898 (CAN), 14229-1 und ISO 15765 (UDS on CAN), ISO 22900 (MVCI) und ISO 22901 (ODX).

1 Einleitung

Ohne den Einsatz mikrocontrollerbasierter Steuergeräte im Fahrzeug sind viele Fahrzeugfunktionen entweder gar nicht oder nur mit erheblichem Aufwand an Material und Kosten umsetzbar. Zur Realisierung der gewünschten Funktion verarbeiten die Steuergeräte Eingangsinformationen (Input Info) von Sensoren zu Ausgangsinformationen (Output Info) zur Ansteuerung von Aktoren.

Typische Beispiele für Eingangsinformationen sind die Umgebungstemperatur, die aktuelle Fahrzeuggeschwindigkeit und die Motordrehzahl, aber auch der Befehl des Fahrers, das Fernlicht einzuschalten. Typische Beispiele für Ausgangsinformationen sind Signale zur Ansteuerung der ABS-Ventile, Informationen über den Zündzeitpunkt oder der Befehl zum Verriegeln der Türen.

Viele Informationen werden zur Realisierung mehrerer Funktionen benötigt. Beispielsweise wird der Wert der Umgebungstemperatur sowohl vom Steuergerät für die Klimaanlage, als auch vom Motorsteuergerät benötigt.

Heute werden innerhalb eines Fahrzeugs jede Sekunde viele tausend Informationen ermittelt, geprüft, aktualisiert und weiterverarbeitet. Um systemweite Datenkonsistenz sicherzustellen, ist es sinnvoll, jede Information nur einmal zu ermitteln und dann allen Steuergeräten, die diese Informationen benötigen, zur Verfügung zu stellen. Um das zu erreichen, werden die Steuergeräte physikalisch und logisch miteinander vernetzt. Dabei ist die verwendete Vernetzungstechnologie (z.B. LIN, CAN, MOST, FlexRay) abhängig vom Anwendungsfall.

Vernetzte Steuergeräte können Eingangsinformationen von anderen Steuergeräten verarbeiten und anderen Steuergeräten Informationen in Form von Daten zur Verfügung zu stellen. Dieser Austausch von Informationen wird als *in-vehicle communication* (auch: onboard communication oder normal communication) bezeichnet (Abbildung 1).

