

6 Kommunikationssysteme

6.1 Übersicht

Die in diesem Abschnitt beschriebenen Kommunikationssysteme basieren auf PC-Hardware mit Windows¹ als Betriebssystem. PC-basierte Kommunikationssysteme werden in allen Bereichen der Prozesskette (Entwicklung, Produktion, Werkstatt) eingesetzt, z. B. als Software-Werkzeug in der Steuergeräte-Entwicklung, als Komponente eines Testsystems, zum Befüllen der Bremsanlage in der Fahrzeugmontage oder als Diagnosetester in der Werkstatt. Dabei spielt es prinzipiell keine Rolle, ob das Kommunikationssystem an ein einzelnes Steuergerät (z. B. im Entwicklungslabor), an mehrere Steuergeräte (z. B. beim Test von Modulen), oder an das Fahrzeug (in der Regel über den OBD-Steckverbinder) angeschlossen ist.

Je nach Einsatzgebiet erfüllen Kommunikationssysteme unterschiedliche Aufgaben, oft auch in Kombination, z. B.:

- Anzeige der Onboard-Kommunikation (Monitoring)
- Restbussimulation
- Senden und Empfangen von Diagnosebotschaften (Services)
- Aufzeichnung von Daten im Fahrversuch
- Re-Programmierung durch Flash-Download
- Variantencodierung
- Kommunikation mit dem ABS-Steuergerät zum Befüllen des Bremssystems
- Electronic Checkout (ECOS)
- Diagnose in der Werkstatt (Kundendienst)

Bild 6.1 zeigt den prinzipiellen Aufbau eines PC-basierten Kommunikationssystems für die Datenkommunikation mit Fahrzeug-Steuergeräten. Es besteht aus einer Datenbasis, einem Editor, einer Applikation (Anwendung), einem Laufzeitsystem und einem Interface.

Das *Laufzeitsystem* verbindet über drei Schnittstellen die Datenbasis, die Applikation und das Interface miteinander. Es dient der Applikation als Server, der intern komplexe Abläufe durchführt. Es steuert das Kommunikationssystem unter Einbeziehung der Datenbasis und der angeschlossenen Bussysteme.

1. Es gibt auch Kommunikationssysteme mit anderen Betriebssystemen, wie z. B. LINUX oder RTOS.

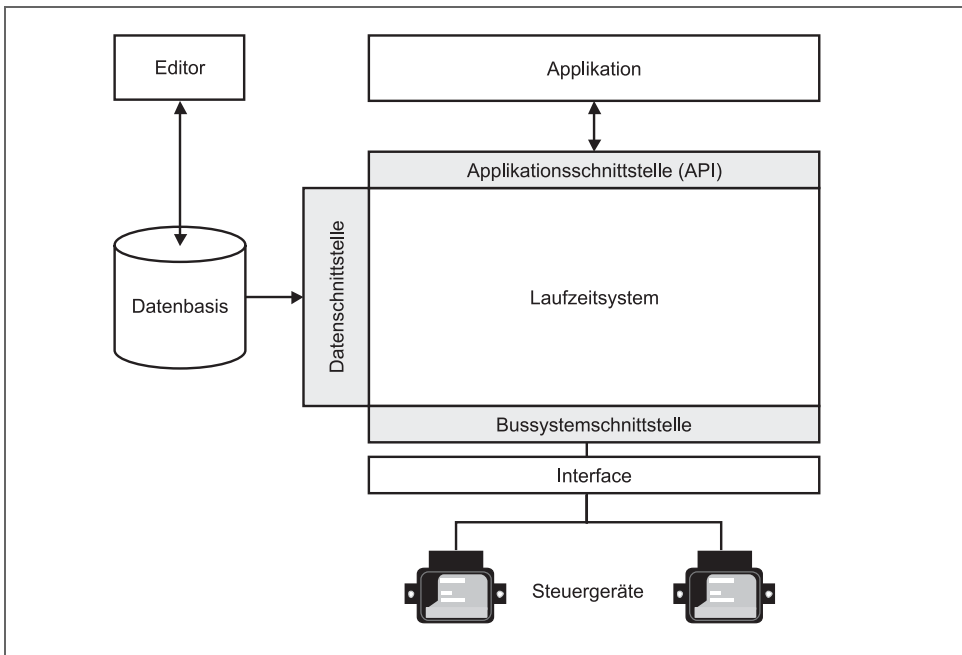


Bild 6.1: Kommunikationssysteme

Die *Applikation*² ist der Client des Laufzeitsystems und realisiert die Funktionen des Kommunikationssystems. Die Applikation kann auch eine Schnittstelle für den Benutzer beinhalten und als solche Eingaben (z. B. über die Tastatur oder die Maus) verarbeiten und Prozesse oder Daten auf einem Bildschirm visualisieren. In diesem Fall bildet die Applikation die Schnittstelle für eine Mensch-Maschine-Kommunikation und wird als *HMI (Human Machine Interface)* bezeichnet. Falls das Kommunikationssystem Teil einer automatisierten Anlage ist, kann das HMI entfallen. In Kapitel 8 wird eine Auswahl unterschiedlicher Applikationen beschrieben.

