

ODX- UND DEXT-DATEIEN KONSISTENT GENERIEREN

Der Lebenslauf eines Fehlercodes

Im AUTOSAR-Format DEXT spiegelt sich die Variabilität der Fahrzeugelektronik wider. In ODX von ASAM dagegen sammeln sich über die Lebensdauer eines Fahrzeuges die verschiedenen Software-Versionen eines Steuergeräts an. Fasst man alle diese Informationen in einer gemeinsamen Datenbasis zusammen, so entsteht die einzige Quelle der Wahrheit. Dadurch wird sichergestellt, dass ODX und DEXT-Dateien immer die gleichen Informationen beinhalten und damit konsistent sind.

Die heute verwendeten Fehlercodes (DTCs) gehen auf die Blinkcodes von Steuergeräten zurück, die durch Auszählen ausgelesen wurden. Mit steigender Komplexität der Elektronik und damit mehr und komplizierteren Fehlerbedingungen reichen solche proprietären Diagnose-Verfahren nicht mehr aus. So führte die Entwicklung hin zu den aktuell bekannten 2-Byte (OBD Legislation Codes) und 3-Byte DTCs (UDS standardisiert).

Aufgrund des vorhandenen Kostendruckes ist es für die Fahrzeughersteller

und ihre Zulieferer sinnvoll, hier zusammenzuarbeiten und gemeinsame Austauschformate zu benutzen. In der Elektronikentwicklung entstand so das DEXT-Format von AUTOSAR und für die OffBoard-Diagnose das vergleichbare ODX-Format des ASAM.

DTC im Steuergerät

Der Weg eines DTC von der Erkennung in einem Steuergerät bis zur Diagnose in der Werkstatt ist in Bild 1 dargestellt.

Applikationssoftware

Die Applikationssoftware überwacht zyklisch die Signale auf die Einhaltung der Grenzwerte mit Monitoren. Die Einhaltung oder Nichteinhaltung der Grenzwerte wird in Form von Ereignissen (Events) an den Diagnostic Manager (DM) signalisiert.

Die eigentliche Monitorfunktion mit ihren Grenzwerten 11,2 V und 15,2 V bleiben proprietär, jedoch werden die Events der Applikationen in DEXT definiert (Bild 2).