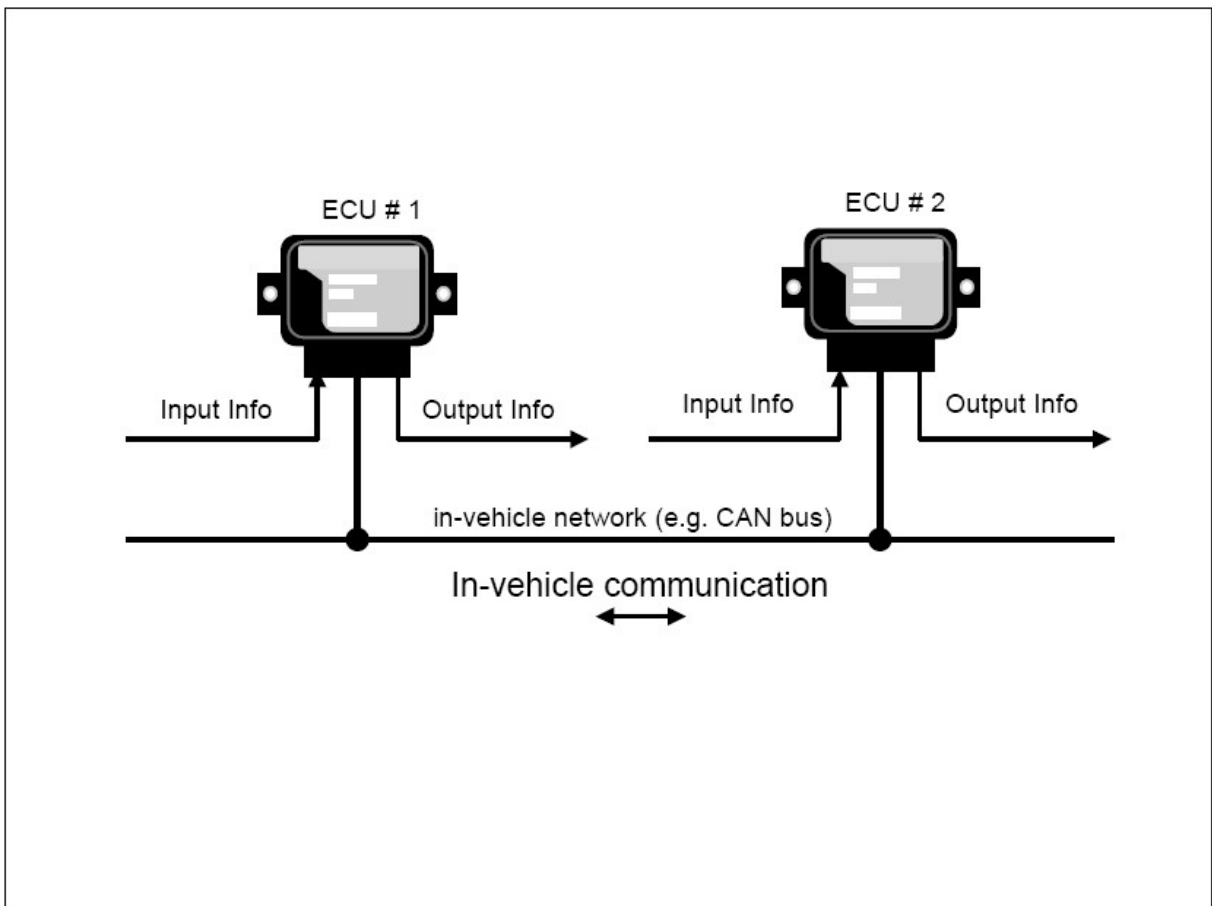


Abbildung 1: In-vehicle communication

Wird ein externer Tester (external test equipment) mit dem in-vehicle network verbunden, so kann der Tester die in-vehicle communication mithören (monitoring) und aufzeichnen (tracing) oder seinerseits Informationen senden, z.B. um das Kommunikationsverhalten nicht vorhandener Steuergeräte zu simulieren. Diese Funktion eines Testers wird auch als Restbussimulation bezeichnet.

Darüber hinaus kann ein externer Tester z.B. Fehlerspeicher lesen und löschen, neue Datensätze oder Software per Flash-Download programmieren oder spezielle Steuergeräte-Routinen starten, z.B. zum Anlernen des Einklemmschutzes beim Fensterheber oder zum Befüllen der Bremsanlage in der Fahrzeugproduktion.

Befehle, die der Tester als Master (Client) an die Steuergeräte senden kann und die von den Steuergeräten als Slaves (Server) ausgeführt werden, sind in einem *Diagnoseprotokoll* festgelegt. Kommt ein Diagnoseprotokoll zum Einsatz, wird dies als Diagnosekommunikation (diagnostic communication) bezeichnet (Abbildung 2).