Die *Datenbasis* beinhaltet Informationen, die für die Applikation benötigt werden. In der Regel sind in der Datenbasis auch Informationen für die Datenkommunikation (z. B. Protokolle und deren Kommunikationsparameter) gespeichert. Aufbau und Struktur von Datenbanken werden in Kapitel 7 detailliert beschrieben.

Das *Interface* bildet die Schnittstelle des Laufzeitsystems zum Bussystem und damit zu den Steuergeräten. Ein Interface besteht aus Interface-Hardware (Beispiele siehe Abschnitt 6.6) und -Firmware.

Mit der Zunahme elektronischer Systeme im Fahrzeug wurden von den Fahrzeugherstellern und Zulieferern zahlreiche, nicht kompatible und proprietäre Kommunikationssysteme entwickelt. In Abschnitt 6.2 werden zwei dieser Systeme als Beispiele beschrieben.

2. Mit „Applikation“ ist hier eine Anwendung (Programm) gemeint, das interaktive oder programmgesteuerte Funktionen ausführt, nicht die „Applikation eines Motors“, bei der es um Optimierung des Motors bzgl. Abgasverhalten, Drehmoment, Verbrauch und Leistung geht.

Da der Einsatz von Standards nicht nur bei den Bussystemen, den Protokollen für die Kommunikation und der Steuergeräte-Software, sondern auch bei den externen Kommunikationssystemen Kosten reduziert und die Qualität erhöht, wurde vom ASAM e.V. und der ISO ein Kommunikationssystem mit der Bezeichnung MVCI (Modular Vehicle Communication Interface) und das Format der Datenbasis als *ODX* (Open Diagnostic Data Exchange) standardisiert. Die Standards des ASAM e.V. und das ISO-standardisierte MVCI-System werden in Abschnitt 6.3 beschrieben. Eine detaillierte Darstellung des Standards ODX findet sich in Kapitel 7.

6.2 Proprietäre Kommunikationssysteme

6.2.1 EDIS

EDIS (Elektronik **D**iagnose **I**nterface **S**ystem) besteht aus einem Interface mit der Bezeichnung EDIC (Elektronik **D**iagnose **I**nterface **C**omputer)³ und dem Laufzeitsystem EDIABAS.

EDIABAS (Elektronik **D**Iagnose **B**ASissystem) wurde von Softing im Auftrag der BMW AG entwickelt. Bild 6.2 zeigt den Aufbau eines EDIABAS-Kommunikationssystems. Es besteht aus dem EDIABAS-Laufzeitsystem, der EDIABAS-Applikation, der Datenbasis mit Steuergeräte-Beschreibungsdatei (SGBD), dem Datenbasiseditor BEST/2 und dem Diagnose-Interface EDIC.

