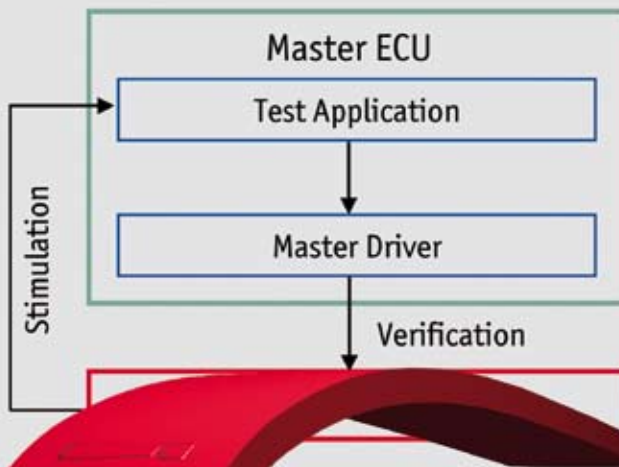
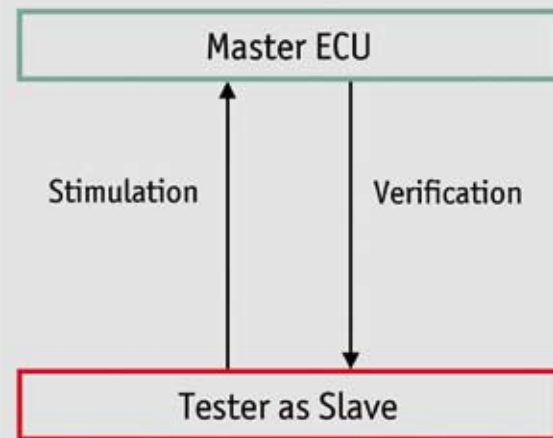


Software



Gray Box Test



Black Box Test



Aktuelle und zukünftige Strategien für LIN-Master-Konformitätstests

Eine deutliche Zunahme LIN-basierter (Local Interconnect Network) Komponenten in den Automobilentwicklungen ist zu verzeichnen. Effiziente Teststrategien für dieses kostenoptimierte Bussystem gewinnen deshalb an Bedeutung. Vector Informatik zeigt, wie weit die Möglichkeiten auf Grundlage aktueller LIN-Testspezifikationen bei Konformitätstests für Master und Slave heute reichen. Anhand des Entwicklungs- und Simulationstools CANoe.LIN wird zudem eine anwendergerechte Teststrategie für Black-Box-Mastertests aufgezeigt, wie sie mit Standardmitteln bislang nicht realisierbar ist.

1 Einleitung

Vom LIN-Konsortium stammen nicht nur die grundlegenden LIN-Spezifikationen für die verschiedenen Protokollversionen, sondern auch Testspezifikationen für Konformitätstests. Diese Tests verifizieren, ob der Prüfling die entsprechende Protokollspezifikation einhält und dienen auch als Basis für die Akkreditierung. Da LIN-Netze nach dem Master-Slave-Prinzip arbeiten, ist die Protokollkonformität des Masters von zentraler Bedeutung. Die LIN-Konformitätstests sind spezifiziert für die verschiedenen OSI-Layer: Physical Layer, Data-Link-Layer und Network-Management sowie Node-Configuration. Nur die Application Layertests müssen von OEM oder Zulieferer selbst spezifiziert werden.

Zur methodischen Umsetzung von Konformitätstests eignen sich Black-Box-Tests am besten. Diese sind dadurch charakterisiert, dass man zum Stimulieren und Verifizieren ausschließlich externe Schnittstellen heranzieht, zum Beispiel die LIN-Schnittstelle des Steuergeräts (ECU). Dem gegenüber ist ein White-Box-Test stets mit einem Eingriff in die Interna des Steuergeräts, beispielsweise die standardisierte Schnittstelle des LIN-Treibers, verbunden. Da ein LIN-Master-Steuerggerät über seine externen Schnittstellen nur wenige Stimulationsmöglichkeiten anbietet, zum Beispiel über CAN, sind Gray-Box-Tests, in denen beide Testmethoden kombiniert werden, derzeit die gebräuchlichste Methode für einen

LIN- Master-Konformitätstest, **Bild 1**.

