

Echtzeit-Ethernet Restbussimulation: Frühzeitiges Modellbasiertes Testen in Fahrzeugnetzwerken der nächsten Generation

Masterkolloquium

Florian Bartols

Hochschule für Angewandte Wissenschaften Hamburg

17. April 2014



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Echtzeit-Ethernet Restbussimulation: Frühzeitiges Modellbasiertes Testen in Fahrzeugnetzwerken der nächsten Generation

Masterkolloquium

Florian Bartols

Hochschule für Angewandte Wissenschaften Hamburg

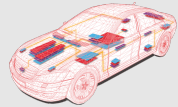
17. April 2014



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Motivation & Einleitung

Elektrische Komponenten im Automobil



- > 70 Steuergeräte im Oberklassefahrzeug
- Einsatz für Sicherheit, Komfort und Unterhaltung
- ≈ 2000 Nachrichten
- ≈ 4000 Informationen

Motivation & Einleitung

Elektrische Komponenten im Automobil



RT-Ethernet Restbussimulation

F. Bartols

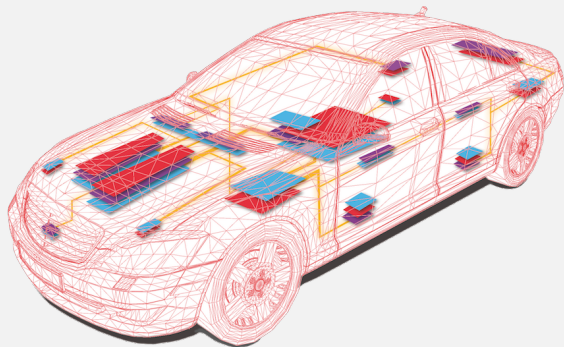
Motivation & Einleitung

Hintergrund

Echtzeit Ethernet Restbussimulation

Anwendung & Ergebnisse

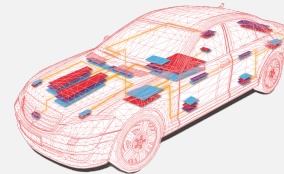
Zusammenfassung & Ausblick



- > 70 Steuergeräte im Oberklassefahrzeug
- Einsatz für Sicherheit, Komfort und Unterhaltung
- ≈ 2000 Nachrichten
- ≈ 4000 Informationen

Motivation & Einleitung

Elektrische Komponenten im Automobil



- > 70 Steuergeräte im Oberklassefahrzeug
- Einsatz für Sicherheit, Komfort und Unterhaltung
- ≈ 2000 Nachrichten
- ≈ 4000 Informationen

Motivation & Einleitung

Frühzeitiges Testen von Elektronik

- Stetige Zunahme von Elektronik hat einen großen Einfluss auf die Kosten
- 1985 ca. 3% Heute über 25%
- Funktionen werden heute in Software realisiert
- Frühzeitiges Testen wichtig für die Kontrolle der Entwicklungskosten
- Dezentrale Entwicklung erschwert das Frühzeitige Testen
- Restbussimulation ermöglicht frühzeitiges Testen

1. Moderne Fahrzeuge sind komplexe verteilte Systeme, deren Funktionsumfang stetig wächst
2. Mehr als 70 Steuergeräten sorgen für die Sicherheit, Komfort und Unterhaltung
3. Innerhalb dieses Systems werden ca. verschiedene 2000 Nachrichten ausgetauscht
4. Diese Nachrichten beinhalten ca. 4000 verschiedene Signale die verschiedene Informationen beinhalten

Motivation & Einleitung

Frühzeitiges Testen von Elektronik



RT-Ethernet Restbussimulation

F. Bartols

Motivation & Einleitung

Hintergrund

Echtzeit Ethernet Restbussimulation

Anwendung & Ergebnisse

Zusammenfassung & Ausblick

- Stetige Zunahme von Elektronik hat einen großen Einfluss auf die Kosten
- 1985 ca. 3% Heute über 25%
- Funktionen werden heute in Software realisiert
- Frühzeitiges Testen wichtig für die Kontrolle der Entwicklungskosten
- Dezentrale Entwicklung erschwert das Frühzeitige Testen

■ Restbussimulation ermöglicht frühzeitiges Testen

Motivation & Einleitung

Frühzeitiges Testen von Elektronik

- Stetige Zunahme von Elektronik hat einen großen Einfluss auf die Kosten
 - 1985 ca. 3% Heute über 25%
 - Funktionen werden heute in Software realisiert
 - Frühzeitiges Testen wichtig für die Kontrolle der Entwicklungskosten
 - Dezentrale Entwicklung erschwert das Frühzeitige Testen
- Restbussimulation ermöglicht frühzeitiges Testen

Motivation & Einleitung

Frühzeitiges Testen von Elektronik

- Stetige Zunahme von Elektronik hat einen großen Einfluss auf die Kosten
- 1985 ca. 3% Heute über 25%
- Funktionen werden heute in Software realisiert
- Frühzeitiges Testen wichtig für die Kontrolle der Entwicklungskosten
- Dezentrale Entwicklung erschwert das Frühzeitige Testen

■ Restbussimulation ermöglicht frühzeitiges Testen

1. Durch den stetigen Funktionszuwachs nimmt auch die Elektronik im Automobil stetig zu und hat einen großen Einfluss auf die Kosten
2. 1985 lag der Anteil bei ca. 3% Heute liegt er bei ca. 25%

Motivation & Einleitung

Frühzeitiges Testen von Elektronik



RT-Ethernet Restbussimulation

F. Bartols

Motivation & Einleitung

Hintergrund

Echtzeit Ethernet Restbussimulation

Anwendung & Ergebnisse

Zusammenfassung & Ausblick

- Stetige Zunahme von Elektronik hat einen großen Einfluss auf die Kosten
- 1985 ca. 3% Heute über 25%
- Funktionen werden heute in Software realisiert
- Frühzeitiges Testen wichtig für die Kontrolle der Entwicklungskosten
- Dezentrale Entwicklung erschwert das Frühzeitige Testen

■ Restbussimulation ermöglicht frühzeitiges Testen

Motivation & Einleitung

Frühzeitiges Testen von Elektronik

- Stetige Zunahme von Elektronik hat einen großen Einfluss auf die Kosten
- 1985 ca. 3% Heute über 25%
- Funktionen werden heute in Software realisiert
- Frühzeitiges Testen wichtig für die Kontrolle der Entwicklungskosten
- Dezentrale Entwicklung erschwert das Frühzeitige Testen

■ Restbussimulation ermöglicht frühzeitiges Testen

Motivation & Einleitung

Frühzeitiges Testen von Elektronik

- Stetige Zunahme von Elektronik hat einen großen Einfluss auf die Kosten
- 1985 ca. 3% Heute über 25%
- Funktionen werden heute in Software realisiert
- Frühzeitiges Testen wichtig für die Kontrolle der Entwicklungskosten
- Dezentrale Entwicklung erschwert das Frühzeitige Testen

■ Restbussimulation ermöglicht frühzeitiges Testen

1. Neue Funktionen im Automobil werden in Software realisiert
2. Damit die Kosten kontrollierbar sind, muss frühzeitig getestet werden.
3. Je später Fehler aufgedeckt werden, desto höher sind die Kosten in der Beseitigung
4. Der dezentrale Entwicklungsprozess erschwert das frühzeitige Testen, da viele verschiedene Unternehmen bei der Entwicklung beteiligt sind
5. Systeme werden von Zulieferern auf Basis von Spezifikationen die vom Hersteller verteilt werden, entwickelt
6. Teile des Verteilten Systems können zum Entwicklungszeitpunkt

Motivation & Einleitung

Frühzeitiges Testen von Elektronik



RT-Ethernet Restbussimulation

F. Bartols

Motivation & Einleitung

Hintergrund

Echtzeit Ethernet Restbussimulation

Anwendung & Ergebnisse

Zusammenfassung & Ausblick

- Stetige Zunahme von Elektronik hat einen großen Einfluss auf die Kosten
 - 1985 ca. 3% Heute über 25%
 - Funktionen werden heute in Software realisiert
 - Frühzeitiges Testen wichtig für die Kontrolle der Entwicklungskosten
 - Dezentrale Entwicklung erschwert das Frühzeitige Testen
- Restbussimulation ermöglicht frühzeitiges Testen

Motivation & Einleitung

Frühzeitiges Testen von Elektronik

- Stetige Zunahme von Elektronik hat einen großen Einfluss auf die Kosten
 - 1985 ca. 3% Heute über 25%
 - Funktionen werden heute in Software realisiert
 - Frühzeitiges Testen wichtig für die Kontrolle der Entwicklungskosten
 - Dezentrale Entwicklung erschwert das Frühzeitige Testen
- Restbussimulation ermöglicht frühzeitiges Testen

Motivation & Einleitung

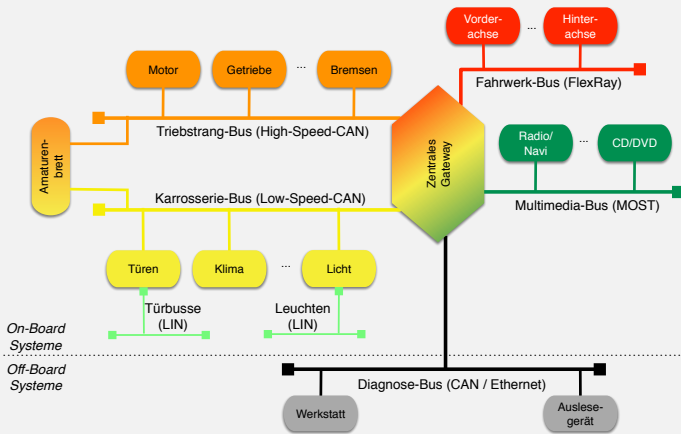
Kollaps im Bordnetzwerk



1. Restbussimulation ermöglicht das frühzeitige Testen
2. In dem fehlende Teile des Systems simuliert werden
3. Das System wird ein vollständiges System vorgetäuscht

Motivation & Einleitung

Kollaps im Bordnetzwerk



- Die Gesamtkomplexität des Bordnetzwerks ist schwer beherrschbar



RT-Ethernet
Restbussimulation

F. Bartols

Motivation &
Einleitung

Hintergrund

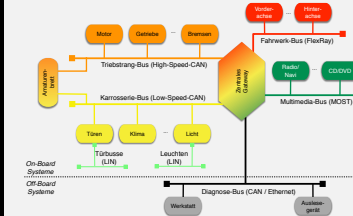
Echtzeit Ethernet
Restbussimulation

Anwendung &
Ergebnisse

Zusammenfassung &
Ausblick

Motivation & Einleitung

Kollaps im Bordnetzwerk



- Die Gesamtkomplexität des Bordnetzwerks ist schwer beherrschbar

Motivation & Einleitung

Kollaps im Bordnetzwerk



- RT-Ethernet Bordnetzwerk kann Probleme heutiger Netzwerke lösen

1. Die stetige Zunahme an Elektronik hat allerdings auch Auswirkungen auf das Bordnetzwerk im Automobil
2. Heutige Bordnetzwerke sind heterogene Strukturen mit verschiedenen Netzwerksystemen dessen Gesamtkomplexität nur noch schwer beherrschbar ist
3. Aktuell dient FlexRay und CAN für echtzeit Kommunikation im Antriebsstrang
4. CAN und LIN für kostengünstige Kommunikation in der Karosserie
5. MOST für Multimedia dienste Verwendet, die keine echtzeit Anforderungen besitzen
6. Zentrales Gateway für Interdomänenkommunikation

Motivation & Einleitung

Kollaps im Bordnetzwerk



RT-Ethernet
Restbussimulation

F. Bartols

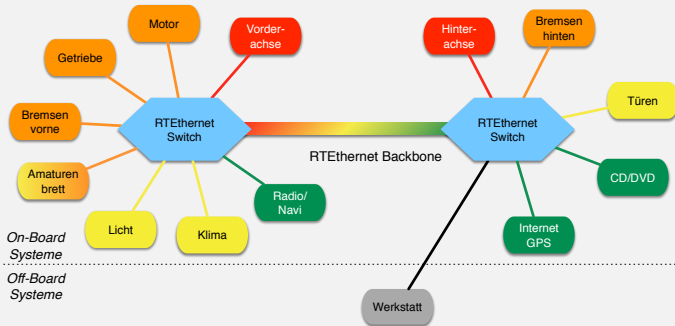
Motivation &
Einleitung

Hintergrund

Echtzeit Ethernet
Restbussimulation

Anwendung &
Ergebnisse

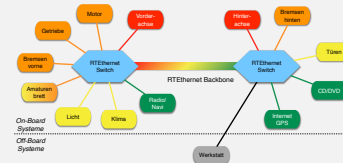
Zusammenfassung &
Ausblick



- RT-Ethernet Bordnetzwerk kann Probleme heutiger Netzwerke lösen

Motivation & Einleitung

Kollaps im Bordnetzwerk



- RT-Ethernet Bordnetzwerk kann Probleme heutiger Netzwerke lösen

Agenda

- 1 Motivation & Einleitung
- 2 Hintergrund RT-Ethernet & Restbussimulation
- 3 Echtzeit Ethernet Restbussimulation
- 4 Anwendung & Ergebnisse
- 5 Zusammenfassung & Ausblick