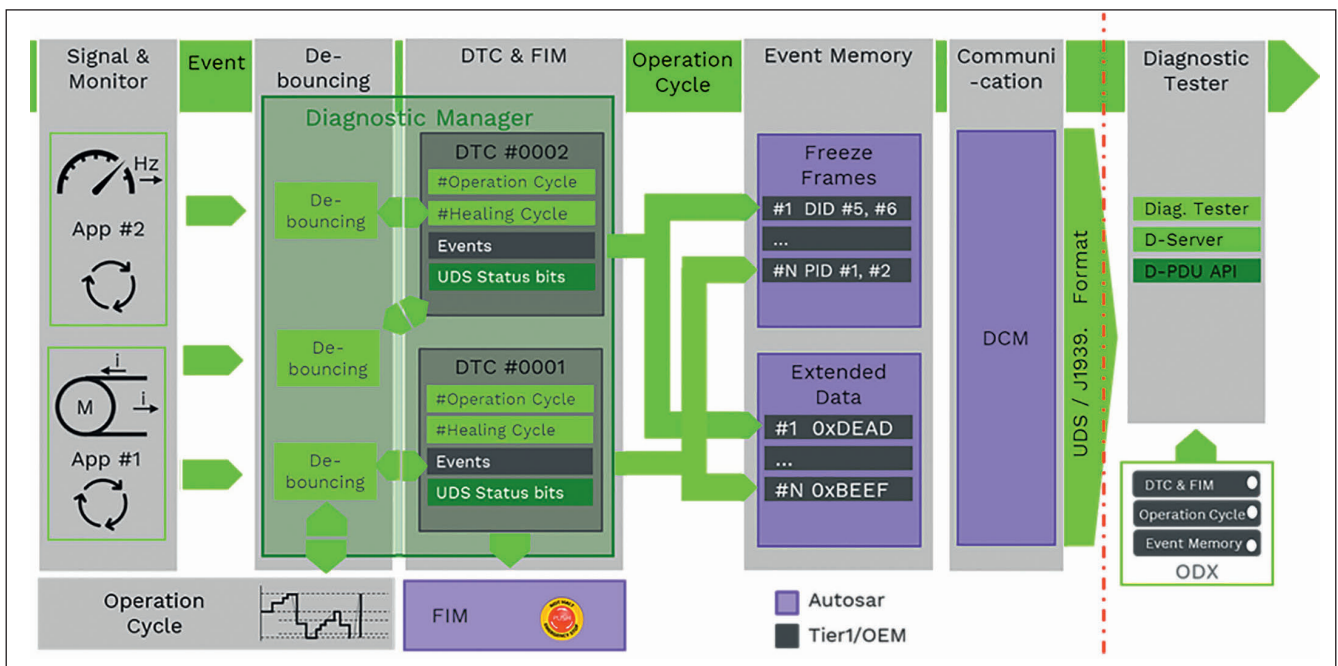


Bild 1: Prozessablauf eines Diagnostic Trouble Codes (DTC). © KPIT Technologies

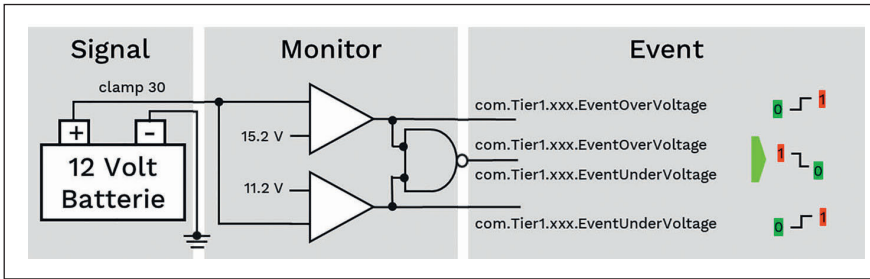


Bild 2: Ein Monitor zur Batterieüberwachung mit zwei Ereignissen (Events). © KPIT Technologies

med“ hochgezählt oder bis -128 für „Passed“ heruntergezählt. So kann ein fein granulares Entprellverhalten moduliert werden.

- Nicht entprellt (Internal Debouncing): Es findet kein Entprellen durch den DM statt und der Monitor liefert bereits entprellte Events.