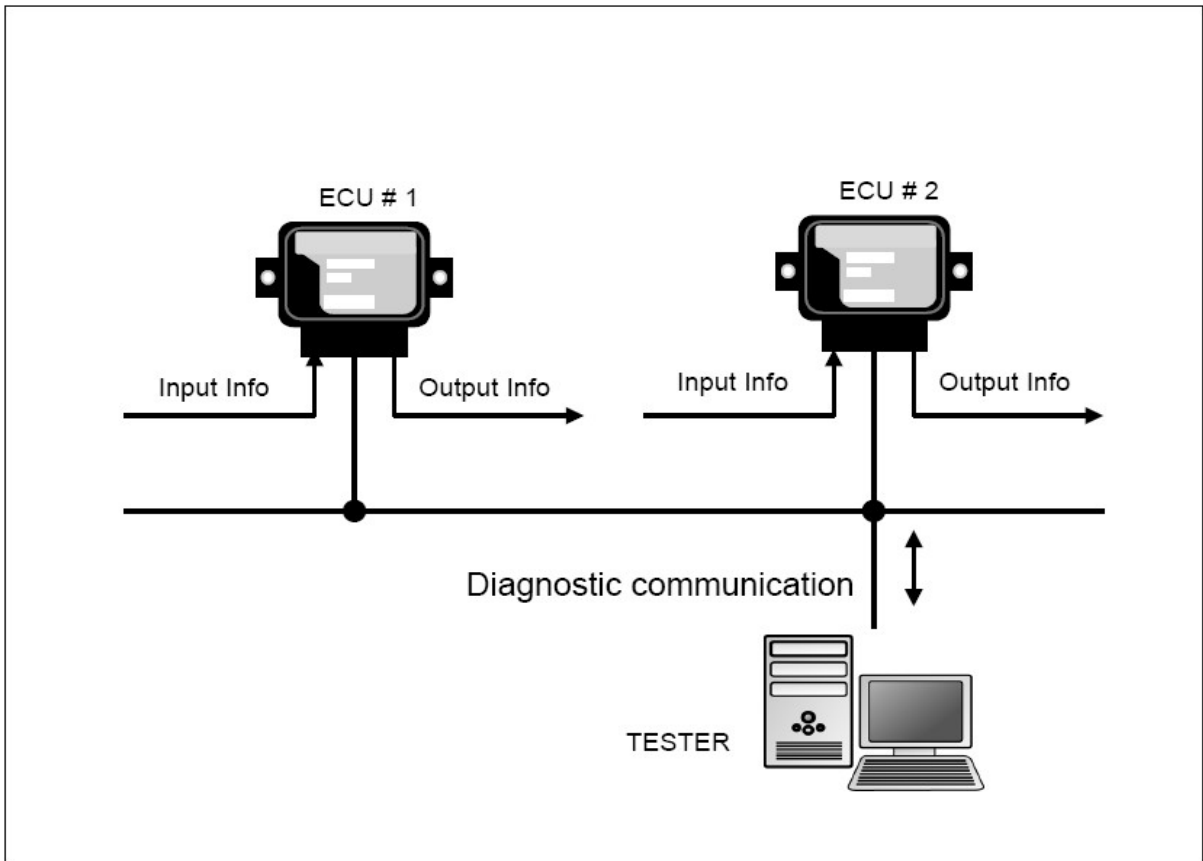


Abbildung 2: Diagnostic communication

Zusammenfassung: Steuergeräte im Fahrzeug verarbeiten Eingangsinformationen von Sensoren und anderen Steuergeräten zu Ausgangsinformationen für Aktoren und andere Steuergeräte. Neben der Fähigkeit zur in-vehicle communication unterstützen die Steuergeräte ein Diagnoseprotokoll zur Diagnosekommunikation mit einem externen Tester.

2 Controller Area Network (ISO 11898)

Das Anfang der 90er Jahre vorgestellte Controller Area Network (CAN) dient der Vernetzung von Steuergeräten zur in-vehicle communication. CAN ist spezifiziert in ISO 11898: Road vehicles - Controller area network (CAN). ISO 11898 besteht aus vier Teilen:

Part 1	2003	Data link layer and physical signalling
Part 2	2003	High-speed medium access unit
Part 3	2006	Low-speed, fault-tolerant, medium-dependent interface
Part 4	2004	Time-triggered communication

CAN besteht aus einer (verdrillten) Zweidrahtleitung mit Differenzsignal. Die Topologie ist ein linearer Bus, die maximale Datenrate 1 MBit/s bei einer Länge bis zu 40m. In der Praxis wird der CAN mit maximal 500 kbit/s betrieben.

Alle Steuergeräte am CAN sind gleichberechtigte Kommunikationspartner. Es gibt keinen Master, der die Kommunikation steuert. Die über einen CAN-Bus verbundenen Steuergeräte haben keine physikalischen Adressen.

Die Tatsache, dass alle Steuergeräte gleichberechtigt sind, bedeutet jedoch nicht, dass alle Botschaften die gleiche Priorität besitzen. Die Priorität der CAN-Botschaften wird über einen CAN-Identifizier (CAN-ID) gesteuert, der zudem Aufschluss über den Inhalt der Botschaft gibt. Es wird unterschieden zwischen 11-bit CAN-ID (CAN-A) und 29-bit CAN-ID (CAN-B). Unabhängig davon können mit jeder Botschaft maximal 8 Byte an Daten übertragen werden.

Da alle Steuergeräte gleichberechtigt sind, können Kollisionen auftreten, wenn gleichzeitig mehrere Steuergeräte senden wollen. Kollisionen werden durch ein wirkungsvolles Arbitrierungsverfahren (CSMA/CA) mit Hilfe des CAN-ID aufgelöst.

3 UDS on CAN (ISO 14229-1 & ISO 15765-3)

ISO 14229-1 (2006): Road vehicles - Unified Diagnostic Services - Part 1: Specification and requirements, beschreibt ein Diagnoseprotokoll. Es beinhaltet 25 Diagnoseservices, die der Tester an ein oder mehrere Steuergeräte senden kann. Ein Diagnoseservice besteht aus einem Request des Testers und mindestens einer Antwort (Response). Jedem Diagnoseservice ist ein eindeutiger Service-Identifizier (SID) zugeordnet. Es wird unterschieden zwischen Request-SID und Response-SID.

Hinweis: Der Response-SID ergibt sich mathematisch aus dem Request-SID durch Addition von 40hex.

Die in der ISO 14229-1 spezifizierten Services sind in Tabelle 1 gelistet.

Kann das Steuergerät den Request des Testers nicht ausführen, sendet es eine *Negative Response* (Response SID: \$7F). Eine Negative Response enthält Informationen über die Gründe für die Ablehnung, z.B. *Service Not Supported* oder *Conditions Not Correct*.