1. Die Lösung der Komplexität: ein RT-Ethernet basiertes Netzwerk
2. Alle Komponenten befinden sich im gleichen Netz
3. Die Interdomänenkommunikation direkt möglich
4. Dabei stellt hohe Bandbreiten zur Verfügung und ist gleichzeitig echtzeitfähig

1 Motivation & Einleitung

2 Hintergrund RT-Ethernet & Restbussimulation

3 Echtzeit Ethernet Restbussimulation

4 Anwendung & Ergebnisse

5 Zusammenfassung & Ausblick

Agenda

1 Motivation & Einleitung

2 Hintergrund RT-Ethernet & Restbussimulation

3 Echtzeit Ethernet Restbussimulation

4 Anwendung & Ergebnisse

5 Zusammenfassung & Ausblick

Agenda

1 Motivation & Einleitung

2 Hintergrund RT-Ethernet & Restbussimulation

3 Echtzeit Ethernet Restbussimulation

4 Anwendung & Ergebnisse

5 Zusammenfassung & Ausblick

RT-Ethernet Restbussimulation

F. Bartols

Motivation &
Einleitung

Hintergrund
Modellbasierte
Entwicklung im
Automobil
Restbussimulation als
Testmethodik
Echtzeit Ethernet im
Automobil

Echtzeit Ethernet
Restbussimulation

Anwendung &
Ergebnisse

Zusammenfassung &
Ausblick

1 Motivation & Einleitung

2 Hintergrund RT-Ethernet & Restbussimulation

3 Echtzeit Ethernet Restbussimulation

4 Anwendung & Ergebnisse

5 Zusammenfassung & Ausblick

Agenda

1 Motivation & Einleitung

2 Hintergrund RT-Ethernet & Restbussimulation

3 Echtzeit Ethernet Restbussimulation

4 Anwendung & Ergebnisse

5 Zusammenfassung & Ausblick

Modellbasierte Entwicklung

- Unterstützt die Entwicklung von komplexen Systemen
- Systeme werden schrittweise im Top-Down Verfahren entwickelt
- Frühzeitiges Analysieren von Systemen um Fehler zu erkennen
- Modelle als Spezifikation in der Systementwicklung
- Ermöglicht eine gemeinsame Sprache im verteilten Entwicklungsprozess

- Unterstützt die Entwicklung von komplexen Systemen
- Systeme werden schrittweise im Top-Down Verfahren entwickelt
- Frühzeitiges Analysieren von Systemen um Fehler zu erkennen
- Modelle als Spezifikation in der Systementwicklung
- Ermöglicht eine gemeinsame Sprache im verteilten Entwicklungsprozess

Modellbasierte Entwicklung

- Unterstützt die Entwicklung von komplexen Systemen
- Systeme werden schrittweise im Top-Down Verfahren entwickelt
- Frühzeitiges Analysieren von Systemen um Fehler zu erkennen
- Modelle als Spezifikation in der Systementwicklung
- Ermöglicht eine gemeinsame Sprache im verteilten Entwicklungsprozess

Modellbasiertes Testen

- Einsatz hauptsächlich als Blackbox-Verfahren
- Modelle als Spezifikation des System-under-Test (SUT)
- Systematische Ableitung der Testfälle anhand von Modellen
- Wiederverwendung der Testfälle innerhalb verschiedener Reifegrade des Systems
- Ausführung der Testfälle auf verschiedenen Test-Plattformen
 - MIL, SIL, PIL, HiL, Restbussimulation

1. Modelbasierte Entwicklung unterstützt den Entwicklungsprozess von komplexen verteilten Systemen
2. Die Systeme werden in Modellen Schrittweise im Top-Down Verfahren entwickelt. Und beginnt mit einer abstrakten Modellierung des Systems und wird Stufe für Stufe konkreter
3. Die entwickelten Modelle basieren in der Regel auf formalen Hintergrund, sodass Analysen oder Simulationen auf und mit den Modellen durchführbar sind
4. Sie dienen weiterhin als Spezifikation und Dokumentation der Systeme gegen die das System entwickelt wird
5. Können als gemeinsame Sprache im verteilten Entwicklungsprozess

- Einsatz hauptsächlich als Blackbox-Verfahren
- Modelle als Spezifikation des System-under-Test (SUT)
- Systematische Ableitung der Testfälle anhand von Modellen
- Wiederverwendung der Testfälle innerhalb verschiedener Reifegrade des Systems
- Ausführung der Testfälle auf verschiedenen Test-Plattformen
 - MiL, SiL, PiL, HiL, Restbussimulation

Modellbasiertes Testen

- Einsatz hauptsächlich als Blackbox-Verfahren
- Modelle als Spezifikation des System-under-Test (SUT)
- Systematische Ableitung der Testfälle anhand von Modellen
- Wiederverwendung der Testfälle innerhalb verschiedener Reifegrade des Systems
- Ausführung der Testfälle auf verschiedenen Test-Plattformen
 - MiL, SiL, PiL, HiL, Restbussimulation

Restbussimulation als Testplattform

- Aus der Sicht des SUT wird der „Rest des Busses (Netzwerks)“ simuliert
- Verbindung von SUT und Simulator direkt über die Kommunikationsschnittstelle
- Eine Restbussimulation muss transparent für das SUT erscheinen
 - Verhalten und Übertragung der Nachrichten müssen dem realen Netzwerk entsprechen
 - Netzwerkspezifische Attribute müssen ausgeführt werden
- Analyse des Tests findet auf Datenebene statt
- Offline nachdem ein Testfall ausgeführt wurde

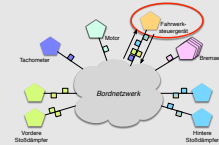
1. Modellbasiertes Testen wird hauptsächlich als Blackboxtestverfahren angewendet
2. Modelle werden als Spezifikation des Testobjetes verwendet
3. Testfälle werden systematischer erstellt. Die Erstellung hängt nicht von der Intuition des Testers ab
4. Es ermöglicht die Wiederverwendung der Testfälle innerhalb verschiedener Reifegrade
5. Die erzeugten und abgeleiteten Testfälle werden auf verschiedenen Testplattformen Ausgeführt
6. Zum Beispiel Model-in-the-Loop, Software-in-the-Loop, Processor-in-the-Loop, Hardware-in-the-Loop /

- Aus der Sicht des SUT wird der „Rest des Busses (Netzwerks)“ simuliert
- Verbindung von SUT und Simulator direkt über die Kommunikationsschnittstelle
- Eine Restbussimulation muss transparent für das SUT erscheinen
 - Verhalten und Übertragung der Nachrichten müssen dem realen Netzwerk entsprechen
 - Netzwerkspezifische Attribute müssen ausgeführt werden
- Analyse des Tests findet auf Datenebene statt
- Offline nachdem ein Testfall ausgeführt wurde

Restbussimulation als Testplattform

- Aus der Sicht des SUT wird der „Rest des Busses (Netzwerks)“ simuliert
- Verbindung von SUT und Simulator direkt über die Kommunikationsschnittstelle
- Eine Restbussimulation muss transparent für das SUT erscheinen
 - Verhalten und Übertragung der Nachrichten müssen dem realen Netzwerk entsprechen
 - Netzwerkspezifische Attribute müssen ausgeführt werden
- Analyse des Tests findet auf Datenebene statt
- Offline nachdem ein Testfall ausgeführt wurde

Beispiel einer Restbussimulation



1. Ermöglicht das Testen von verteilten Systemen

Beispiel einer Restbussimulation



RT-Ethernet
Restbussimulation

F. Bartols

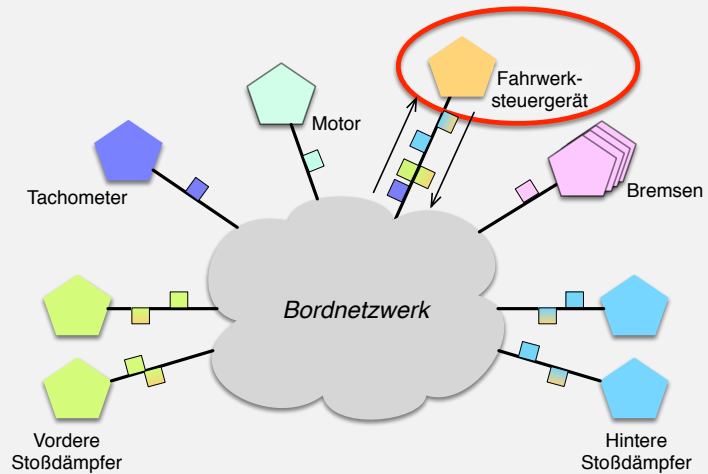
Motivation &
Einleitung

Hintergrund
Modellbasierte
Entwicklung im
Automobil
**Restbussimulation als
Testmethodik**
Echtzeit Ethernet im
Automobil

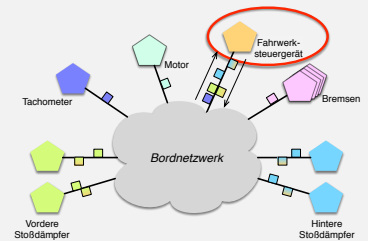
Echtzeit Ethernet
Restbussimulation

Anwendung &
Ergebnisse

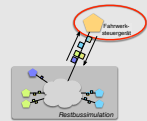
Zusammenfassung &
Ausblick



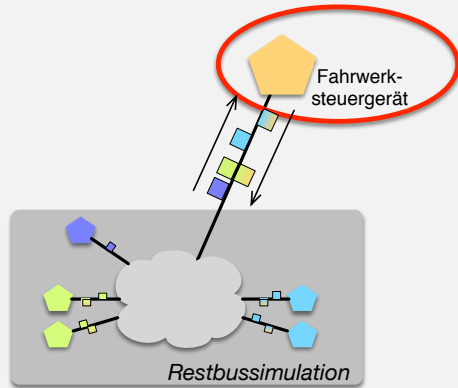
Beispiel einer Restbussimulation



Beispiel einer Restbussimulation



Beispiel einer Restbussimulation



RT-Ethernet
Restbussimulation

F. Bartols

Motivation &
Einleitung

Hintergrund

Modellbasierte
Entwicklung im
Automobil

**Restbussimulation als
Testmethodik**

Echtzeit Ethernet im
Automobil

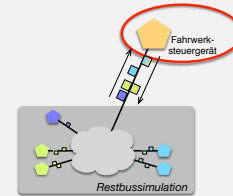
Echtzeit Ethernet
Restbussimulation

Anwendung &
Ergebnisse

Zusammenfassung &
Ausblick

10 / 29

Beispiel einer Restbussimulation



Eigenschaften von Time-Triggered
Ethernet

- 3 Nachrichtenklassen mit unterschiedlichen Attributen
- Statisches Routing für deterministisches Verhalten
- Zeitsynchronisierung für globale Zeit

Eigenschaften von Time-Triggered Ethernet

- 3 Nachrichtenklassen mit unterschiedlichen Attributen
- Statisches Routing für deterministisches Verhalten
- Zeitsynchronisierung für globale Zeit



RT-Ethernet
Restbussimulation

F. Bartols

Motivation &
Einleitung

Hintergrund
Modellbasierte
Entwicklung im
Automobil
Restbussimulation als
Testmethodik
**Echtzeit Ethernet im
Automobil**

Echtzeit Ethernet
Restbussimulation

Anwendung &
Ergebnisse

Zusammenfassung &
Ausblick

11 / 29

Eigenschaften von Time-Triggered Ethernet

- 3 Nachrichtenklassen mit unterschiedlichen Attributen
- Statisches Routing für deterministisches Verhalten
- Zeitsynchronisierung für globale Zeit

Eigenschaften von Time-Triggered Ethernet

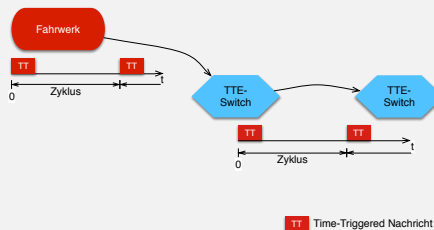
- 3 Nachrichtenklassen mit unterschiedlichen Attributen
- Statisches Routing für deterministisches Verhalten
- Zeitsynchronisierung für globale Zeit



1. TTEthernet ist ein Echtzeit-Ethernet Protokoll und ist von TTTech auf der Basis von Forschungsergebnisse der TU Wien entwickelt worden
2. Es unterstützt 3 Nachrichtenklassen die jeweils unterschiedliche zeitliche Attribute aufweisen
3. Der Nachrichtenfluss wird in TTEthernet durch statisches Routing modelliert, und sorgt für deterministisches Verhalten
4. Damit eine zeitgesteuerte Übertragung auf allen Geräten möglich ist, sorgt ein Zeitsynchronisierungsprotokoll für eine globale Zeit