Neben den drei beschriebenen Entprell-Strategien gibt es zahlreiche weitere

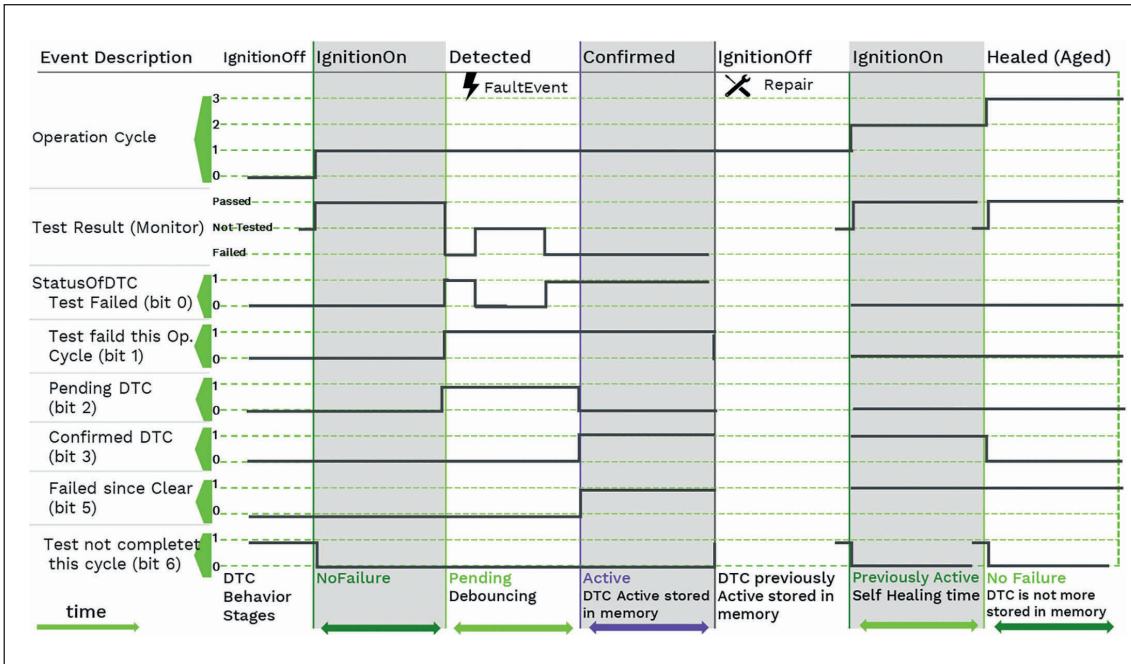


Bild 3: Das dynamische Verhalten von DTC im Fahrzeug. © KPIT Technologies

Diagnostic Events

Die Events der Monitore (typisch sind drei Zeitscheiben in Raster 10 ms, 100 ms und 1.000 ms) sind den jeweils spezifischen DTC zugeordnet und werden über die Middleware zum DM transportiert. Fehlercodes sind primär für den Aftersales-Service gedacht und von der jeweiligen Diagnosestrategie abhängig. Typischerweise legt der OEM die zu verwendenden DTC einheitlich für seine Fahrzeuge fest.

Mehrere Events können auch demselben DTC zugeordnet werden. Die beiden Events „EventOverVoltage“ und „EventUnderVoltage“ könnten hier sowohl zwei verschiedenen DTCs zugeordnet werden oder nur einen gemeinsamen DTC etwa „PowerSupply“ besitzen. Ein Autorenwerkzeug sollte hier für ein ECU die Events der Monitore mit den DTC des jeweiligen OEM verknüpfen. Da nicht jeder Monitor ein Event senden muss, aber jeder DTC im DM

mindestens 4 Byte Speicher benötigt, sind unbenutzte DTC zu entfernen.

Entprellen der Events (Debouncing)

Nicht jede kleine Störung soll sofort einen DTC erzeugen und dadurch die verbundene Fehlerreaktion auslösen. Wenn z. B. die Batteriespannung für wenige Sekunden zu gering ist, dann ist das Abschalten des ABS schlicht eine Überreaktion. Aus diesem Grund werden die meisten Events entprellt. Letztlich gibt es drei unterschiedliche Strategien, die sich erheblich in ihrer Definition unterscheiden.

- Zeitgesteuertes Entprellen (Time-based debouncing): Hier muss ein Fehler eine gewisse Zeit vorhanden sein, um als DTC wirksam zu werden (Confirmed).
- Zählergestütztes Entprellen (Counter-based debouncing): Hier wird mit verschiedenen Schrittweiten ein Fehlerzähler bis 127 für „Confir-

Einstellungen in DEXT, welche das genaue Verhalten eines DTCs beeinflussen. Beispiele sind das Speichern des Debounce-Counters in Non-Volatile Memory, der Reset des Debounce-Counters bei ClearDTC und der Reset bei einem neuen Operation-Cycle.

Funktionsdeaktivierung, Function Inhibition Manager (FIM)

Wird ein DTC als bestätigt (confirmed) gespeichert, muss das Fahrzeug angemessen auf diesen Fehler reagieren. Der Fahrzeughersteller analysiert bei der Entwicklung des Fahrzeuges solche Fehler und ihre Auswirkungen. Dies wird typischerweise mit Hilfe einer FMEA beschrieben und die entsprechenden Reaktionen des Fahrzeugs festgelegt. Der FIM setzt letztlich die Teile der FMEA im Steuergerät um.