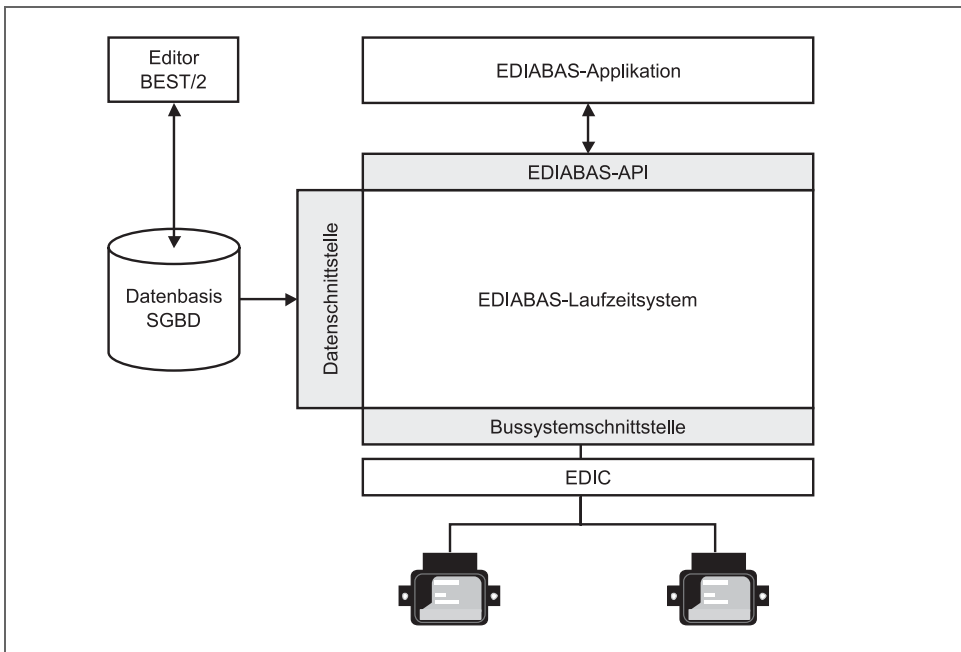


Bild 6.2: EDIS – Elektronik Diagnose Interface System

3. siehe Abschnitt 6.6

EDIABAS arbeitet mit Diensten, die als *Jobs* bezeichnet werden. Ein Job ist ein in sich geschlossener Auftrag, Daten vom Steuergerät auszulesen und auszuwerten und wird nach seiner Funktion benannt, z. B. FEHLERSPEICHER_LESEN oder MESSWERTEBLOCK_LESEN.

Ein Job liefert nach seiner Ausführung Ergebnisse, die von der Applikation verarbeitet werden. Die Jobs sind nicht Bestandteil des Laufzeitsystems, sondern Bestandteil der Datenbasis, die aus Steuergeräte-Beschreibungsdateien besteht. Für jedes Steuergerät gibt es eine separate, binär-kodierte Steuergeräte-Beschreibungsdatei (SGBD), deren Inhalt in der Regel vertraulich gehalten wird. Auch das Laufzeitsystem ist auf dem freien Markt nicht verfügbar. Der Inhalt der Steuergeräte-Beschreibungsdateien wird mit BEST/2 beschrieben. BEST/2 ist eine Beschreibungssprache für Steuergeräte mit ähnlicher Syntax wie die der Programmiersprache „C“.

EDIABAS wird in allen Bereichen der Prozesskette eingesetzt. In der Steuergeräteentwicklung für Volkswagen und AUDI ist EDIABAS Bestandteil des Diagnostesters VAS 5163⁴, der hauptsächlich zum Prüfen der Diagnosekommunikation mit dem Steuergerät verwendet wird.

Auch in der Produktion wird EDIABAS eingesetzt, zum Beispiel zur Re-Programmierung von Steuergeräten durch Flash-Download. Die Werkstatttester DIS (BMW), VAS (Volkswagen/AUDI) und DAVIE (DAF Trucks) verwenden EDIABAS als Laufzeitsystem. VAS und DAVIE werden in Kapitel 8 vorgestellt.