Eigenschaften von Time-Triggered Ethernet

- 3 Nachrichtenklassen mit unterschiedlichen Attributen
- Statisches Routing für deterministisches Verhalten
- Zeitsynchronisierung für globale Zeit



RT-Ethernet
Restbussimulation

F. Bartols

Motivation &
Einleitung

Hintergrund
Modellbasierte
Entwicklung im
Automobil
Restbussimulation als
Testmethodik
Echtzeit Ethernet im
Automobil

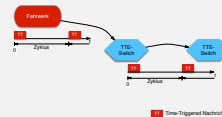
Echtzeit Ethernet
Restbussimulation

Anwendung &
Ergebnisse

Zusammenfassung &
Ausblick

Eigenschaften von Time-Triggered Ethernet

- 3 Nachrichtenklassen mit unterschiedlichen Attributen
- Statisches Routing für deterministisches Verhalten
- Zeitsynchronisierung für globale Zeit



Eigenschaften von Time-Triggered Ethernet

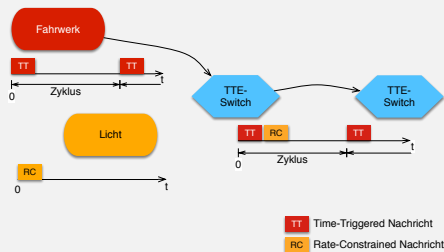
- 3 Nachrichtenklassen mit unterschiedlichen Attributen
- Statisches Routing für deterministisches Verhalten
- Zeitsynchronisierung für globale Zeit



1. In TTEthernet werden TT-Nachrichten für periodische Echtzeitdaten verwendet und haben die höchste Priorität. Sie werden gemäß eines festen Schedules übertragen. Der Schedule wird auf allen Endsystemen ausgeführt

Eigenschaften von Time-Triggered Ethernet

- 3 Nachrichtenklassen mit unterschiedlichen Attributen
- Statisches Routing für deterministisches Verhalten
- Zeitsynchronisierung für globale Zeit



RT-Ethernet
Restbussimulation
F. Bartols

Motivation &
Einleitung

Hintergrund
Modellbasierte
Entwicklung im
Automobil
Restbussimulation als
Testmethodik
Echtzeit Ethernet im
Automobil

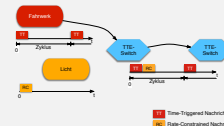
Echtzeit Ethernet
Restbussimulation

Anwendung &
Ergebnisse

Zusammenfassung &
Ausblick

Eigenschaften von Time-Triggered Ethernet

- 3 Nachrichtenklassen mit unterschiedlichen Attributen
- Statisches Routing für deterministisches Verhalten
- Zeitsynchronisierung für globale Zeit



Eigenschaften von Time-Triggered Ethernet

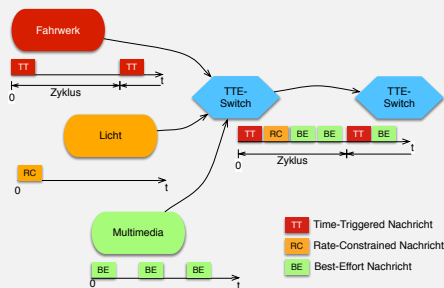
- 3 Nachrichtenklassen mit unterschiedlichen Attributen
- Statisches Routing für deterministisches Verhalten
- Zeitsynchronisierung für globale Zeit



1. RC-Nachrichten werden ebenfalls für Echtzeitdaten mit einer Ereignisgesteuerten Übertragung verwendet und werden mit der zweithöchsten Priorität übertragen. Die Bandbreite dieser Nachrichten ist limitiert und wird durch eine Bandwidth allocation Gap realisiert, die eine Lücke zwischen 2 Nachrichten darstellt

Eigenschaften von Time-Triggered Ethernet

- 3 Nachrichtenklassen mit unterschiedlichen Attributen
- Statisches Routing für deterministisches Verhalten
- Zeitsynchronisierung für globale Zeit



RT-Ethernet
Restbussimulation
F. Bartols

Motivation &
Einleitung

Hintergrund
Modellbasierte
Entwicklung im
Automobil
Restbussimulation als
Testmethodik
Echtzeit Ethernet im
Automobil

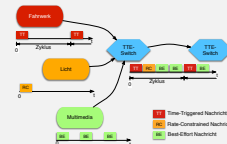
Echtzeit Ethernet
Restbussimulation

Anwendung &
Ergebnisse

Zusammenfassung &
Ausblick

Eigenschaften von Time-Triggered Ethernet

- 3 Nachrichtenklassen mit unterschiedlichen Attributen
- Statisches Routing für deterministisches Verhalten
- Zeitsynchronisierung für globale Zeit



Agenda

- Motivation & Einleitung
- Hintergrund RT-Ethernet & Restbussimulation
- Echtzeit Ethernet Restbussimulation
- Anwendung & Ergebnisse
- Zusammenfassung & Ausblick

1. BE-Nachrichten werden für unkritische Daten verwendet und haben keine Garantie in der Übertragung. Sie entsprechen standard Ethernet Verkehr.

1 Motivation & Einleitung

2 Hintergrund RT-Ethernet & Restbussimulation

3 Echtzeit Ethernet Restbussimulation

4 Anwendung & Ergebnisse

5 Zusammenfassung & Ausblick

RT-Ethernet Restbussimulation

F. Bartols

Motivation &
Einleitung

Hintergrund

Echtzeit Ethernet
Restbussimulation

Anwendung einer RBS
in RT-Ethernet
Modellierung von
Systemspezifikationen
Modellierung abstrakter
Testfälle
Verwendung von
Testfällen zur
Verhaltensmodellierung
Umsetzung einer
Echtzeit Ethernet Rest-
bussimulationsumgebung

Anwendung &
Ergebnisse

Zusammenfassung &
Ausblick

12 / 29

Agenda

1 Motivation & Einleitung

2 Hintergrund RT-Ethernet & Restbussimulation

3 Echtzeit Ethernet Restbussimulation

4 Anwendung & Ergebnisse

5 Zusammenfassung & Ausblick

RT-Ethernet RBS

RBS bisher

- Verwendung in busbasierten Systemen
- Verifikation funktionaler Anforderungen in diesen Systemen

RBS für RT-Ethernet

- Überprüfung von Zeitanforderungen der Nachrichtenübertragung
- Senderate, Reaktionsgeschwindigkeit und dessen Varianzen müssen getestet werden
- Implementierungsfehler führen zu Fehlfunktionen des Netzwerks

RBS bisher

- Verwendung in busbasierten Systemen
- Verifikation funktionaler Anforderungen in diesen Systemen

RBS für RT-Ethernet

- Überprüfung von Zeitanforderungen der Nachrichtenübertragung
- Senderate, Reaktionsgeschwindigkeit und dessen Varianzen müssen getestet werden
- Implementierungsfehler führen zu Fehlfunktionen des Netzwerks

RT-Ethernet Restbussimulation

F. Bartols

Motivation &
Einleitung

Hintergrund

Echtzeit Ethernet
Restbussimulation

**Anwendung einer RBS
in RT-Ethernet**
Modellierung von
Systemspezifikationen
Modellierung abstrakter
Testfälle
Verwendung von
Testfällen zur
Verhaltensmodellierung
Umsetzung einer
Echtzeit Ethernet Rest-
bussimulationsumgebung

Anwendung &
Ergebnisse

Zusammenfassung &
Ausblick

RT-Ethernet RBS

RBS bisher

- Verwendung in busbasierten Systemen
- Verifikation funktionaler Anforderungen in diesen Systemen

RBS für RT-Ethernet

- Überprüfung von Zeitanforderungen der Nachrichtenübertragung
- Senderate, Reaktionsgeschwindigkeit und dessen Varianzen müssen getestet werden
- Implementierungsfehler führen zu Fehlfunktionen des Netzwerks

RT-Ethernet RBS

RBS bisher

- Verwendung in busbasierten Systemen
- Verifikation funktionaler Anforderungen in diesen Systemen

RBS für RT-Ethernet

- Überprüfung von Zeitanforderungen der Nachrichtenübertragung
- Senderate, Reaktionsgeschwindigkeit und dessen Varianzen müssen getestet werden
- Implementierungsfehler führen zu Fehlfunktionen des Netzwerks

1. RBS bisher nur für Busbasierte Systeme, wie CAN, FlexRay, Lin und MOST verwendet
2. Produktiv eingesetzte Systeme erlauben auch neuerdings auch standard Ethernet
3. Einsatz der RBS hauptsächlich für funktionale Anforderungen

RBS bisher

- Verwendung in busbasierten Systemen
- Verifikation funktionaler Anforderungen in diesen Systemen

RBS für RT-Ethernet

- Überprüfung von Zeitanforderungen der Nachrichtenübertragung
- Senderate, Reaktionsgeschwindigkeit und dessen Varianzen müssen getestet werden
- Implementierungsfehler führen zu Fehlfunktionen des Netzwerks

RT-Ethernet Restbussimulation

F. Bartols

Motivation &
Einleitung

Hintergrund

Echtzeit Ethernet
Restbussimulation

**Anwendung einer RBS
in RT-Ethernet**
Modellierung von
Systemspezifikationen
Modellierung abstrakter
Testfälle
Verwendung von
Testfällen zur
Verhaltensmodellierung
Umsetzung einer
Echtzeit Ethernet Rest-
bussimulationsumgebung

Anwendung &
Ergebnisse

Zusammenfassung &
Ausblick

13 / 29

RT-Ethernet RBS

RBS bisher

- Verwendung in busbasierten Systemen
- Verifikation funktionaler Anforderungen in diesen Systemen

RBS für RT-Ethernet

- Überprüfung von Zeitanforderungen der Nachrichtenübertragung
- Senderate, Reaktionsgeschwindigkeit und dessen Varianzen müssen getestet werden
- Implementierungsfehler führen zu Fehlfunktionen des Netzwerks

Spezifikationsmodellierung

- Anforderungen müssen definiert werden um Systeme zu testen
- Mehrere Möglichkeiten Anforderungen zu Beschreiben
 - Natürliche Sprache
 - Modellierung in Modellen
- Modelle enthalten die wesentlichen Attribute des Systems
- Zwei Arten lassen sich modellieren
 - Funktionalen Anforderungen
 - Nicht-funktionale Anforderungen

1. Zusätzlich zu funktionalen Anforderungen spielen in RT-Ethernet auch Zeitanforderungen bei der Übertragung
2. Es muss möglich sein die Übertragungsstrategie zu überprüfen
3. Senderate, wie oft wir eine bestimmte Nachricht versendet
4. Reaktionsgeschwindigkeit, wie schnell reagiert das Gerät auf den Erhalt von Nachrichten
5. Implementierungsfehler können zu Fehlerhaften Verhalten in der Nachrichtenübertragung führen sodass keine Kommunikations stattfinden kann

- Anforderungen müssen definiert werden um Systeme zu testen
- Mehrere Möglichkeiten Anforderungen zu Beschreiben
 - Natürliche Sprache
 - Modellierung in Modellen
- Modelle enthalten die wesentlichen Attribute des Systems
- Zwei Arten lassen sich modellieren
 - Funktionalen Anforderungen
 - Nicht-funktionale Anforderungen

Spezifikationsmodellierung

- Anforderungen müssen definiert werden um Systeme zu testen
- Mehrere Möglichkeiten Anforderungen zu Beschreiben
 - Natürliche Sprache
 - Modellierung in Modellen
- Modelle enthalten die wesentlichen Attribute des Systems
- Zwei Arten lassen sich modellieren
 - Funktionalen Anforderungen
 - Nicht-funktionale Anforderungen

Spezifikationsmodellierung

- Anforderungen müssen definiert werden um Systeme zu testen
- Mehrere Möglichkeiten Anforderungen zu Beschreiben
 - Natürliche Sprache
 - Modellierung in Modellen
- Modelle enthalten die wesentlichen Attribute des Systems
- Zwei Arten lassen sich modellieren
 - Funktionalen Anforderungen
 - Nicht-funktionale Anforderungen

1. Damit überhaupt etwas getestet werden kann müssen Anforderungen definiert werden
2. Bei der Erstellung von Anforderungen gibt es mehrere Möglichkeiten
3. In der Regel werden Anforderungen zu erst in einer natürlichen Sprache beschrieben
4. Anschließend findet eine Überführung in Modellen statt

- Anforderungen müssen definiert werden um Systeme zu testen
- Mehrere Möglichkeiten Anforderungen zu Beschreiben
 - Natürliche Sprache
 - Modellierung in Modellen
- Modelle enthalten die wesentlichen Attribute des Systems
- Zwei Arten lassen sich modellieren
 - Funktionalen Anforderungen
 - Nicht-funktionale Anforderungen

RT-Ethernet Restbussimulation

F. Bartols

Motivation &
Einleitung

Hintergrund

Echtzeit Ethernet
Restbussimulation

Anwendung einer RBS
in RT-Ethernet

**Modellierung von
Systemspezifikationen**
Modellierung abstrakter
Testfälle

Verwendung von
Testfällen zur
Verhaltensmodellierung
Umsetzung einer
Echtzeit Ethernet Rest-
bussimulationsumgebung