Requests und Responses können Parameter enthalten, die den jeweiligen Service genauer spezifizieren. Bei der Abfrage von Daten enthält das Datenfeld der Response in der Regel als Parameter die abgefragte Information.

ISO 14229-1 beschreibt die Diagnoseservices unabhängig vom in-vehicle network. Im vorliegenden Beispiel sind Tester und Steuergeräte über CAN miteinander verbunden. Die Umsetzung der Diagnosekommunikation mit UDS auf CAN wird beschrieben in ISO 15765: Road vehicles - Diagnostics on Controller Area Network (CAN). ISO 15765 besteht aus vier Teilen, wobei hier ISO 15765 - Part 3 (2004): Implementation of unified diagnostic services (UDS on CAN) relevant ist.

Request SID	Response SID	Service
\$10	\$50	DiagnosticSessionControl
\$11	\$51	EcuReset
\$27	\$67	SecurityAccess
\$28	\$68	CommunicationControl
\$3E	\$7E	TesterPresent
\$83	\$C3	AccessTimingParameter
\$84	\$C4	SecuredDataTransmission
\$85	\$C5	ControlDTCSetting
\$86	\$C6	ResponseOnEvent
\$87	\$C7	LinkControl
\$22	\$62	ReadDataByIdentifier
\$23	\$63	ReadMemoryByAddress
\$24	\$64	ReadScalingDataByIdentifier
\$2A	\$6A	ReadDataByPeriodicIdentifier
\$2C	\$6C	DynamicallyDefineDataIdentifier
\$2E	\$6E	WriteDataByIdentifier
\$3D	\$7D	WriteMemoryByAddress
\$14	\$54	ClearDiagnosticInformation
\$19	\$59	ReadDTCInformation
\$2F	\$6F	InputOutputControlByIdentifier
\$31	\$71	RoutineControl
\$34	\$74	RequestDownload
\$35	\$75	RequestUpload
\$36	\$76	TransferData
\$37	\$77	RequestTransferExit
	\$7F	Negative response

Tabelle 1: Die Unified Diagnostic Services und ihre Service-Identifizierer

Der Tester sendet einen Diagnose-Service-Request an ein Steuergerät. Das angesprochene Steuergerät führt den Service aus und sendet eine zum Request gehörende Response.

Da die Steuergeräte und auch der Tester am CAN keine physikalischen Adressen haben, erfolgt die Adressierung über jeweils festgelegte CAN-Identifizier. Dabei wird unterschieden zwischen physikalischen Request CAN-ID und Response CAN-ID. Darüber hinaus kann der Tester mehrere Steuergeräte gleichzeitig über eine funktionale CAN-ID ansprechen.

4 MVCI (ISO 22900) und ODX (ISO 22901)

Abbildung 3 zeigt stark vereinfacht den Aufbau eines Kommunikationssystems. Das in-vehicle network verbindet die Steuergeräte für die in-vehicle communication. In diesem Beispiel besitzt das linke Steuergerät zwei CAN-Anschlüsse für die Datenkommunikation. Es separiert als Gateway die in-vehicle communication von der Diagnosekommunikation, bei der ein Diagnoseprotokoll zum Einsatz kommt.

Anmerkung: Der zweite Anschluss wird oft als Diagnose-CAN bezeichnet. Dieser Anschluss muss nicht vorhanden sein. Es ist auch möglich, den Tester direkt mit dem in-vehicle CAN zu verbinden.