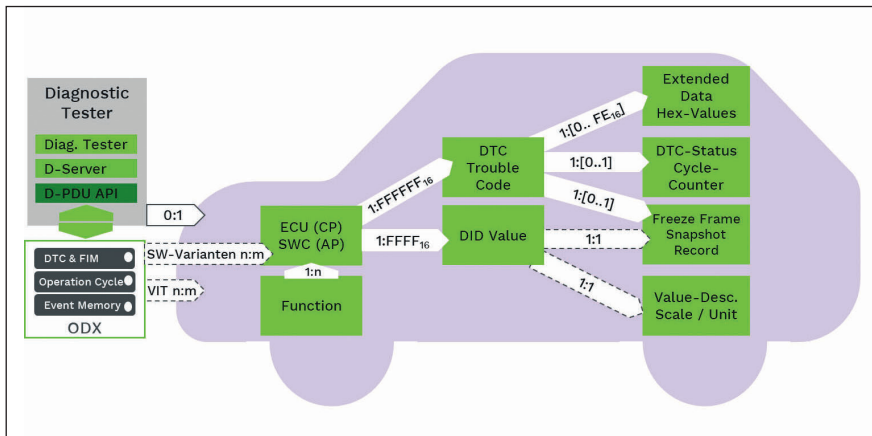


Bild 4: DTCs und die dazu zugehörigen Datenobjekte.

© KPIT Technologies

DTC im Diagnose Tester (ODX)

Je nach gewähltem 0x19-ReadDTC Subfunktion müssen neben den DTC auch noch die FreezeFrame oder die entsprechende ExtendedData Information im Tester wieder decodiert werden. Natürlich muss die Codierung der UDS-Daten in der ECU auf Basis von DEXT mit der Decodierung auf Basis von ODX identisch sein (Bild 4).

Für den Tester existieren noch zusätzliche Herausforderungen. Zunächst muss er das konkrete Fahrzeug herausfinden und die zutreffende VehicleInformationTable (VIT) aus ODX auswählen. Danach wird die genaue SW-Version der im Fahrzeug verbauten ECUs, oder im Falle von Adaptive AUTOSAR des Software Cluster (SWC), mittels einer Variantenidentifikation ermittelt. Daher enthalten die ODX-Daten jedes Steuergerätes für die verschiedenen Softwarevarianten die jeweils nötige Interpretation. Neben der Dateninterpretation und der Varianten sind auch in ODX noch weitere Daten vorhanden, wie z. B. Audience, Sprachinformation und Metainformationen, die wiederum in DEXT fehlen.

Einzige Quelle der Wahrheit (Single Source of Truth)

ODX und DEXT sind in der Praxis bewährte und weit verbreitete Austauschformate. Ein großer Teil der Inhalte von DEXT und ODX für einen DTC sind gleichwertig. Beide Formate enthalten jedoch nicht alle notwendigen Informationen für einen DTC.

In DEXT spiegelt sich die Variabilität der Fahrzeugelektronik wider. In ODX sammeln sich über die Lebensdauer eines Fahrzeuges die verschiedenen Software-Versionen eines Steuergerätes an. Fasst man alle diese Informationen in einer gemeinsamen Datenbasis zusammen, so entsteht die einzige Quelle der Wahrheit. Dadurch wird sichergestellt, dass ODX und DEXT-Dateien immer die gleichen Informationen beinhalten und damit konsistent sind.

Diese Erkenntnisse hat KPIT bereits in seinem K-DCP ECU-Editor umgesetzt.

Operation-Cycle

Der Operation-Cycle ist ein definierter Zyklus im Fahrzeug und der Puls für DTC. Er sorgt dafür, dass DTC entstehen und auch wieder ausheilen.

Angetrieben und abhängig von diesem Puls hat jeder DTC einen dynamischen Status, der in der ISO 14229-1:2013 (UDS-Standard) definiert ist. Die Bits 0 bis 6 des StatusOfDTC beschreiben dabei, wo sich der DTC in seinem Lebenslauf gerade befindet. Bild 3 stellt exemplarisch das dynamische Verhalten eines DTC mit zyklischem Monitor dar. Der Zyklus wird mit „Zündung ein“ gestartet.

Fehlerspeicher (Event-Memory)

Ein Fehlercode mit Status reicht nur bei einfachen Fehlern aus. Bei komplexeren Fehlern werden zusätzlich auch die Umweltbedingungen wie Drehzahl, Temperatur, Geschwindigkeit und ggf. weitere interne Informationen zur genaueren Analyse eines Fehlers benötigt.