Anwendung &
Ergebnisse

Zusammenfassung &
Ausblick

14 / 29

Spezifikationsmodellierung

- Anforderungen müssen definiert werden um Systeme zu testen
- Mehrere Möglichkeiten Anforderungen zu Beschreiben
 - Natürliche Sprache
 - Modellierung in Modellen
- Modelle enthalten die wesentlichen Attribute des Systems
- Zwei Arten lassen sich modellieren
 - Funktionalen Anforderungen
 - Nicht-funktionale Anforderungen

Spezifikationsmodellierung

- Modeling and Analysis of Real-time Embedded Systems (MARTE)
- UML nicht geeignet für eingebettete echtzeit Systeme

1. Modelle dienen als Spezifikation in denen Anforderungen modelliert sind
2. Funktionale Anforderungen: Welche Funktion soll das System umsetzen
3. Nicht-Funktionale Anforderungen: Wie die Funktion umzusetzen ist

- *Modeling and Analysis of Real-time Embedded Systems (MARTE)*
- UML nicht geeignet für eingebettete echtzeit Systeme



RT-Ethernet
Restbussimulation

F. Bartols

Motivation &
Einleitung

Hintergrund

Echtzeit Ethernet
Restbussimulation

Anwendung einer RBS
in RT-Ethernet

**Modellierung von
Systemspezifikationen**
Modellierung abstrakter
Testfälle
Verwendung von
Testfällen zur
Verhaltensmodellierung
Umsetzung einer
Echtzeit Ethernet Rest-
bussimulationsumgebung

Anwendung &
Ergebnisse

Zusammenfassung &
Ausblick

15 / 29

Spezifikationsmodellierung

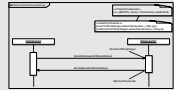
UML-MARTE

- *Modeling and Analysis of Real-time Embedded Systems (MARTE)*
- UML nicht geeignet für eingebettete echtzeit Systeme

Spezifikationsmodellierung

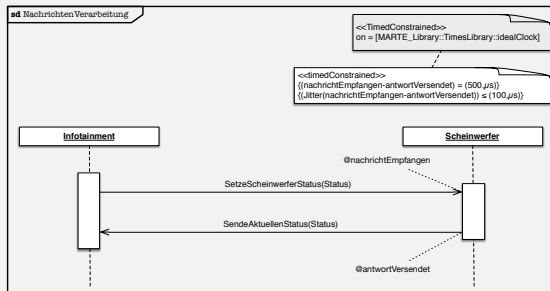
UML-MARTE

- *Modeling and Analysis of Real-time Embedded Systems (MARTE)*
- UML nicht geeignet für eingebettete echtzeit Systeme

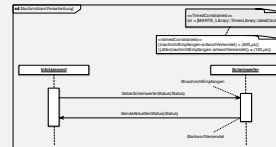


1. In dieser Arbeit wurde das UML-Profil Modeling and Analysis of Real-time Embedded Systems verwendet
2. Da UML kann nicht zur Modellierung von nicht-funktionalen Anforderungen für eingebettete Systeme verwendet werden

- *Modeling and Analysis of Real-time Embedded Systems (MARTE)*
- UML nicht geeignet für eingebettete echtzeit Systeme



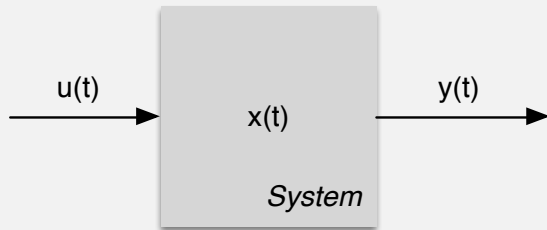
- *Modeling and Analysis of Real-time Embedded Systems (MARTE)*
- UML nicht geeignet für eingebettete echtzeit Systeme



- Testfälle müssen unabhängig der späteren Testplattform sein (Abstrakt)
- Abstraktes Testfallmodell basiert auf Zustandsraummodell
- Modellierung mehrerer Eingänge (U) und Ausgängen (Y) des SUT

1. MARTE für die Value Specific Language ein, die es erlaubt UML-Diagramme mit Annotationen zu versehen
2. Zeitliche Anforderungen werden auf diese Weise modelliert
3. Zeitpunkte werden innerhalb der Diagramme mit einem @ definiert

Logische Darstellung eines System-under-Test



- Testfälle müssen unabhängig der späteren Testplattform sein (Abstrakt)
- Abstraktes Testfallmodell basiert auf Zustandsraummodell
- Modellierung mehrerer Eingänge (U) und Ausgängen (Y) des SUT

RT-Ethernet
Restbussimulation

F. Bartols

Motivation &
Einleitung

Hintergrund

Echtzeit Ethernet
Restbussimulation

Anwendung einer RBS
in RT-Ethernet
Modellierung von
Systemspezifikationen
**Modellierung abstrakter
Testfälle**

Verwendung von
Testfällen zur
Verhaltensmodellierung
Umsetzung einer
Echtzeit Ethernet Rest-
bussimulationsumgebung

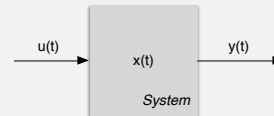
Anwendung &
Ergebnisse

Zusammenfassung &
Ausblick

16 / 29

Abstraktes Testfallmodell

Logische Darstellung eines System-under-Test



- Testfälle müssen unabhängig der späteren Testplattform sein (Abstrakt)
- Abstraktes Testfallmodell basiert auf Zustandsraummodell
- Modellierung mehrerer Eingänge (U) und Ausgängen (Y) des SUT

Abstraktes Testfallmodell

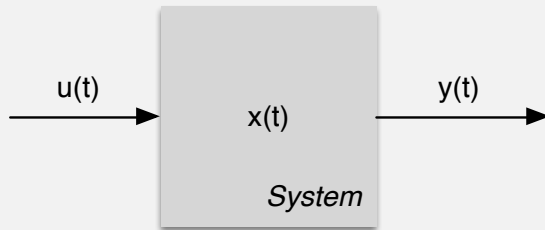
Logische Darstellung eines System-under-Test



- Testfälle müssen unabhängig der späteren Testplattform sein (Abstrakt)
- Abstraktes Testfallmodell basiert auf Zustandsraummodell
- Modellierung mehrerer Eingänge (U) und Ausgängen (Y) des SUT

1. Durch die Modellierung von Spezifikationen können anschließend Testfälle erstellt werden
2. Damit Modell für Testfälle entwickelt werden kann muss der logische Aufbau des SUT bekannt sein
3. Das System-under-Test besteht aus einer Menge von Eingängen (U)
4. Und aus einer Menge von Ausgängen (Y)
5. Die Menge der Ein- und Ausgänge definiert bei einer Restbussimulation die Anzahl der empfangenen und gesendeten Nachrichten

Logische Darstellung eines System-under-Test



- Testfälle müssen unabhängig der späteren Testplattform sein (Abstrakt)
- Abstraktes Testfallmodell basiert auf Zustandsraummodell
- Modellierung mehrerer Eingänge (U) und Ausgängen (Y) des SUT

RT-Ethernet
Restbussimulation

F. Bartols

Motivation &
Einleitung

Hintergrund

Echtzeit Ethernet
Restbussimulation

Anwendung einer RBS
in RT-Ethernet
Modellierung von
Systemspezifikationen
**Modellierung abstrakter
Testfälle**

Verwendung von
Testfällen zur
Verhaltensmodellierung
Umsetzung einer
Echtzeit Ethernet Rest-
bussimulationsumgebung

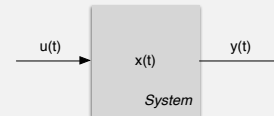
Anwendung &
Ergebnisse

Zusammenfassung &
Ausblick

16 / 29

Abstraktes Testfallmodell

Logische Darstellung eines System-under-Test



- Testfälle müssen unabhängig der späteren Testplattform sein (Abstrakt)
- Abstraktes Testfallmodell basiert auf Zustandsraummodell
- Modellierung mehrerer Eingänge (U) und Ausgängen (Y) des SUT

Abstraktes Testfallmodell Eigenschaften

Grundlagemodell für funktionale Anforderungen:

- $AS_{FA} = (T, U, Y_{out})$
- Modellierung konkreter Werte der Ein-, Ausgänge und Zuständen zu bestimmten Zeitpunkten

Testfallmodell für zeitliche Anforderungen einer RBS

- Senderate & Reaktionszeit muss modellierbar sein
- Erweiterung des Modells um diese Eigenschaften
- $AS_{FA} = (T, U, Y_{out}, I_{send}, R_{react}, P_{max})$

1. Im modellbasierten Testen ist es wichtig, dass die Testfälle auf verschiedenen Testplattformen ausgeführt werden können
2. Es ist deshalb wichtig, dass ein modellierter Testfall unabhängig von der späteren Ausführung ist und eine Abstraktion bereitstellt
3. Modell der abstrakten Testfälle basiert auf dem Zustandsraummodell
4. Zustandsraummodell wird in der Regelungstechnik u.A. eingesetzt um dynamische Systeme analysieren zu können
5. Das Zustandsraummodell erlaubt in diesem Fall das genaue modellieren von Ein und Ausgängen an einem System unter Test



RT-Ethernet Restbussimulation

F. Bartols

Motivation &
Einleitung

Hintergrund

Echtzeit Ethernet
Restbussimulation

Anwendung einer RBS
in RT-Ethernet
Modellierung von
Systemspezifikationen
**Modellierung abstrakter
Testfälle**

Verwendung von
Testfällen zur
Verhaltensmodellierung
Umsetzung einer
Echtzeit Ethernet Rest-
bussimulationsumgebung

Anwendung &
Ergebnisse

Zusammenfassung &
Ausblick

Grundlagemodell für funktionale Anforderungen:

- $AS_{FA} = (T, U, Y_{soll})$
- Modellierung konkreter Werte der Ein-, Ausgänge und Zuständen zu bestimmten Zeitpunkten

Testfallmodell für zeitliche Anforderungen einer RBS

- Senderate & Reaktionszeit muss modellierbar sein
- Erweiterung des Modells um diese Eigenschaften
- $AS_{LA} = (T, U, Y_{soll}, L_{soll}, R_{soll}, J_{Lsoll}, J_{Rsoll})$

Abstraktes Testfallmodell

Eigenschaften

Grundlagemodell für funktionale Anforderungen:

- $AS_{FA} = (T, U, Y_{soll})$
- Modellierung konkreter Werte der Ein-, Ausgänge und Zuständen zu bestimmten Zeitpunkten

Testfallmodell für zeitliche Anforderungen einer RBS

- Senderate & Reaktionszeit muss modellierbar sein
- Erweiterung des Modells um diese Eigenschaften
- $AS_{LA} = (T, U, Y_{soll}, L_{soll}, R_{soll}, J_{Lsoll}, J_{Rsoll})$

Abstraktes Testfallmodell

Eigenschaften

Grundlagemodell für funktionale Anforderungen:

- $AS_{FA} = (T, U, Y_{soll})$
- Modellierung konkreter Werte der Ein-, Ausgänge und Zuständen zu bestimmten Zeitpunkten

Testfallmodell für zeitliche Anforderungen einer RBS

- Senderate & Reaktionszeit muss modellierbar sein
- Erweiterung des Modells um diese Eigenschaften
- $AS_{LA} = (T, U, Y_{soll}, L_{soll}, R_{soll}, J_{Lsoll}, J_{Rsoll})$

1. Als Grundlage wurde ein abstraktes Testfälle verwendet, dass zur Modellierung von funktionale Anforderungen dient
2. Und wurde bereits bei einer CAN-Restbussimulation bereits angewendet
3. Ist ein 3-Tupel dass aus dem Zeitvektor T, sowie den Eingangsvektor U und den erwarteten Ausgangsvektor Y des System-under-Test beschreibt
4. Der Zeitvektor gibt dabei an, zu welchem Zeitpunkt der Eingang des SUT mit welchen Werten belegt wird
5. und welche Werte am Ausgang zu diesem Zeitpunkt erwartet werden



RT-Ethernet Restbussimulation

F. Bartols

Motivation &
Einleitung

Hintergrund

Echtzeit Ethernet
Restbussimulation

Anwendung einer RBS
in RT-Ethernet
Modellierung von
Systemspezifikationen
**Modellierung abstrakter
Testfälle**

Verwendung von
Testfällen zur
Verhaltensmodellierung
Umsetzung einer
Echtzeit Ethernet Rest-
bussimulationsumgebung

Anwendung &
Ergebnisse

Zusammenfassung &
Ausblick

Grundlagemodell für funktionale Anforderungen:

- $AS_{FA} = (T, U, Y_{soll})$
- Modellierung konkreter Werte der Ein-, Ausgänge und Zuständen zu bestimmten Zeitpunkten