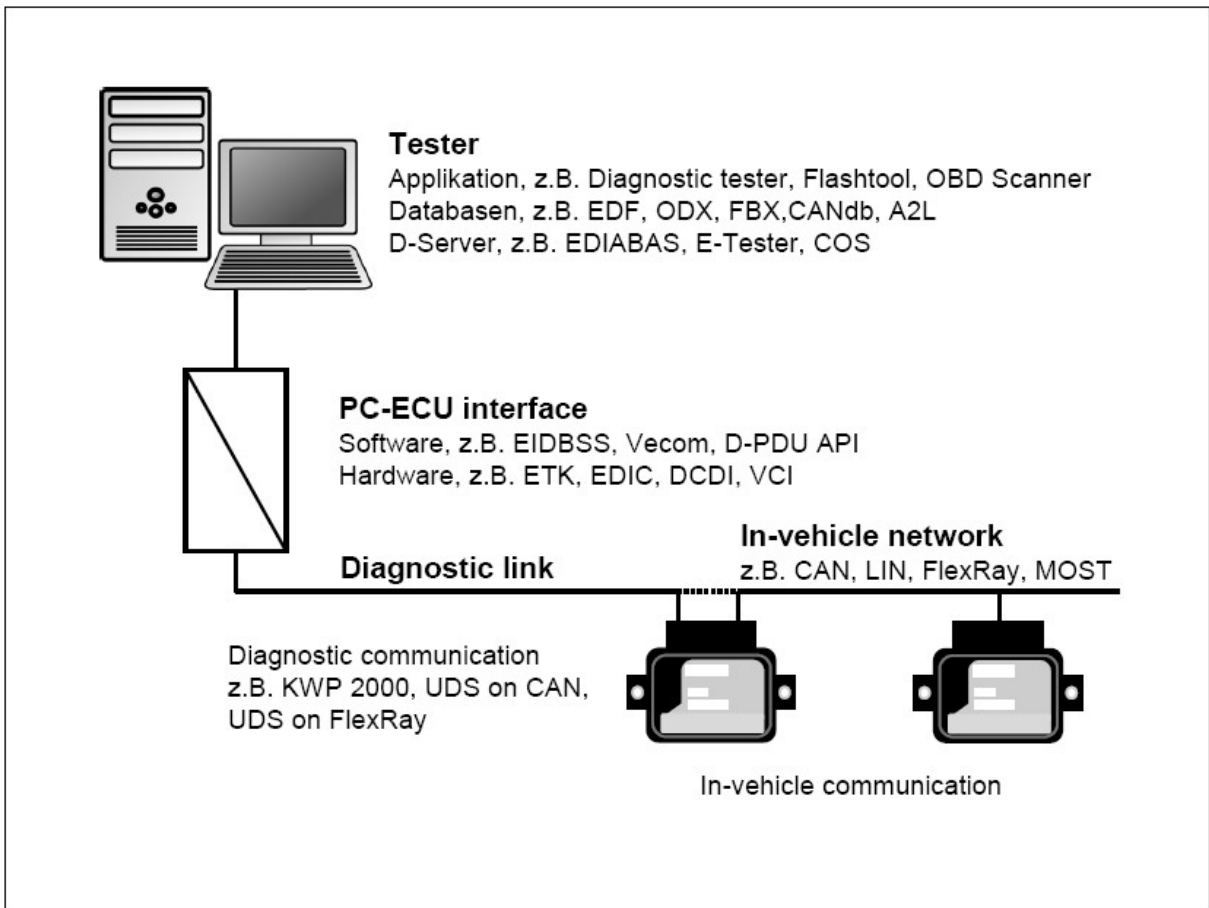


Abbildung 3: Die Komponenten eines Kommunikationssystems

Ein intelligentes PC-ECU Interface verbindet den Diagnose-CAN mit dem PCbasierten Tester, der z.B. als Diagnosetester, Flashtool oder als OBD-Scanner

verwendet werden kann. Die dazu benötigte Software besteht neben dem PC Betriebssystem (z.B. XP oder Vista) aus einem datengetriebenen Diagnose-Server (D-Server).

Abbildung 4 zeigt die Software-Module (grau) des PC-basierten Testers. Das Basissystem verbindet über Software-Interfaces die Applikation mit dem PC-ECU Interface. Die steuengerätespezifischen Eigenschaften für die Diagnosekommunikation (u.a. das Diagnoseprotokoll) sind in einer Datenbasis abgelegt, die mit einem Editor erstellt, verändert und geprüft werden kann. In der Vergangenheit haben Fahrzeughersteller und Zulieferer proprietäre Testersoftware und Datenbasisformate entwickelt. Ein typisches Beispiel dafür ist EDIABAS (Elektronik Diagnose Basissystem) mit Steuengerätebeschreibungsdateien (SGBD oder EDF = ECU Description Files) als Datenbasis.