6.2.2 E-Tester

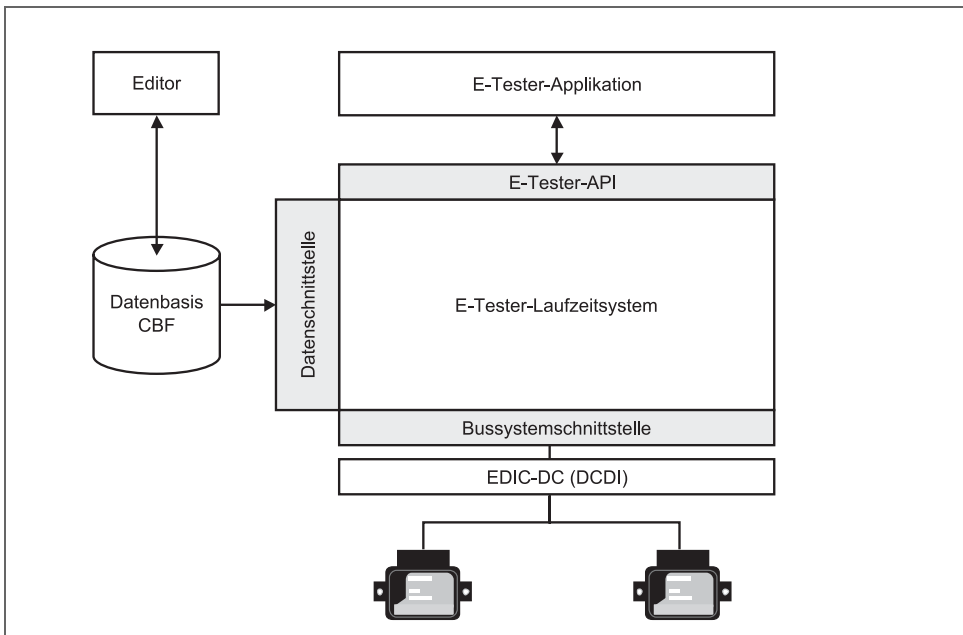


Bild 6.3: E-Tester 6

4. siehe Kapitel 8 für Details zum VAS 5163

Bild 6.3 zeigt den prinzipiellen Aufbau eines Entwicklungswerkzeugs für Steuergeräte der Daimler AG. Dieses als *E-Tester* bezeichnete System besteht aus dem Laufzeitsystem, einer Datenbasis im CBF-Format, dem Editor für die Datenbasis und der Interface-Hardware EDIC-DC⁵.

Wie die Steuergeräte-Beschreibungsdateien für das EDIABAS-Laufzeitsystem ist auch das Datenformat des E-Tester-Laufzeitsystems binär und proprietär. Für jedes Steuergerät gibt es eine separate Datei im sogenannten CBF-Format (Caesar Binary File)⁶. Als Editor wird CANdelaStudio von Vector Informatik verwendet. Bild 6.4 zeigt ein Bildschirmfoto der Applikation des E-Testers.

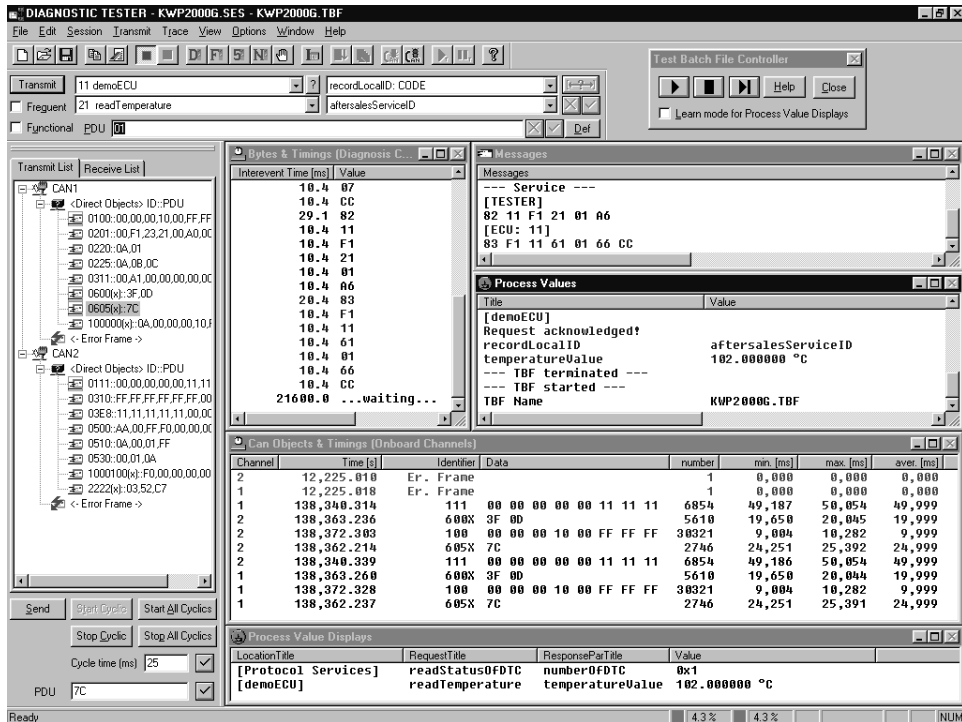


Bild 6.4: E-Tester Bildschirmfoto

Die in Bild 6.4 dargestellte Applikation wird sowohl zur Entwicklung als auch zur Prüfung der Diagnosekommunikation eingesetzt. Dazu besteht die Applikation aus mehreren Fenstern zur Steuerung der Kommunikation mit dem Steuergerät und zur Anzeige von Diagnosedaten auf unterschiedlichen Layern des ISO/OSI-Modells, z. B. Bytes & Timings, Messages und Process Values.

Eine andere Applikation auf der API des E-Tester-Laufzeitsystems ist CANDI (siehe Bild 6.5). CANDI beinhaltet eine vom Anwender konfigurierbare Benutzeroberfläche zur grafischen Darstellung der Diagnosedaten, die vom Steuergerät als Service-Responses geliefert wer-

5. siehe Abschnitt 6.6

6. CAESAR = Common Access to Electronic Systems of Automotive Requirements