Testfallmodell für zeitliche Anforderungen einer RBS

- Senderate & Reaktionszeit muss modellierbar sein
- Erweiterung des Modells um diese Eigenschaften
- $AS_{LA} = (T, U, Y_{soll}, L_{soll}, R_{soll}, J_{Lsoll}, J_{Rsoll})$

Abstraktes Testfallmodell

Eigenschaften

Grundlagemodell für funktionale Anforderungen:

- $AS_{FA} = (T, U, Y_{soll})$
- Modellierung konkreter Werte der Ein-, Ausgänge und Zuständen zu bestimmten Zeitpunkten

Testfallmodell für zeitliche Anforderungen einer RBS

- Senderate & Reaktionszeit muss modellierbar sein
- Erweiterung des Modells um diese Eigenschaften
- $AS_{LA} = (T, U, Y_{soll}, L_{soll}, R_{soll}, J_{Lsoll}, J_{Rsoll})$

Abstraktes Testfallmodell

Eigenschaften

Grundlagemodell für funktionale Anforderungen:

- $AS_{FA} = (T, U, Y_{soll})$
- Modellierung konkreter Werte der Ein-, Ausgänge und Zuständen zu bestimmten Zeitpunkten

Testfallmodell für zeitliche Anforderungen einer RBS

- Senderate & Reaktionszeit muss modellierbar sein
- Erweiterung des Modells um diese Eigenschaften
- $AS_{LA} = (T, U, Y_{soll}, L_{soll}, R_{soll}, J_{Lsoll}, J_{Rsoll})$

1. Damit die geforderte Senderate und die Reaktionszeit modelliert werden kann wurde das Modell mit diesen Attributen erweitert



RT-Ethernet
Restbussimulation

F. Bartols

Motivation &
Einleitung

Hintergrund

Echtzeit Ethernet
Restbussimulation

Anwendung einer RBS
in RT-Ethernet
Modellierung von
Systemspezifikationen
**Modellierung abstrakter
Testfälle**

Verwendung von
Testfällen zur
Verhaltensmodellierung
Umsetzung einer
Echtzeit Ethernet Rest-
bussimulationsumgebung

Anwendung &
Ergebnisse

Zusammenfassung &
Ausblick

Grundlagemodell für funktionale Anforderungen:

- $AS_{FA} = (T, U, Y_{soll})$
- Modellierung konkreter Werte der Ein-, Ausgänge und Zuständen zu bestimmten Zeitpunkten

Testfallmodell für zeitliche Anforderungen einer RBS

- Senderate & Reaktionszeit muss modellierbar sein
- Erweiterung des Modells um diese Eigenschaften
- $AS_{LA} = (T, U, Y_{soll}, L_{soll}, R_{soll}, J_{Lsoll}, J_{Rsoll})$

Abstraktes Testfallmodell

Eigenschaften

Grundlagemodell für funktionale Anforderungen:

- $AS_{FA} = (T, U, Y_{soll})$
- Modellierung konkreter Werte der Ein-, Ausgänge und Zuständen zu bestimmten Zeitpunkten

Testfallmodell für zeitliche Anforderungen einer RBS

- Senderate & Reaktionszeit muss modellierbar sein
- Erweiterung des Modells um diese Eigenschaften
- $AS_{LA} = (T, U, Y_{soll}, L_{soll}, R_{soll}, J_{Lsoll}, J_{Rsoll})$

Abstraktes Testfallmodell

Beispiel in Tabellenform

■ $AS_{LA} = (T, U, Y_{soll}, L_{soll}, R_{soll}, J_{Lsoll}, J_{Rsoll})$

T	1a	5a	9a
U	a1 = ALLES_AUS	a2 = BLINDEN_EIN	a3 = BLINDEN_AUS
Y_soll	y1 = ALLES_AUS	y2 = BLINDEN_EIN	y3 = ALLES_AUS
L_soll	L_soll = 100ms		
R_soll	R_soll = 100ms		
J_Lsoll	J_Lsoll = 100ms		
J_Rsoll	J_Rsoll = 100ms		

■ Darstellung von funktionalen und nicht-funktionalen Anforderungen im gleichen Testfall

1. Ein abstrakter Testfall zur Überprüfung von nicht-funktionalen zeitlichen Anforderungen wird als 7 Tupel definiert

Abstraktes Testfallmodell

Beispiel in Tabellenform



RT-Ethernet
Restbussimulation

F. Bartols

■ $AS_{LA} = (T, U, Y_{soll}, L_{soll}, R_{soll}, J_{Lsoll}, J_{Rsoll})$

T	1s	5s	9s
U	$u_1 = ALLES_AUS$	$u_1 = BLINKER_EIN$	$u_1 = BLINKER_AUS$
Y_{soll}	$y_1 = ALLES_AUS$	$y_1 = BLINKER_EIN$	$y_1 = ALLES_AUS$
L_{soll}	$h_1(u_1, y_1) = 500\mu s$		
R_{soll}	$r_1(y_1) = 5000\mu s$		
J_{Lsoll}	$j_{L1}(h_1) \leq 100\mu s$		
J_{Rsoll}	$j_{R1}(r_1) \leq 10\mu s$		

- Darstellung von funktionalen und nicht-funktionalen Anforderungen im gleichen Testfall

Motivation &
Einleitung

Hintergrund

Echtzeit Ethernet
Restbussimulation

Anwendung einer RBS
in RT-Ethernet
Modellierung von
Systemspezifikationen
**Modellierung abstrakter
Testfälle**
Verwendung von
Testfällen zur
Verhaltensmodellierung
Umsetzung einer
Echtzeit Ethernet Rest-
bussimulationsumgebung

Anwendung &
Ergebnisse

Zusammenfassung &
Ausblick

Abstraktes Testfallmodell

Beispiel in Tabellenform

■ $AS_{LA} = (T, U, Y_{soll}, L_{soll}, R_{soll}, J_{Lsoll}, J_{Rsoll})$

T	1s	5s	9s
U	$u_1 = ALLES_AUS$	$u_1 = BLINKER_EIN$	$u_1 = BLINKER_AUS$
Y_{soll}	$y_1 = ALLES_AUS$	$y_1 = BLINKER_EIN$	$y_1 = ALLES_AUS$
L_{soll}	$h_1(u_1, y_1) = 500\mu s$		
R_{soll}	$r_1(y_1) = 5000\mu s$		
J_{Lsoll}	$j_{L1}(h_1) \leq 100\mu s$		
J_{Rsoll}	$j_{R1}(r_1) \leq 10\mu s$		

- Darstellung von funktionalen und nicht-funktionalen Anforderungen im gleichen Testfall

Abstraktes Testfallmodell

Beispiel in Tabellenform

■ $AS_{LA} = (T, U, Y_{soll}, L_{soll}, R_{soll}, J_{Lsoll}, J_{Rsoll})$

T	1s	5s	9s
U	$u_1 = ALLES_AUS$	$u_1 = BLINKER_EIN$	$u_1 = BLINKER_AUS$
Y_{soll}	$y_1 = ALLES_AUS$	$y_1 = BLINKER_EIN$	$y_1 = ALLES_AUS$
L_{soll}	$h_1(u_1, y_1) = 500\mu s$		
R_{soll}	$r_1(y_1) = 5000\mu s$		
J_{Lsoll}	$j_{L1}(h_1) \leq 100\mu s$		
J_{Rsoll}	$j_{R1}(r_1) \leq 10\mu s$		

- Darstellung von funktionalen und nicht-funktionalen Anforderungen im gleichen Testfall

1. T ist der Zeitvektor der die Zeitpunkte definiert an denen die Werte am Eingang und Ausgang anliegen

Abstraktes Testfallmodell

Beispiel in Tabellenform



RT-Ethernet
Restbussimulation

F. Bartols

■ $AS_{LA} = (T, U, Y_{soll}, L_{soll}, R_{soll}, J_{Lsoll}, J_{Rsoll})$

T	1s	5s	9s
U	$u_1 = ALLES_AUS$	$u_1 = BLINKER_EIN$	$u_1 = BLINKER_AUS$
Y_{soll}	$y_1 = ALLES_AUS$	$y_1 = BLINKER_EIN$	$y_1 = ALLES_AUS$
L_{soll}	$h_1(u_1, y_1) = 500\mu s$		
R_{soll}	$r_1(y_1) = 5000\mu s$		
J_{Lsoll}	$j_{L1}(h_1) \leq 100\mu s$		
J_{Rsoll}	$j_{R1}(r_1) \leq 10\mu s$		

- Darstellung von funktionalen und nicht-funktionalen Anforderungen im gleichen Testfall

Motivation &
Einleitung

Hintergrund

Echtzeit Ethernet
Restbussimulation

Anwendung einer RBS
in RT-Ethernet
Modellierung von
Systemspezifikationen
**Modellierung abstrakter
Testfälle**

Verwendung von
Testfällen zur
Verhaltensmodellierung
Umsetzung einer
Echtzeit Ethernet Rest-
bussimulationsumgebung

Anwendung &
Ergebnisse

Zusammenfassung &
Ausblick

Abstraktes Testfallmodell

Beispiel in Tabellenform

■ $AS_{LA} = (T, U, Y_{soll}, L_{soll}, R_{soll}, J_{Lsoll}, J_{Rsoll})$

T	1s	5s	9s
U	$u_1 = ALLES_AUS$	$u_1 = BLINKER_EIN$	$u_1 = BLINKER_AUS$
Y_{soll}	$y_1 = ALLES_AUS$	$y_1 = BLINKER_EIN$	$y_1 = ALLES_AUS$
L_{soll}	$h_1(u_1, y_1) = 500\mu s$		
R_{soll}	$r_1(y_1) = 5000\mu s$		
J_{Lsoll}	$j_{L1}(h_1) \leq 100\mu s$		
J_{Rsoll}	$j_{R1}(r_1) \leq 10\mu s$		

- Darstellung von funktionalen und nicht-funktionalen Anforderungen im gleichen Testfall

Abstraktes Testfallmodell

Beispiel in Tabellenform

■ $AS_{LA} = (T, U, Y_{soll}, L_{soll}, R_{soll}, J_{Lsoll}, J_{Rsoll})$

T	1s	5s	9s
U	$u_1 = ALLES_AUS$	$u_1 = BLINKER_EIN$	$u_1 = BLINKER_AUS$
Y_{soll}	$y_1 = ALLES_AUS$	$y_1 = BLINKER_EIN$	$y_1 = ALLES_AUS$
L_{soll}	$h_1(u_1, y_1) = 500\mu s$		
R_{soll}	$r_1(y_1) = 5000\mu s$		
J_{Lsoll}	$j_{L1}(h_1) \leq 100\mu s$		
J_{Rsoll}	$j_{R1}(r_1) \leq 10\mu s$		

- Darstellung von funktionalen und nicht-funktionalen Anforderungen im gleichen Testfall

1. T ist der Zeitvektor der die Zeitpunkte definiert an denen die Werte am Eingang und Ausgang anliegen
2. L beschreibt die Reaktionszeit (Latenz) des SUTs auf einen Eingang

Abstraktes Testfallmodell

Beispiel in Tabellenform



RT-Ethernet
Restbussimulation

F. Bartols

■ $AS_{LA} = (T, U, Y_{soll}, L_{soll}, R_{soll}, J_{Lsoll}, J_{Rsoll})$

T	1s	5s	9s
U	$u_1 = ALLES_AUS$	$u_1 = BLINKER_EIN$	$u_1 = BLINKER_AUS$
Y_{soll}	$y_1 = ALLES_AUS$	$y_1 = BLINKER_EIN$	$y_1 = ALLES_AUS$
L_{soll}	$h_1(u_1, y_1) = 500\mu s$		
R_{soll}	$r_1(y_1) = 5000\mu s$		
J_{Lsoll}	$j_{L1}(h_1) \leq 100\mu s$		
J_{Rsoll}	$j_{R1}(r_1) \leq 10\mu s$		

- Darstellung von funktionalen und nicht-funktionalen Anforderungen im gleichen Testfall

Motivation &
Einleitung

Hintergrund

Echtzeit Ethernet
Restbussimulation

Anwendung einer RBS
in RT-Ethernet
Modellierung von
Systemspezifikationen
**Modellierung abstrakter
Testfälle**

Verwendung von
Testfällen zur
Verhaltensmodellierung
Umsetzung einer
Echtzeit Ethernet Rest-
bussimulationsumgebung

Anwendung &
Ergebnisse

Zusammenfassung &
Ausblick

Abstraktes Testfallmodell

Beispiel in Tabellenform

■ $AS_{LA} = (T, U, Y_{soll}, L_{soll}, R_{soll}, J_{Lsoll}, J_{Rsoll})$

T	1s	5s	9s
U	$u_1 = ALLES_AUS$	$u_1 = BLINKER_EIN$	$u_1 = BLINKER_AUS$
Y_{soll}	$y_1 = ALLES_AUS$	$y_1 = BLINKER_EIN$	$y_1 = ALLES_AUS$
L_{soll}	$h_1(u_1, y_1) = 500\mu s$		
R_{soll}	$r_1(y_1) = 5000\mu s$		
J_{Lsoll}	$j_{L1}(h_1) \leq 100\mu s$		
J_{Rsoll}	$j_{R1}(r_1) \leq 10\mu s$		