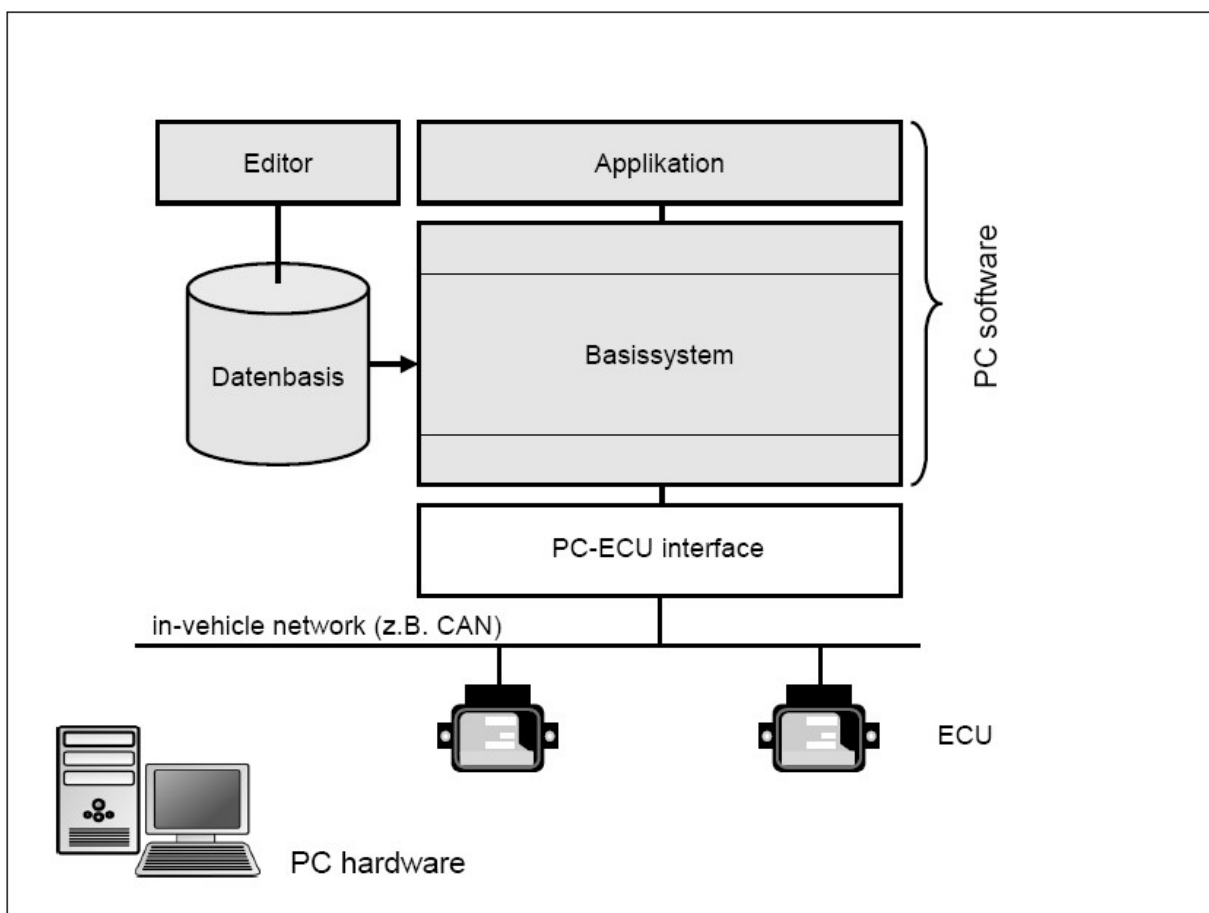


Abbildung 4: Die Komponenten eines PC-basierten Testers

Die Vorteile standardisierter Technologien gelten nicht nur für Bussysteme (hier CAN) und Diagnoseprotokolle (hier UDS on CAN), sondern auch für die Komponenten PC-basierter Tester und den Datenformaten. Seit vielen Jahren arbeitet der ASAM e.V. (Association for Standardisation of Automation and Measuring Systems) an der Standardisierung von Schnittstellen und Datenformaten.

Abbildung 5 zeigt den Diagnoseteil des ASAM-MCD Systems für die Diagnosekommunikation eines PC-basierten Testers mit einem Steuergerät. Das MCD-System besteht aus dem D-Server als Basissystem mit der ASAM MCD-3D Schnittstelle zur Applikation. Die Diagnosedaten werden im Format ASAM MCD-2D beschrieben. Dieses Diagnosedaten-Austauschformat wird auch als ODX (Open Diagnostic Data Exchange) bezeichnet.

Im Laufe der Jahre hat der ASAM e.V. mehrere Versionen der ASAM MCD-3D und ASAM MCD-2D Spezifikationen veröffentlicht. Abbildung 6 zeigt diese Versionen in den Kombinationen, die in der Praxis eingesetzt werden.

