



PRODUKT-DATENBLATT

TRACE-CHECK beschleunigt fundierte Analysen

TRACE-CHECK ist ein Werkzeug, mit dem aufgezeichnete Messdaten umfassend und automatisch analysiert werden können. Es ist besonders dort hilfreich, wo signalbasierte Testergebnisse in Form von Reports gebraucht werden, die leicht lesbar und schnell zu interpretieren sind – unabhängig davon, aus welchen Aufzeichnungssystemen die Daten stammen oder in welche Tool-Kette das Werkzeug integriert werden soll.

Leistungsmerkmale im Überblick

- Einfache Analyse-Spezifikation durch
 - Getriggerte Analysen
 - Timing-Diagramme
 - Python-Schnittstelle
- Unterstützung zahlreicher Aufzeichnungsformate
- Hohe Wiederverwendbarkeit der Analysen
- Intuitive grafische Bedienoberfläche
- Aussagekräftige Ergebnisdarstellung
 - Übergang zum interaktiven SignalViewer
 - Mit Ergebnisdaten angereicherte Plots

Unterstützte Formate

Busbeschreibung:

- ARXML (Classic Platform) 4.1.1 bis 4.3.1
- ARXML (Adaptive Platform) bis 10/18
- DBC
- FIBEX bis 4.1.1/FIBEX für Ethernet 4.1.2
- LIN Description File (LDF)

Steuergerätebeschreibung:

- ASAP2 Datenbank (A2L)
- Executable and Linkable Format (ELF)
- Intel HEX
- Motorola S19

Rohdaten-Trace-Formate:

- ASC, BLF, TTL: CAN- und FlexRay-Aufzeichnungen
- PCAP: Ethernet-Aufzeichnungen

Signalbasierte Trace-Formate:

- CSV
- ERG: CarMaker
- ECAL 5.0, 5.1
- MAT: MATLAB/Simulink, ControlDesk
- MDF 3.0, 3.1, 3.2, 3.3, 4.0, 4.1
- RDB: VTD
- STI, STZ 2.0.1 ASAM XiL-API
- TDMS: National Instruments

Schnittstellen

Über **Automatisierungsschnittstellen** (COM) lassen sich viele TRACE-CHECK-Arbeitsschritte ansteuern, zum Beispiel für einen nahtlosen und vollautomatischen Betrieb in einer bestehenden Werkzeugkette. Auf Anfrage stellen wir auch gerne eine angepasste Variante des **ECU-TEST Jenkins Plug-ins** bereit.

Spezifikationsmittel

Dem Nutzer stehen verschiedene Beschreibungssprachen zur Verfügung, um Anforderungen in Form von Analyse-Bausteinen zu formalisieren:

In **Trigger-Blöcken** und **Berechnungsschritten** können beliebige logische Ausdrücke aus Signalnamen, Package-Variablen und internen Funktionen genutzt werden.

Mit **Timing-Diagrammen** lassen sich sowohl einfache als auch komplexe Signal-Zusammenhänge leicht verständlich und gleichzeitig formal beschreiben.

Die **Python-Schnittstelle** erlaubt es, die Analyse-Vorschriften selbst zu programmieren. Dabei kann sich der Benutzer ganz auf die eigentliche Anforderung konzentrieren. Viele Aufgaben, vom Einlesen der Traces über die Interpolation bei unterschiedlichen Zeitachsen bis hin zur Erstellung von Reporteinträgen, werden vollautomatisch von TRACE-CHECK erledigt.

Für die effiziente Verarbeitung der Signaldaten werden dem Benutzer alle Funktionalitäten der Programmibliotheken NumPy und SciPy zur Verfügung gestellt.

Zur Synchronisation multipler Aufnahmen stellt TRACE-CHECK verschiedene Methoden bereit:

- EqualnessMatching
- ExpectationMatching
- Offset