- Darstellung von funktionalen und nicht-funktionalen Anforderungen im gleichen Testfall

Abstraktes Testfallmodell

Beispiel in Tabellenform

■ $AS_{LA} = (T, U, Y_{soll}, L_{soll}, R_{soll}, J_{Lsoll}, J_{Rsoll})$

T	1s	5s	9s
U	$u_1 = ALLES_AUS$	$u_1 = BLINKER_EIN$	$u_1 = BLINKER_AUS$
Y_{soll}	$y_1 = ALLES_AUS$	$y_1 = BLINKER_EIN$	$y_1 = ALLES_AUS$
L_{soll}	$h_1(u_1, y_1) = 500\mu s$		
R_{soll}	$r_1(y_1) = 5000\mu s$		
J_{Lsoll}	$j_{L1}(h_1) \leq 100\mu s$		
J_{Rsoll}	$j_{R1}(r_1) \leq 10\mu s$		

- Darstellung von funktionalen und nicht-funktionalen Anforderungen im gleichen Testfall

1. T ist der Zeitvektor der die Zeitpunkte definiert an denen die Werte am Eingang und Ausgang anliegen
2. L beschreibt die Reaktionszeit (Latenz) des SUTs auf einen Eingang
3. R beschreibt die Senderate von Paketen an einem Ausgang

Abstraktes Testfallmodell

Beispiel in Tabellenform



RT-Ethernet
Restbussimulation

F. Bartols

■ $AS_{LA} = (T, U, Y_{soll}, L_{soll}, R_{soll}, J_{Lsoll}, J_{Rsoll})$

T	1s	5s	9s
U	$u_1 = \text{ALLES_AUS}$	$u_1 = \text{BLINKER_EIN}$	$u_1 = \text{BLINKER_AUS}$
Y_{soll}	$y_1 = \text{ALLES_AUS}$	$y_1 = \text{BLINKER_EIN}$	$y_1 = \text{ALLES_AUS}$
L_{soll}	$h_1(u_1, y_1) = 500\mu s$		
R_{soll}	$r_1(y_1) = 5000\mu s$		
J_{Lsoll}	$j_{L1}(h_1) \leq 100\mu s$		
J_{Rsoll}	$j_{R1}(r_1) \leq 10\mu s$		

■ Darstellung von funktionalen und nicht-funktionalen Anforderungen im gleichen Testfall

Motivation &
Einleitung

Hintergrund

Echtzeit Ethernet
Restbussimulation

Anwendung einer RBS
in RT-Ethernet
Modellierung von
Systemspezifikationen
**Modellierung abstrakter
Testfälle**

Verwendung von
Testfällen zur
Verhaltensmodellierung
Umsetzung einer
Echtzeit Ethernet Rest-
bussimulationsumgebung

Anwendung &
Ergebnisse

Zusammenfassung &
Ausblick

Abstraktes Testfallmodell

Beispiel in Tabellenform

■ $AS_{LA} = (T, U, Y_{soll}, L_{soll}, R_{soll}, J_{Lsoll}, J_{Rsoll})$

T	1s	5s	9s
U	$u_1 = \text{ALLES_AUS}$	$u_1 = \text{BLINKER_EIN}$	$u_1 = \text{BLINKER_AUS}$
Y_{soll}	$y_1 = \text{ALLES_AUS}$	$y_1 = \text{BLINKER_EIN}$	$y_1 = \text{ALLES_AUS}$
L_{soll}	$h_1(u_1, y_1) = 500\mu s$		
R_{soll}	$r_1(y_1) = 5000\mu s$		
J_{Lsoll}	$j_{L1}(h_1) \leq 100\mu s$		
J_{Rsoll}	$j_{R1}(r_1) \leq 10\mu s$		

■ Darstellung von funktionalen und nicht-funktionalen Anforderungen im gleichen Testfall

Abstraktes Testfallmodell

Beispiel in Tabellenform

■ $AS_{LA} = (T, U, Y_{soll}, L_{soll}, R_{soll}, J_{Lsoll}, J_{Rsoll})$

T	1s	5s	9s
U	$u_1 = \text{ALLES_AUS}$	$u_1 = \text{BLINKER_EIN}$	$u_1 = \text{BLINKER_AUS}$
Y_{soll}	$y_1 = \text{ALLES_AUS}$	$y_1 = \text{BLINKER_EIN}$	$y_1 = \text{ALLES_AUS}$
L_{soll}	$h_1(u_1, y_1) = 500\mu s$		
R_{soll}	$r_1(y_1) = 5000\mu s$		
J_{Lsoll}	$j_{L1}(h_1) \leq 100\mu s$		
J_{Rsoll}	$j_{R1}(r_1) \leq 10\mu s$		

■ Darstellung von funktionalen und nicht-funktionalen Anforderungen im gleichen Testfall

1. T ist der Zeitvektor der die Zeitpunkte definiert an denen die Werte am Eingang und Ausgang anliegen
2. L beschreibt die Reaktionszeit (Latenz) des SUTs auf einen Eingang
3. R beschreibt die Senderate von Paketen an einem Ausgang
4. J beschreibt die jeweiligen Jitter (Varianzen), in diesem Fall Varianzen der Reaktionszeit und Senderate

Abstraktes Testfallmodell

Beispiel in Tabellenform



RT-Ethernet
Restbussimulation

F. Bartols

Motivation &
Einleitung

Hintergrund

Echtzeit Ethernet
Restbussimulation

Anwendung einer RBS
in RT-Ethernet
Modellierung von
Systemspezifikationen
**Modellierung abstrakter
Testfälle**

Verwendung von
Testfällen zur
Verhaltensmodellierung
Umsetzung einer
Echtzeit Ethernet Rest-
bussimulationsumgebung

Anwendung &
Ergebnisse

Zusammenfassung &
Ausblick

■ $AS_{LA} = (T, U, Y_{soll}, L_{soll}, R_{soll}, J_{Lsoll}, J_{Rsoll})$

T	1s	5s	9s
U	$u_1 = ALLES_AUS$	$u_1 = BLINKER_EIN$	$u_1 = BLINKER_AUS$
Y_{soll}	$y_1 = ALLES_AUS$	$y_1 = BLINKER_EIN$	$y_1 = ALLES_AUS$
L_{soll}	$l_1(u_1, y_1) = 500\mu s$		
R_{soll}	$r_1(y_1) = 5000\mu s$		
J_{Lsoll}	$j_{L1}(l_1) \leq 100\mu s$		
J_{Rsoll}	$j_{R1}(r_1) \leq 10\mu s$		

- Darstellung von funktionalen und nicht-funktionalen Anforderungen im gleichen Testfall

Abstraktes Testfallmodell

Beispiel in Tabellenform

■ $AS_{LA} = (T, U, Y_{soll}, L_{soll}, R_{soll}, J_{Lsoll}, J_{Rsoll})$

T	1s	5s	9s
U	$u_1 = ALLES_AUS$	$u_1 = BLINKER_EIN$	$u_1 = BLINKER_AUS$
Y_{soll}	$y_1 = ALLES_AUS$	$y_1 = BLINKER_EIN$	$y_1 = ALLES_AUS$
L_{soll}	$l_1(u_1, y_1) = 500\mu s$		
R_{soll}	$r_1(y_1) = 5000\mu s$		
J_{Lsoll}	$j_{L1}(l_1) \leq 100\mu s$		
J_{Rsoll}	$j_{R1}(r_1) \leq 10\mu s$		

- Darstellung von funktionalen und nicht-funktionalen Anforderungen im gleichen Testfall

Verhaltensmodellierung einer RBS

- Verhalten der RBS wird durch das Senden von bestimmten Daten bestimmt
- Modell der abstrakten Testfälle ist geeignet zur Verhaltensmodellierung
- Die Eingänge des SUT können mit konkreten Werten belegt werden
- Zeitvektor gibt explizite Zeitpunkte vor an denen die Werten anliegen werden
- Das Verhalten einer Restbussimulation wird zu bestimmten Zeitpunkten bestimmt

1. T ist der Zeitvektor der die Zeitpunkte definiert an denen die Werte am Eingang und Ausgang anliegen
2. L beschreibt die Reaktionszeit (Latenz) des SUTs auf einen Eingang
3. R beschreibt die Senderate von Paketen an einem Ausgang
4. J beschreibt die jeweiligen Jitter (Varianzen), in diesem Fall Varianzen der Reaktionszeit und Senderate
5. Auf diese Weise lassen sich funktionale und nicht-funktionael Anforderungen im gleiche Testfall abbilden

- Verhalten der RBS wird durch das Senden von bestimmten Daten bestimmt
- Modell der abstrakten Testfälle ist geeignet zur Verhaltensmodellierung
- Die Eingänge des SUT können mit konkreten Werten belegt werden
- Zeitvektor gibt explizite Zeitpunkte vor an denen die Werten anliegen werden
- Das Verhalten einer Restbussimulation wird zu bestimmten Zeitpunkten bestimmt

Verhaltensmodellierung einer RBS

- Verhalten der RBS wird durch das Senden von bestimmten Daten bestimmt
- Modell der abstrakten Testfälle ist geeignet zur Verhaltensmodellierung
- Die Eingänge des SUT können mit konkreten Werten belegt werden
- Zeitvektor gibt explizite Zeitpunkte vor an denen die Werten anliegen werden
- Das Verhalten einer Restbussimulation wird zu bestimmten Zeitpunkten bestimmt

Implementierung des Restbussimulators

Anforderungen:

- Protokollkonforme Übertragung der Nachrichten
- Ausführung des abstrakten Testfallmodells
- Aufzeichnung des gesamten Datenverkehrs zu offline Auswertung des Testfalls

1. Die Modellierung von Testfällen lässt es zu funktionale und nicht-funktionale Anforderungen eines Systems zu testen
2. Damit eine Restbussimulation ausgeführt werden kann muss außerdem ein Modell vorhanden sein, dass das Verhalten bestimmt
3. Das Modell der abstrakten Testfälle ist gut zur Verhaltensmodellierung geeignet,
4. da die Eingänge mit bestimmten Werten belegt werden können
5. Der Zeitvektor erlaubt es dabei die Werte zu bestimmten Zeitpunkten zu belegen
6. Auf diese Weise lässt sich das Verhalten einer Restbussimulation genau bestimmen

Anforderungen:

- Protokollkonforme Übertragung der Nachrichten
- Ausführung des abstrakten Testfallmodells
- Aufzeichnung des gesamten Datenverkehrs zu offline Auswertung des Testfalls

RT-Ethernet
Restbussimulation

F. Bartols

Motivation &
Einleitung

Hintergrund

Echtzeit Ethernet
Restbussimulation

Anwendung einer RBS
in RT-Ethernet
Modellierung von
Systemspezifikationen
Modellierung abstrakter
Testfälle

Verwendung von
Testfällen zur
Verhaltensmodellierung

Umsetzung einer
Echtzeit Ethernet Rest-
bussimulationsumgebung

Anwendung &
Ergebnisse

Zusammenfassung &
Ausblick

20 / 29

Implementierung des Restbussimulators

Anforderungen:

- Protokollkonforme Übertragung der Nachrichten
- Ausführung des abstrakten Testfallmodells
- Aufzeichnung des gesamten Datenverkehrs zu offline Auswertung des Testfalls

Implementierung des Restbussimulators

Anforderungen:

- Protokollkonforme Übertragung der Nachrichten
- Ausführung des abstrakten Testfallmodells
- Aufzeichnung des gesamten Datenverkehrs zu offline Auswertung des Testfalls



1. Durch die Modellierung von Testfällen und das Verhalten einer Restbussimulation kann die Restbussimulation implementiert werden.
2. Die Restbussimulationsplattform muss eine Übertragung der Nachrichten erlauben
3. Sämtliche Nachrichtenklassen und Synchronisationsmechanismen muss die Hardwareplattform unterstützen
4. Das Modell der abstrakten Testfälle muss ausführbar sein
5. Damit eine Auswertung stattfinden kann, muss der Datenverkehr aufgezeichnet werden

Anforderungen:

- Protokollkonforme Übertragung der Nachrichten
- Ausführung des abstrakten Testfallmodells
- Aufzeichnung des gesamten Datenverkehrs zu offline Auswertung des Testfalls



Konfigurator

RT-Ethernet Restbussimulation

F. Bartols

Motivation &
Einleitung

Hintergrund

Echtzeit Ethernet
Restbussimulation

Anwendung einer RBS
in RT-Ethernet
Modellierung von
Systemspezifikationen
Modellierung abstrakter
Testfälle
Verwendung von
Testfällen zur
Verhaltensmodellierung

**Umsetzung einer
Echtzeit Ethernet Rest-
bussimulationsumgebung**

Anwendung &
Ergebnisse

Zusammenfassung &
Ausblick

Implementierung des Restbussimulators

Anforderungen:

- Protokollkonforme Übertragung der Nachrichten
- Ausführung des abstrakten Testfallmodells
- Aufzeichnung des gesamten Datenverkehrs zu offline Auswertung des Testfalls

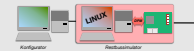


Konfigurator

Implementierung des Restbussimulators

Anforderungen:

- Protokollkonforme Übertragung der Nachrichten
- Ausführung des abstrakten Testfallmodells
- Aufzeichnung des gesamten Datenverkehrs zu offline Auswertung des Testfalls



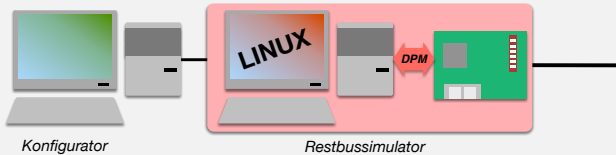
Konfigurator

Restbussimulator

1. Der Konfigurator dient dazu Testfälle zu modellieren und eine Analyse des aufgezeichneten Verkehrs zu erledigen

Anforderungen:

- Protokollkonforme Übertragung der Nachrichten
- Ausführung des abstrakten Testfallmodells
- Aufzeichnung des gesamten Datenverkehrs zu offline Auswertung des Testfalls



RT-Ethernet Restbussimulation

F. Bartols

Motivation &
Einleitung

Hintergrund

Echtzeit Ethernet
Restbussimulation

Anwendung einer RBS
in RT-Ethernet
Modellierung von
Systemspezifikationen
Modellierung abstrakter
Testfälle
Verwendung von
Testfällen zur
Verhaltensmodellierung
Umsetzung einer
Echtzeit Ethernet Rest-
bussimulationsumgebung

Anwendung &
Ergebnisse

Zusammenfassung &
Ausblick

20 / 29

Implementierung des Restbussimulators

Anforderungen:

- Protokollkonforme Übertragung der Nachrichten
- Ausführung des abstrakten Testfallmodells
- Aufzeichnung des gesamten Datenverkehrs zu offline Auswertung des Testfalls



Implementierung des Restbussimulators

Anforderungen:

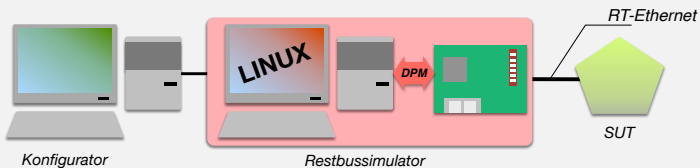
- Protokollkonforme Übertragung der Nachrichten
- Ausführung des abstrakten Testfallmodells
- Aufzeichnung des gesamten Datenverkehrs zu offline Auswertung des Testfalls



1. Die Hardwareplattform basiert auf zwei Komponenten die über Dual-Port-Memory miteinander verbunden sind
2. X86-Workstation liefert abstrakte Testfälle ein bereitet die Daten zur Übertragung vor
3. Mikrocontroller sorgt anschließend für die protokollkonforme Übertragung der Nachrichten
4. Und versieht gesendete und empfangene Nachrichten mit einem Zeitstempel damit nicht-funktionale Zeitanforderungen überprüft werden können

Anforderungen:

- Protokollkonforme Übertragung der Nachrichten
- Ausführung des abstrakten Testfallmodells
- Aufzeichnung des gesamten Datenverkehrs zu offline Auswertung des Testfalls



RT-Ethernet Restbussimulation

F. Bartols

Motivation &
Einleitung

Hintergrund

Echtzeit Ethernet
Restbussimulation

Anwendung einer RBS
in RT-Ethernet
Modellierung von
Systemspezifikationen
Modellierung abstrakter
Testfälle
Verwendung von
Testfällen zur
Verhaltensmodellierung
**Umsetzung einer
Echtzeit Ethernet Rest-
bussimulationsumgebung**

Anwendung &
Ergebnisse

Zusammenfassung &
Ausblick

20 / 29

Implementierung des Restbussimulators

Anforderungen:

- Protokollkonforme Übertragung der Nachrichten
- Ausführung des abstrakten Testfallmodells
- Aufzeichnung des gesamten Datenverkehrs zu offline Auswertung des Testfalls



Agenda

- Motivation & Einleitung
- Hintergrund RT-Ethernet & Restbussimulation
- Echtzeit Ethernet Restbussimulation
- **Anwendung & Ergebnisse**
- Zusammenfassung & Ausblick

1. Das System-under-Test empfängt die vom Simulator generierten und sendet Nachrichten die vom Simulator empfangen werden

1 Motivation & Einleitung

2 Hintergrund RT-Ethernet & Restbussimulation

3 Echtzeit Ethernet Restbussimulation

4 Anwendung & Ergebnisse

5 Zusammenfassung & Ausblick

Agenda

- 1 Motivation & Einleitung
- 2 Hintergrund RT-Ethernet & Restbussimulation
- 3 Echtzeit Ethernet Restbussimulation
- 4 Anwendung & Ergebnisse
- 5 Zusammenfassung & Ausblick

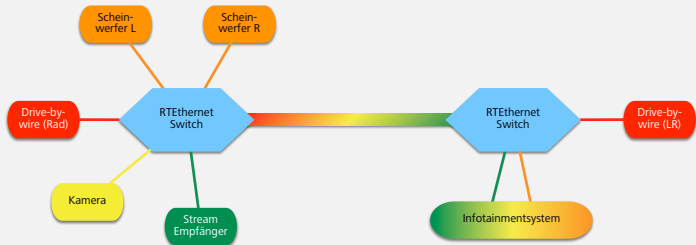
Beschreibung des Anwendungsbeispiels Systemüberblick



- Infotainmentsystem sendet Lichtzustände
- Scheinwerfer quittieren eine Zustandsänderung
- Scheinwerfer Senden periodisch aktuellen Zustand
- Infotainmentsystem stellt den aktuellen Zustand dar

Beschreibung des Anwendungsbeispiels

Systemüberblick



- Infotainmentsystem sendet Lichtzustände
- Scheinwerfer Quittieren eine Zustandsänderung
- Scheinwerfer Senden periodisch aktuellen Zustand
- Infotainmentsystem stellt den aktuellen Zustand dar

RT-Ethernet
Restbussimulation

F. Bartols

Motivation &
Einleitung

Hintergrund

Echtzeit Ethernet
Restbussimulation

Anwendung &
Ergebnisse

Zusammenfassung &
Ausblick

Beschreibung des Anwendungsbeispiels

Systemüberblick



- Infotainmentsystem sendet Lichtzustände
- Scheinwerfer Quittieren eine Zustandsänderung
- Scheinwerfer Senden periodisch aktuellen Zustand
- Infotainmentsystem stellt den aktuellen Zustand dar

Beschreibung des Anwendungsbeispiels

Systemüberblick

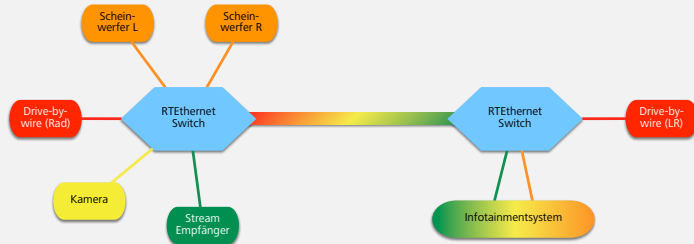


- Infotainmentsystem sendet Lichtzustände
- Scheinwerfer Quittieren eine Zustandsänderung
- Scheinwerfer Senden periodisch aktuellen Zustand
- Infotainmentsystem stellt den aktuellen Zustand dar

1. Gesamtsystem entstammt einem RT-Ethernet Demonstrationssystem, dass verschiedene Anwendungsklassen in RT-Ethernet darstellt

Beschreibung des Anwendungsbeispiels

Systemüberblick



- Infotainmentsystem sendet Lichtzustände
- Scheinwerfer Quittieren eine Zustandsänderung
- Scheinwerfer Senden periodisch aktuellen Zustand
- Infotainmentsystem stellt den aktuellen Zustand dar



RT-Ethernet
Restbussimulation

F. Bartols

Motivation &
Einleitung

Hintergrund

Echtzeit Ethernet
Restbussimulation

Anwendung &
Ergebnisse

Zusammenfassung &
Ausblick

Beschreibung des Anwendungsbeispiels

Systemüberblick



- Infotainmentsystem sendet Lichtzustände
- Scheinwerfer Quittieren eine Zustandsänderung
- Scheinwerfer Senden periodisch aktuellen Zustand
- Infotainmentsystem stellt den aktuellen Zustand dar

Beschreibung des Anwendungsbeispiels

Systemüberblick

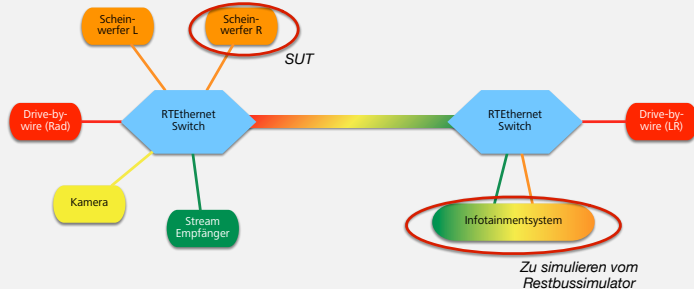


- Infotainmentsystem sendet Lichtzustände
- Scheinwerfer Quittieren eine Zustandsänderung
- Scheinwerfer Senden periodisch aktuellen Zustand
- Infotainmentsystem stellt den aktuellen Zustand dar

1. Das Infotainmentsystem dient als Digitales Armaturenbrett mit dem die Scheinwerfer bedient werden
2. und sendet neue Lichtzustände an die Scheinwerfer
3. Die Scheinwerfer quittieren jede Zustandsänderung und senden periodisch ihren aktuellen Zustand
4. Das Infotainmentsystem stellt den aktuellen Scheinwerferzustand dar

Beschreibung des Anwendungsbeispiels

Systemüberblick



- Infotainmentsystem sendet Lichtzustände
- Scheinwerfer Quittieren eine Zustandsänderung
- Scheinwerfer Senden periodisch aktuellen Zustand
- Infotainmentsystem stellt den aktuellen Zustand dar



RT-Ethernet
Restbussimulation

F. Bartols

Motivation &
Einleitung

Hintergrund

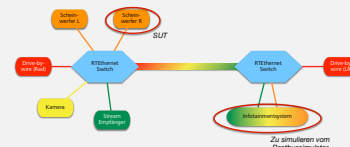
Echtzeit Ethernet
Restbussimulation

Anwendung &
Ergebnisse

Zusammenfassung &
Ausblick

Beschreibung des Anwendungsbeispiels

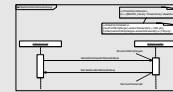
Systemüberblick



- Infotainmentsystem sendet Lichtzustände
- Scheinwerfer Quittieren eine Zustandsänderung
- Scheinwerfer Senden periodisch aktuellen Zustand
- Infotainmentsystem stellt den aktuellen Zustand dar

Verifikation einer Scheinwerferfunktion

Systemspezifikation mittels UML-MARTE Sequenzdiagramm



- Zustandsquittierung durch Sequenzdiagramm
- Zeitanforderungen durch VSL
 - Reaktionsgeschwindigkeit: 900 µs
 - Jitter: 100 µs

1. Das SUT ist in diesem Fall der rechte Scheinwerfer
2. Der Restbussimulator simuliert in diesem Fall das Infotainmentsystem und übernimmt die Uhrensynchronisation
3. Sendet dementsprechend Zustandsänderungsnachrichten und Synchronisationsnachrichten an den Scheinwerfer

Verifikation einer Scheinwerferfunktion

Systemspezifikation mittels UML-MARTE Sequenzdiagramm



RT-Ethernet
Restbussimulation
F. Bartols

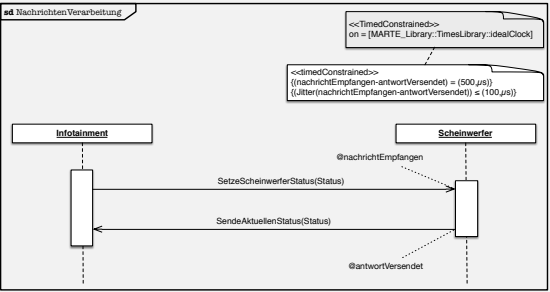
Motivation &
Einleitung

Hintergrund

Echtzeit Ethernet
Restbussimulation

Anwendung &
Ergebnisse

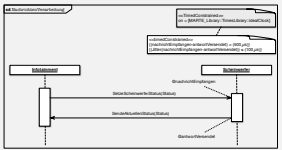
Zusammenfassung &
Ausblick



- Zustandsquittierung durch Sequenzdiagramm
- Zeitanforderungen durch VSL
 - Reaktionsgeschwindigkeit: 500 μs
 - Jitter: 100 μs

Verifikation einer Scheinwerferfunktion

Systemspezifikation mittels UML-MARTE Sequenzdiagramm



- Zustandsquittierung durch Sequenzdiagramm
- Zeitanforderungen durch VSL
 - Reaktionsgeschwindigkeit: 500 μs
 - Jitter: 100 μs