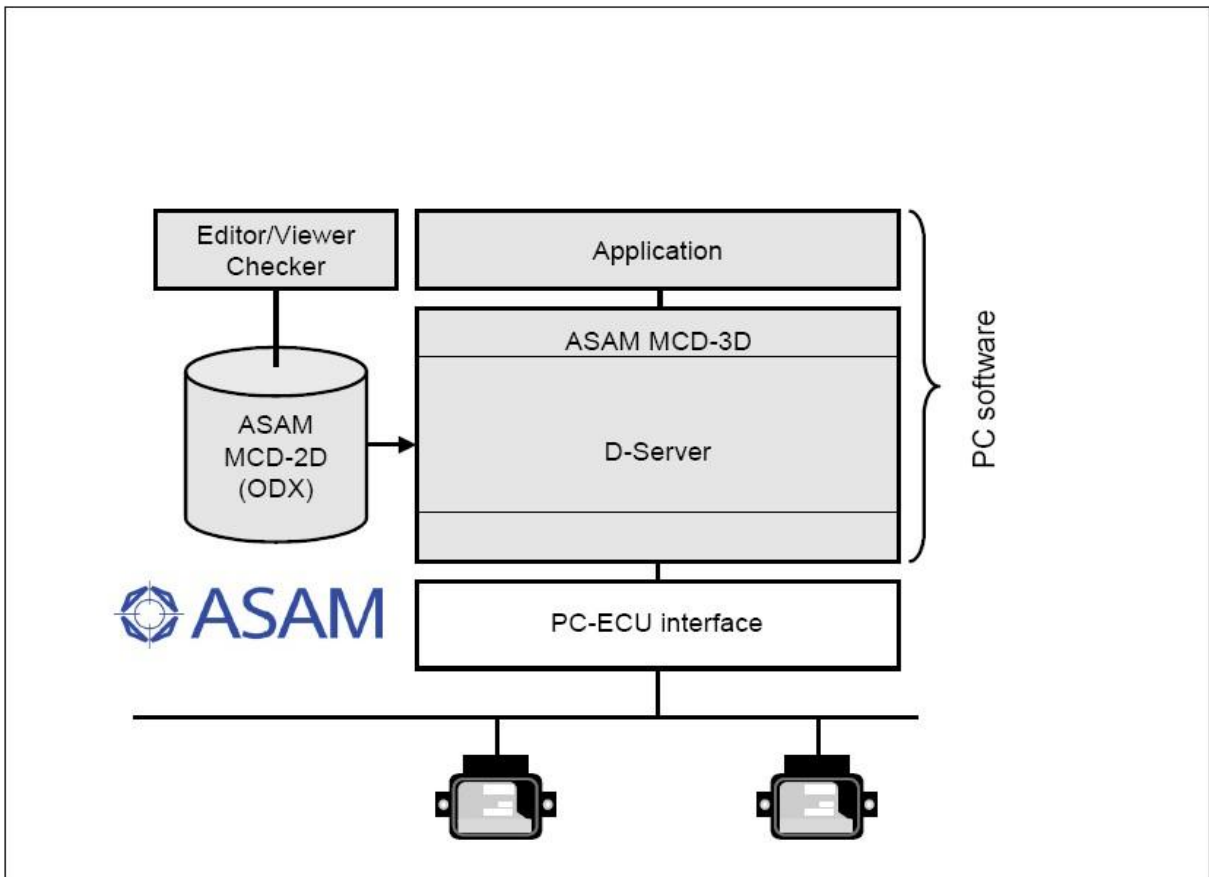


Abbildung 5: Der D-Anteil des ASAM MCD Systems

ISO 22900 - Road vehicles - Modular vehicle communication interface (MVCI) besteht aus drei Teilen:

Part 1	2008	Hardware design requirements
Part 2	2009	Diagnostic protocol data unit application programming interface (D-PDU API)
Part 3	FDIS	Diagnostic server application programming interface (D-Server API)

ISO 22901 – Road vehicles – Open diagnostic data exchange besteht aus zwei Teilen:

Part 1	2008	Data model specification
Part 2	CD	Emissions-related diagnostic data

Die Versionen MCD-2D (ODX) 2.2 und MCD-3D 2.2 wurden bei der ISO zur internationalen Standardisierung eingereicht und schließlich als ISO 22900 und ISO 22901 veröffentlicht. Abbildung 7 zeigt die Komponenten des Kommunikationssystems mit den dazu gehörenden ISO-Standards. Seit Januar 2009 ist mit der D-PDU API auch die Schnittstelle des D-Servers zum PC-ECU Interface spezifiziert.