2 Testumgebung für LIN-Konformitätstests

Der Ablauf eines konkreten Testfalls am Beispiel des Test- und Simulationstools CANoe.LIN von Vector Informatik besteht aus Konfiguration und Initialisierung von Testsystem und zu testendem Steuergerät sowie anschließender Stimulation und Verifikation. Die Slave-Konformitätstests sind nahezu vollständig als Black-Box-Test implementiert, weil hier vom Master eine direkte Stimulation und Verifikation über den LIN-Bus möglich ist. Beim Master-Konformitätstest kommt der Gray-Box-Test zur Anwendung. Dazu liefert der Stuttgarter Toolhersteller ein spezielles Test LIN Description File (LDF) zusammen mit einer Testapplikation aus, die auf das Master-Steuerggerät zu laden sind. Zur Verifikation nutzt man den LIN-Bus.

Etliche LIN-Test-Erfahrungen in Verbindung mit CANoe.LIN hat beispielsweise das Delphi Technical Center Krakau, Polen, gesammelt. Die Entwicklungs- und Testarbeiten beziehen sich sowohl auf Projekte mit LIN 2.0 als auch mit dem J2602-Protokoll, dem amerikanischen LIN-Pendant. Nachdem sich gezeigt hat, dass die LIN-Konformitätstests allein auf Basis der OSI-Schichten nicht ausreichen, wurden die Testaktivitäten in Richtung verschiedener Anwendungstests ausgedehnt. Der Fokus liegt dabei auf dem Tes-

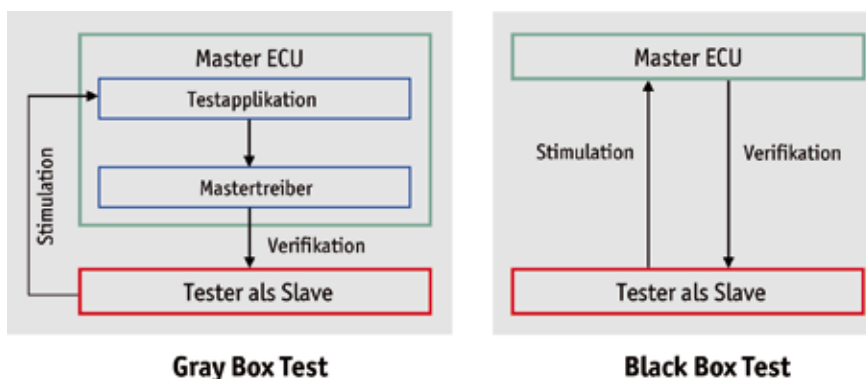


Bild 1: Der LIN-Master-Konformitätstest wird momentan als Gray-Box-Test implementiert; ein Black-Box-Test wäre aber zu bevorzugen

Der Autor



Gavin C. Rogers, B.Eng M.Sc., ist Teamleiter und Product Manager für LIN-Tools bei Vector Informatik in Stuttgart.

ten von LIN-Signalen, der Ablaufplan des LIN-Masters, des Gateways sowie auf Diagnosetests. Als unverzichtbar für die Umsetzung und Automatisierung der Testabläufe haben sich die CAPL-Testfunktionen erwiesen, die CANoe.LIN zur Verfügung stellt. Selbst sehr komplexe Test Cases können in der C-ähnlichen CAPL-Syntax leicht implementiert und erweitert werden.

3 Nachteile von Gray-Box-Mastertests

Aus Anwendersicht scheint jedoch die Realisierung des Master, Konformitätstest als Gray-Box-Test noch nicht die optimale Lösung zu sein. Während der Entwicklung ist kein durchgängiges Testen möglich und die Tests beziehungsweise Testvorbereitungen sind aufgrund der Konfiguration des Treibers mit dem Test-LDF und der jeweiligen Neu-Kompilierung mit erheblichem Aufwand verbunden. Außerdem ist seitens der Zulieferer Unterstützung hinsichtlich Compiler und Download-Tools notwendig.

Beim Gray-Box-Test wird mit Hilfe einer Test-Applikation und Test-LDF direkt auf den Master-Treiber zugegriffen. Zwar lässt sich hierdurch eine volle Testabdeckung erreichen, doch ist damit der wesentliche Nachteil verbunden, dass der übliche V-Modell-Entwicklungsprozess nicht eingehalten werden kann. Eine Verifikation der Master-Konformität ist nur am Anfang des Entwicklungsprozesses möglich. Sobald die produktive Datenbasis (LDF) und Applikation auf das Master-Steuerggerät laufen, sind weitere Verifikationen während der Entwicklung nicht mehr durchführbar. Auch ein Konformitätstest durch den OEM ist nicht einfach möglich.