DTC Snapshot Data (FreezeFrames)

Der Abruf von Messwerten ist auch ohne DTC schon über den ReadDataByIdentifizier-UDS-Service möglich. Für eine sinnvolle Analyse müssen die Messwerte auch zeitlich exakt zu dem auslösenden Ereignis passen. Wenn ein Monitor „Failed“ meldet, werden die relevanten DIDs mit ihren Messwerten zwischengespeichert.

Die Datenstruktur eines Snapshots-Records entspricht einer Aneinanderreihung von einzelnen DID aus dem ReadDataByIdentifizier-Service. Hier ist eine große Überschneidung mit ODX gegeben. Sind die bei einem DTC verwendeten

DID und ihre Datenstrukturen bekannt, so können aus der DEXT-Information die ODX-Daten für readDataByIdentifizier und readDTC generiert werden und umgekehrt.

Extended Data

Der Extended Data Block enthält zusätzliche Informationen zu einem DTC wie z. B. Cycle Counters, Aging Counters, Time of last occurrence, ... und dynamische Daten von Algorithmen, die nicht bereits in den FreezeFrame-Daten enthalten sind. Ähnlich wie bei den DIDs gibt hier eine 1-Byte-Recordnummer die Datenstruktur des Extended-Data-Blocks an. Zu einem DTC können bis zu 255 gespeicherte ExtendedDataRecords existieren. Die Interpretation der Daten ergibt sich eindeutig mittels ihrer RecordNumber.

Die Beschreibung von Snapshot- und Extended-Data sind sich inhaltlich prinzipiell bei ODX und DEXT sehr ähnlich. Es gibt jedoch eine Vielzahl von weiteren Informationen in DEXT, wie z. B. Severity, Debouncing, Malfunction Indicator, Function Inhibit, die nicht in ODX enthalten sind, jedoch für einen DTC in einer ECU zwingend benötigt werden.

Diagnostic Communication Manager (DCM)

Der letzte Schritt auf dem Weg zum Tester ist der DCM, der für die Codierung und Übermittlung der Daten an den Tester verantwortlich ist. Mit dem ReadDTC (0x19) UDS-Service wird ein DTC als Zwei- oder Drei-Byte-Wert entsprechend des eingestellten DTC-Formats codiert und über den Fahrzeugbus gesendet.