	MCD-3 1.0.1	MCD-3 2.0.1	MCD-3 2.0.2	MCD-3 2.1.0	MCD-3D 2.2.0
ODX 1.2.2	✓				
ODX 2.0.0					
ODX 2.0.1		✓	✓		
ODX 2.1.0				(✓)	
ODX 2.2.0					(✓)

✓ = commonly used combinations (✓) = to be used in future

Abbildung 6: Die ASAM MCD-3D und MCD-2D Versionen

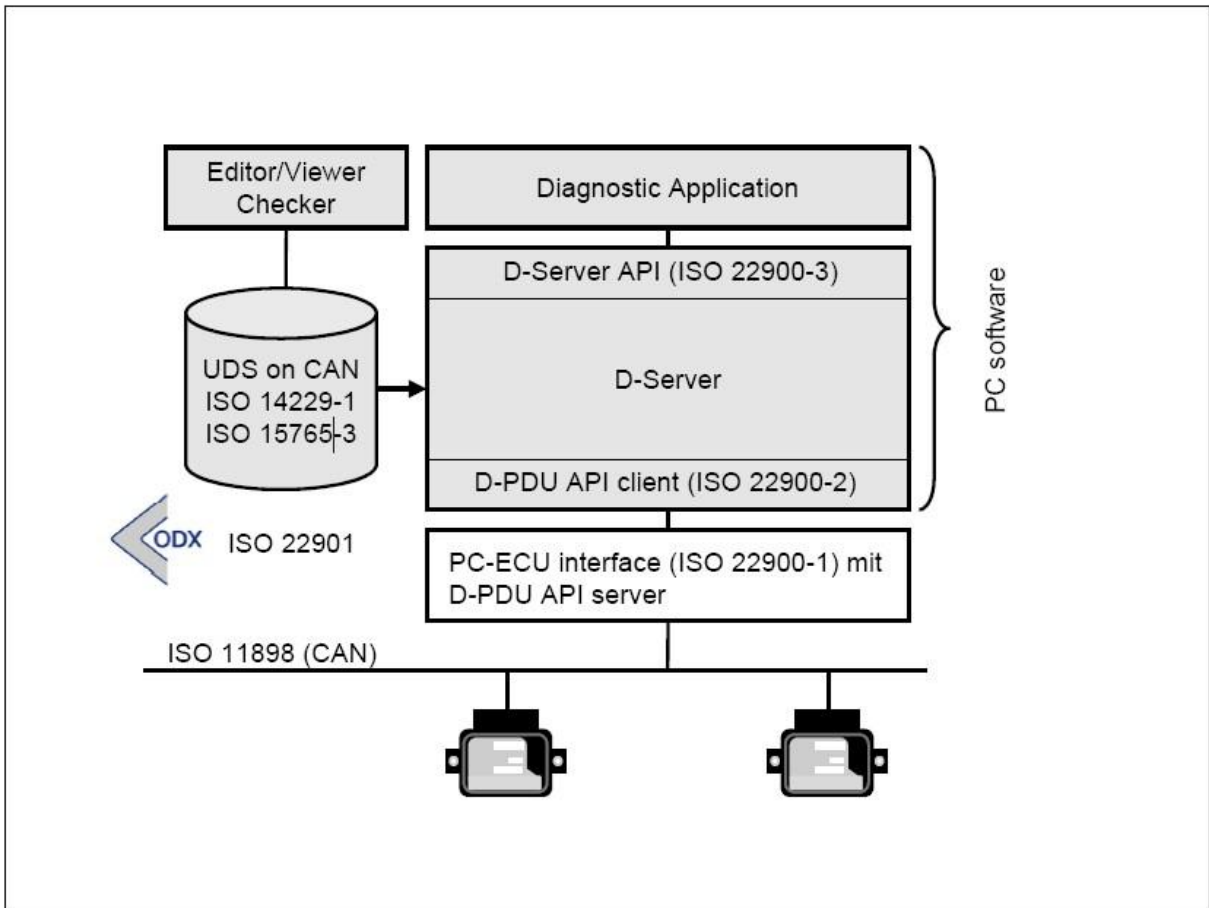


Abbildung 7: Komponenten eines international standardisierten Diagnosesystems

Literatur

- [1] Christoph Marscholik / Peter Subke: Datenkommunikation im Automobil. Hüthig-Verlag, 2007, ISBN 978-3-7785-2969-0
- [2] Christoph Marscholik / Peter Subke: Road vehicles – Diagnostic communication. Hüthig-Verlag, 2008, ISBN 978-3-7785-4048-0
- [3] ISO 11898: Controller area network (CAN)
- [4] ISO 14229: Unified diagnostic services (UDS)
- [5] ISO 15765: Diagnostics on Controller Area Network (CAN)
- [6] ISO 22900: Modular vehicle communication interface (MVCI)
- [7] ISO 22901: Open diagnostic data exchange (ODX)

Dieser Beitrag wurde unter der Leitung von Prof. Bernard Bäker im Tagungsband „Diagnose in mechatronischen Fahrzeugsystemen II“ durch den Expert Verlag veröffentlicht.