4 Der Weg zum Black-Box-Mastertest

Vielfach wird daher die Frage von Fahrzeughersteller und -zulieferern gestellt, ob der Master-Konformitätstest nicht als Black-Box-Test realisiert werden kann. So wären die Prüfungen während des kompletten Entwicklungsprozesses mit minimalem Konfigurationsaufwand sowohl vom OEM als auch vom Zulieferer unabhängig möglich.

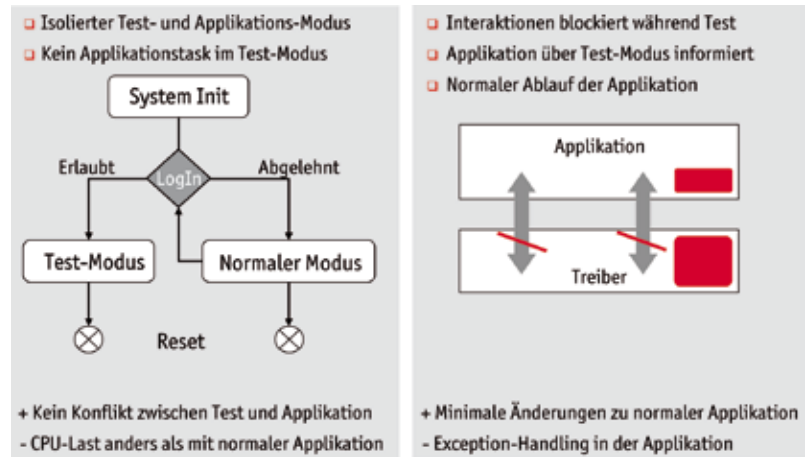


Bild 2: Die erforderliche Testkommunikation für die Realisierung des LIN Master-Konformitätstest als Black-Box-Test

Damit ein LIN-Master-Black-Box-Test echte Vorteile gegenüber einem Gray-Box-Test hat, müssen einige Voraussetzungen erfüllt sein. Zum Beispiel sollte es möglich sein, die identische Treibersoftware während der Tests und der Entwicklung zu nutzen. Dies ist beispielsweise durch eine Erweiterung des LIN-Master-Treibers um eine Testschnittstelle möglich. Weiterhin soll dieser Treiber stets mit dem produktiven LDF konfiguriert werden, so dass das gleiche LDF für Test und Entwicklung verwendet werden kann.

Zur Stimulation und Verifikation des Master-Steuergärts über den LIN-Bus benötigt eine solche Treiberschnittstelle noch spezielle Testdienste, um beispielsweise einen Schedule-Wechsel anzutriggern und das Treiber-Statuswort zu lesen. Die Testkommunikation auf dem LIN-Bus könnte, vergleichbar mit der Rekonfiguration, als spezieller Diagnose-Service ablaufen. Bei Erweiterungen wie dieser kommt es selbstverständlich immer auf einen minimalen Ressourcenverbrauch an. Weiterhin muss die Testschnittstelle von dem Produktiv-Code entfernbar sein,

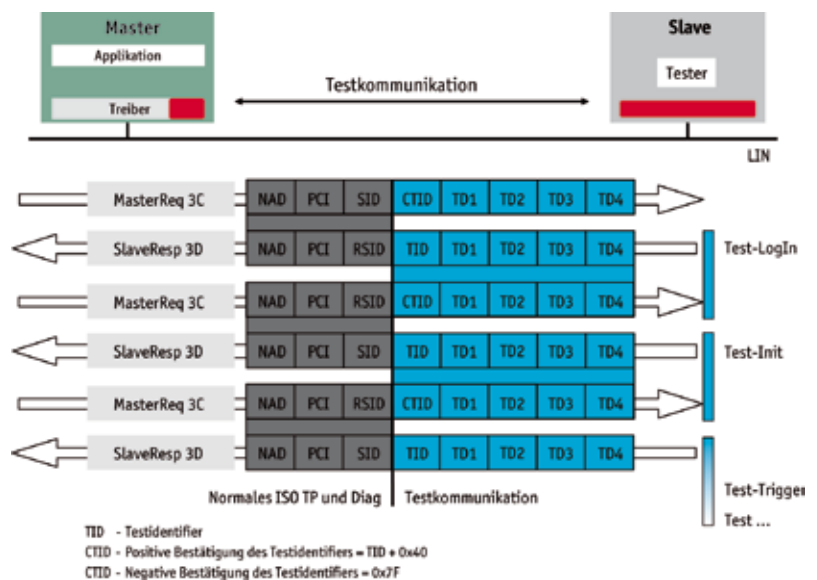


Bild 3: Zwei mögliche Strategien für die harmonische Interaktion zwischen der Master-Applikation und dem Test-Ablauf