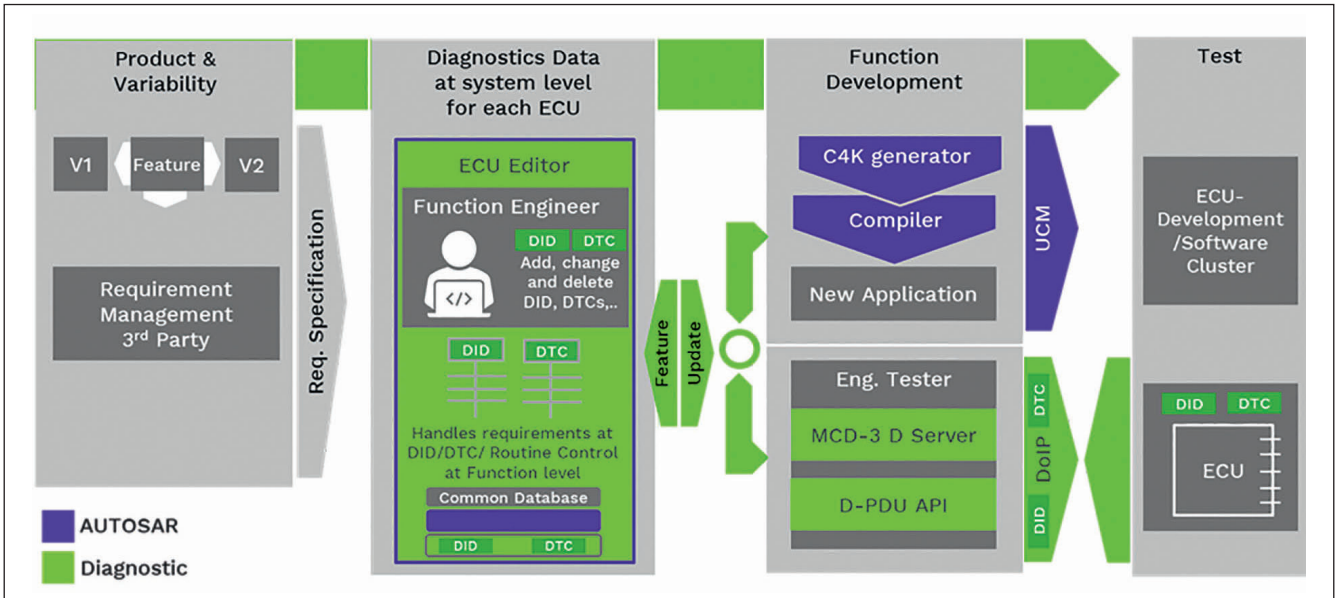


Bild 5: Einheitlicher Bedatungsprozess für DTCs und DIDs. © KPIT Technologies

Die SW-Entwickler können sich auf die technischen Eigenschaften eines DTC und die relevanten Informationen fokussieren und brauchen keine Format-Details zu kennen. Die Diagnose-Entwickler pflegen dann noch zusätzlich diagnose-relevante Metadaten (Texte, Varianten, ...) ein. Durch die gemeinsame Ablage dieser Daten im KPIT ECU-Editor, werden die notwendigen ODX- und

DEXT-Dateien konsistent, redundanz- und widerspruchsfrei auf Knopfdruck generiert und auch wieder importiert.

Ein derartiger gemeinsamer Prozess mit einer einzigen gemeinsamen Datenbasis ist bei der Komplexität von DTCs mit ihren Abhängigkeiten zu Extended- und FreezeFrame-Daten, der aufgrund der Variabilität der ECU und Varianten der SW entstehenden Vielfalt fast schon

zwingend. Siehe hierzu auch den Han- ser-Artikel „Diagnose in Adaptive AUTO- SAR“ aus Heft 10/2019. ■ (oe)

www.kpit.com



Dipl. Informatiker Bernhard Wagner, MBA ist Diagnostics Subject Matter Expert bei KPIT Technologies.

Hybrid-Polymer-Aluminium-Elektrolyt-Kondensatoren in axialem Design

TDK hat das Produktspektrum an Hybrid-Polymer-Aluminium-Elektrolyt-Kondensatoren erweitert und bietet nun zwei Serien in axialem Design an. Für Nennspannungen von 25 V und 35 V eignet sich die Serie B40600 / B40700. Diese deckt ein Kapazitätsspektrum von 780 µF bis 2200 µF ab. Mit der Serie B40640 / B4074 sind zudem Kondensatoren in der Spannungs-klasse 63 V verfügbar mit Kapazitätswerten von 390 µF bis 720 µF. Die Kondensatoren werden in vier Bechergößen von 14 x 25 mm² bis 16 x 30 mm² (D x H) produziert. Erhältlich sind diese in axial-lead Bauform (B406) oder als Lötstern-Ausführung (B407), wodurch vielfältige Montagemöglichkeiten gegeben sind. Der innere Aufbau ist für die thermische Anbindung an einen Kühlkörper optimiert.

Beide Serien sind für eine Betriebstemperatur von -55 °C bis +150 °C ausgelegt, wobei eine Lebensdauer von ca. 4000 h bei 125 °C erreicht wird. Der große Vorteil der Hybrid-Polymer-Technologie liegt jedoch in den extrem geringen ESR-Werten über den großen Temperaturbereich hinweg. Beispielsweise kann bei einer Bechergöße von 16 x 30 mm² und einer Umgebungstemperatur von 20 °C ein ESR von 3,5 mΩ erreicht werden. Typische Applikationen sind bidirektionale Wandler für 48-V-Bordnetze, Motorinverter von Hybridantrieben, elektrisch angetriebene Servolenkungen, Pumpen, elektrische Turbo- und Supercharger, Kühler-ventilatoren, Getriebesteuerungen und Ausgangsfilter für Schaltnetz-teile.

www.tdk-electronics.tdk.com/de/alu_polymer