Test-Kommandos	Parameter	Test-ID
Test LogIn	keine	0x00
Test Presence / Idle	keine	0x01
Test Setup / Trigger	T _{base} Cycles, Reset/Init indication	0x02
Load Parameters	Offset, D1, D2, D3	0x03
Read Result	Offset, D1, D2, D3, D4	0x04
Set schedule table	Table index, Slot index, T _{base} Cycles	0x05
Read Status Word	PID / keine	0x06
Send Config Command	RCSID, RCNAD, RCPCI	0x07
Request Sleep Mode	Sleep cyc., Schedule tab, wakeup Del.	0x08
Set signal	Signal handle, Signal value	0x09
Read signal	Signal handle	0x0A

Bild 4: Eine Übersicht aller 11 Test-Kommandos, die für eine vollständige Durchführung des aktuellen LIN2.0- Konformitäts-Tests erforderlich sind

zum Beispiel über einen Präprozessor-Schalter.

5 Prototyp eines Black-Box-Test für LIN 2.x

Die LIN-Spezialisten von Vector haben auf Wunsch eines OEM einen Prototyp des Black-Box-Master-Konformitätstest konzipiert und spezifiziert. Hierbei ergab sich die wichtige Erkenntnis, dass die Testkommunikation die normale Master-Funktionalität nicht beeinträchtigen darf. Der Weg über eine Master-Request-Slave-Response-Diagnose erfordert deshalb die Definition eines speziellen Diagnose-Services. Während normalerweise der Master die Kommunikation mit den Slaves dominiert, muss es bei der Testkommunikation genau umgekehrt sein. Der Slave schickt in der Rolle des Testers ein Testkommando an den Mas-

ter, auf das dieser nun positiv oder negativ reagiert, **Bild 2**.

Es ist unumgänglich jeweils einen Test-Login und eine Test-Initialisierung durchzuführen. Dies wirft die grundsätzliche Frage auf, inwiefern der Test unabhängig von der Applikation realisierbar ist. Hier hat sich herauskristallisiert, dass ein völlig unabhängiges Nebeneinander von Testdurchführung auf der einen Seite und Applikations-Algorithmen auf der anderen Seite nicht möglich ist. Die Anwendung ist folglich mit einzubinden.

Zur Interaktion zwischen Anwendung und Treiber für Testmodus stehen letztendlich zwei Strategien zur Verfügung, **Bild 3**. Alternativ zu einem Modell, das die Applikation klar darüber informiert, ob ein Test durchgeführt wird oder nicht, kann man diese Information vor der Applikation so weit wie möglich verschleiern. Welcher Weg zu beschreiten ist und welche Strategie mit ihren jeweiligen

Vor- und Nachteilen sich als geeigneter herausstellt, hängt nicht zuletzt vom Feedback und den Wünschen des Master-Lieferants ab. In jedem Fall bedeutet es, dass der existierende Treiber, so wie er heute standardisiert ist, um eine gewisse Test-Server-Fähigkeit erweitert werden müsste. Der dafür erforderliche Zusatzcode liegt schätzungsweise bei zusätzlichen 20 bis 30% – bezogen auf den aktuellen Treiberumfang, was für Master-Steuergeräte in der Regel kein Problem darstellt.

Die Prototyp-Implementierung von Vector umfasst momentan elf Services, mit denen es möglich ist, die Spezifikationen des aktuellen LIN 2.0 Konformitätstests vollständig zu erfüllen, **Bild 4**. Aus Sicht des Herstellers ist das Konzept nicht nur grundsätzlich erweiterbar, sondern auch standardisierbar. Denn einerseits basiert es auf dem existierenden LIN-Description-File (LDF) samt passendem Entwicklungsprozess, andererseits sind die Kommandos offen gelegt und erweiterbar.

6 Ausblick

Vector entwickelt CANoe.LIN weiter. Nachdem die Veröffentlichung der LIN 2.1-Konformitätstest-Spezifikationen für das zweite Quartal 2008 erwartet wird, ist eine entsprechende Unterstützung von LIN 2.1-Slavetests durch CANoe.LIN in Release 7.0 SP5 für das dritte Quartal 2008 eingeplant, sowie die LIN 2.1-Mastertests in Release 7.1 im vierten Quartal 2008. ■

Download des Beitrags unter
www.ATZonline.de

ATZ
online

ATZ
elektronik

Read the English e-magazine.
Order your test issue now:
viwegteubner@abo-service.info