Leistungsanforderungen

Testfall und Ergebnisse einer Restbussimulation

T	S1	S2	S3	S4	S5
T1	10	20	30	40	50
T2	10	20	30	40	50
T3	10	20	30	40	50
T4	10	20	30	40	50
T5	10	20	30	40	50
T6	10	20	30	40	50
T7	10	20	30	40	50
T8	10	20	30	40	50
T9	10	20	30	40	50
T10	10	20	30	40	50

- Funktionale Anforderung erfüllt
- Erwarteten Werte liegen am Ausgang des SUT an
- Nicht-funktionale Zeitanforderungen erfüllt
- Senderate & Reaktionszeit sind im definierten Bereich

1. Als Systemspezifikation dient ein UML-MARTE-Sequenzdiagramm um die funktionale Anforderung der Zustandsquittierung zu modellieren
2. Nicht-funktionale Anforderungen sind durch die Value Specific Language definiert
3. Die Anforderung an die Reaktionsgeschwindigkeit ist mit 500 μs modelliert
4. Der Anforderung des Jitters der Reaktionsgeschwindigkeit ist mit 100 μs modelliert

Leistungsanforderungen

Testfall und Ergebnisse einer Restbussimulation

T	1s	2s	3s	...	61s	62s
U	u ₁ = SW_AUS	u ₁ = HS_EIN	u ₁ = HS_AUS	...	u ₁ = HS_EIN	u ₁ = HS_AUS
Y _{soll}	y ₁ = SW_AUS	y ₁ = HS_EIN	u ₁ = HS_AUS	...	u ₁ = HS_EIN	y ₁ = SW_AUS
Y _{ist}	y ₁ = SW_AUS	y ₁ = HS_EIN	y ₁ = SW_AUS	...	y ₁ = HS_EIN	y ₁ = SW_AUS
L _{soll}	h ₁ (u ₁ , y ₁) = 500µs					
J _{Lsoll}	j _{L1} (h ₁) ≤ 50µs					
L _{ist}	h ₁ (u ₁ , y ₁) = 517 - 519µs, MEDIAN=518µs					
R _{soll}	r ₁ (y ₁) = 5000µs					
J _{Rsoll}	j _{R1} (r ₁) ≤ 10µs					
R _{ist}	r ₁ (y ₁) = 4997 - 5002µs, MEDIAN=5000µs					

- Funktionale Anforderung erfüllt
- Erwarteten Werte liegen am Ausgang des SUT an
- Nicht-funktionale Zeitanforderungen erfüllt
- Senderate & Reaktionszeit sind im definierten Bereich



RT-Ethernet Restbussimulation

F. Bartols

Motivation &
Einleitung

Hintergrund

Echtzeit Ethernet
Restbussimulation

Anwendung &
Ergebnisse

Zusammenfassung &
Ausblick

Leistungsanforderungen

Testfall und Ergebnisse einer Restbussimulation

T	1s	2s	3s	...	61s	62s
U	u ₁ = SW_AUS	u ₁ = HS_EIN	u ₁ = HS_AUS	...	u ₁ = HS_EIN	u ₁ = HS_AUS
Y _{soll}	y ₁ = SW_AUS	y ₁ = HS_EIN	u ₁ = HS_AUS	...	u ₁ = HS_EIN	y ₁ = SW_AUS
Y _{ist}	y ₁ = SW_AUS	y ₁ = HS_EIN	y ₁ = SW_AUS	...	y ₁ = HS_EIN	y ₁ = SW_AUS
L _{soll}	h ₁ (u ₁ , y ₁) = 500µs					
J _{Lsoll}	j _{L1} (h ₁) ≤ 50µs					
L _{ist}	h ₁ (u ₁ , y ₁) = 517 - 519µs, MEDIAN=518µs					
R _{soll}	r ₁ (y ₁) = 5000µs					
J _{Rsoll}	j _{R1} (r ₁) ≤ 10µs					
R _{ist}	r ₁ (y ₁) = 4997 - 5002µs, MEDIAN=5000µs					

- Funktionale Anforderung erfüllt
- Erwarteten Werte liegen am Ausgang des SUT an
- Nicht-funktionale Zeitanforderungen erfüllt
- Senderate & Reaktionszeit sind im definierten Bereich

Leistungsanforderungen

Testfall und Ergebnisse einer Restbussimulation

T	1s	2s	3s	...	61s	62s
U	u ₁ = SW_AUS	u ₁ = HS_EIN	u ₁ = HS_AUS	...	u ₁ = HS_EIN	u ₁ = HS_AUS
Y _{soll}	y ₁ = SW_AUS	y ₁ = HS_EIN	u ₁ = HS_AUS	...	u ₁ = HS_EIN	y ₁ = SW_AUS
Y _{ist}	y ₁ = SW_AUS	y ₁ = HS_EIN	y ₁ = SW_AUS	...	y ₁ = HS_EIN	y ₁ = SW_AUS
L _{soll}	h ₁ (u ₁ , y ₁) = 500µs					
J _{Lsoll}	j _{L1} (h ₁) ≤ 50µs					
L _{ist}	h ₁ (u ₁ , y ₁) = 517 - 519µs, MEDIAN=518µs					
R _{soll}	r ₁ (y ₁) = 5000µs					
J _{Rsoll}	j _{R1} (r ₁) ≤ 10µs					
R _{ist}	r ₁ (y ₁) = 4997 - 5002µs, MEDIAN=5000µs					

- Funktionale Anforderung erfüllt
- Erwarteten Werte liegen am Ausgang des SUT an
- Nicht-funktionale Zeitanforderungen erfüllt
- Senderate & Reaktionszeit sind im definierten Bereich

1. In diesem Testfall sind die funktionalen und nicht-funktionalen Anforderungen definiert worden
2. Testfall besteht aus einer alternierenden Folge, die den Zustand den Scheinwerfer abwechselnd ein und aus schaltet, die der Scheinwerfer mit dem gleichen Datenpaket quittiert
3. Reaktionszeit auf den Erhalt der Zustandsnachricht ist auf 500 µs definiert. Der Jitter liegt bei 50 µs sodass eine Reaktionszeit zwischen 450 und 550 µs erlaubt ist
4. Das gleiche gilt für die Senderate der periodischen Übertragung des aktuellen Zustandes, die mit 5000 µs und einem Jitter von 10 µs definiert ist. Der Abstand zwischen zwei Nachrichten des aktuellen

Leistungsanforderungen

Testfall und Ergebnisse einer Restbussimulation



RT-Ethernet Restbussimulation

F. Bartols

Motivation &
Einleitung

Hintergrund

Echtzeit Ethernet
Restbussimulation

Anwendung &
Ergebnisse

Zusammenfassung &
Ausblick

T	1s	2s	3s	...	61s	62s
U	u ₁ = SW_AUS	u ₁ = HS_EIN	u ₁ = HS_AUS	...	u ₁ = HS_EIN	u ₁ = HS_AUS
Y _{soll}	y ₁ = SW_AUS	y ₁ = HS_EIN	u ₁ = HS_AUS	...	u ₁ = HS_EIN	y ₁ = SW_AUS
Y _{ist}	y ₁ = SW_AUS	y ₁ = HS_EIN	y ₁ = SW_AUS	...	y ₁ = HS_EIN	y ₁ = SW_AUS
L _{soll}	h ₁ (u ₁ , y ₁) = 500µs					
J _{Lsoll}	j _{L1} (h ₁) ≤ 50µs					
L _{ist}	h ₁ (u ₁ , y ₁) = 517 - 519µs, MEDIAN=518µs					
R _{soll}	r ₁ (y ₁) = 5000µs					
J _{Rsoll}	j _{R1} (r ₁) ≤ 10µs					
R _{ist}	r ₁ (y ₁) = 4997 - 5002µs, MEDIAN=5000µs					

- Funktionale Anforderung erfüllt
- Erwarteten Werte liegen am Ausgang des SUT an
- Nicht-funktionale Zeitanforderungen erfüllt
- Senderate & Reaktionszeit sind im definierten Bereich

Leistungsanforderungen

Testfall und Ergebnisse einer Restbussimulation

T	1s	2s	3s	...	61s	62s
U	u ₁ = SW_AUS	u ₁ = HS_EIN	u ₁ = HS_AUS	...	u ₁ = HS_EIN	u ₁ = HS_AUS
Y _{soll}	y ₁ = SW_AUS	y ₁ = HS_EIN	u ₁ = HS_AUS	...	u ₁ = HS_EIN	y ₁ = SW_AUS
Y _{ist}	y ₁ = SW_AUS	y ₁ = HS_EIN	y ₁ = SW_AUS	...	y ₁ = HS_EIN	y ₁ = SW_AUS
L _{soll}	h ₁ (u ₁ , y ₁) = 500µs					
J _{Lsoll}	j _{L1} (h ₁) ≤ 50µs					
L _{ist}	h ₁ (u ₁ , y ₁) = 517 - 519µs, MEDIAN=518µs					
R _{soll}	r ₁ (y ₁) = 5000µs					
J _{Rsoll}	j _{R1} (r ₁) ≤ 10µs					
R _{ist}	r ₁ (y ₁) = 4997 - 5002µs, MEDIAN=5000µs					

- Funktionale Anforderung erfüllt
- Erwarteten Werte liegen am Ausgang des SUT an
- Nicht-funktionale Zeitanforderungen erfüllt
- Senderate & Reaktionszeit sind im definierten Bereich

Leistungsanforderungen

Testfall und Ergebnisse einer Restbussimulation

T	1s	2s	3s	...	61s	62s
U	u ₁ = SW_AUS	u ₁ = HS_EIN	u ₁ = HS_AUS	...	u ₁ = HS_EIN	u ₁ = HS_AUS
Y _{soll}	y ₁ = SW_AUS	y ₁ = HS_EIN	u ₁ = HS_AUS	...	u ₁ = HS_EIN	y ₁ = SW_AUS
Y _{ist}	y ₁ = SW_AUS	y ₁ = HS_EIN	y ₁ = SW_AUS	...	y ₁ = HS_EIN	y ₁ = SW_AUS
L _{soll}	h ₁ (u ₁ , y ₁) = 500µs					
J _{Lsoll}	j _{L1} (h ₁) ≤ 50µs					
L _{ist}	h ₁ (u ₁ , y ₁) = 517 - 519µs, MEDIAN=518µs					
R _{soll}	r ₁ (y ₁) = 5000µs					
J _{Rsoll}	j _{R1} (r ₁) ≤ 10µs					
R _{ist}	r ₁ (y ₁) = 4997 - 5002µs, MEDIAN=5000µs					

- Funktionale Anforderung erfüllt
- Erwarteten Werte liegen am Ausgang des SUT an
- Nicht-funktionale Zeitanforderungen erfüllt
- Senderate & Reaktionszeit sind im definierten Bereich

1. Die Signale die am Ausgang des SUT liegen stimmen über den gesamten Testfall mit denen erwarteten Ausgängen überein
2. Auch die modellierten nicht-funktionalen Anforderungen werden erfüllt. Die Reaktionszeit liegt auf dem Intervall von 517 und 519 µs und im Mittel bei 518 µs.
3. Die Senderate zeitlichen Anforderungen der Senderate werden ebenfalls erfüllt und liegen auf dem Intervall von 4997 und 5002 µs und im Mittel bei 5000 µs.

Leistungsanforderungen

Testfall und Ergebnisse einer Restbussimulation



RT-Ethernet Restbussimulation

F. Bartols

Motivation &
Einleitung

Hintergrund

Echtzeit Ethernet
Restbussimulation

Anwendung &
Ergebnisse

Zusammenfassung &
Ausblick

T	1s	2s	3s	...	61s	62s
U	$u_1 = \text{SW_AUS}$	$u_1 = \text{HS_EIN}$	$u_1 = \text{HS_AUS}$...	$u_1 = \text{HS_EIN}$	$u_1 = \text{HS_AUS}$
Y_{soll}	$y_1 = \text{SW_AUS}$	$y_1 = \text{HS_EIN}$	$u_1 = \text{HS_AUS}$...	$u_1 = \text{HS_EIN}$	$y_1 = \text{SW_AUS}$
Y_{ist}	$y_1 = \text{SW_AUS}$	$y_1 = \text{HS_EIN}$	$y_1 = \text{SW_AUS}$...	$y_1 = \text{HS_EIN}$	$y_1 = \text{SW_AUS}$
L_{soll}	$h_1(u_1, y_1) = 500\mu\text{s}$					
$J_{L_{\text{soll}}}$	$j_{L1}(h_1) \leq 50\mu\text{s}$					
L_{ist}	$h_1(u_1, y_1) = 517 - 519\mu\text{s}$, MEDIAN=518 μs					
R_{soll}	$r_1(y_1) = 5000\mu\text{s}$					
$J_{R_{\text{soll}}}$	$j_{R1}(r_1) \leq 10\mu\text{s}$					
R_{ist}	$r_1(y_1) = 4997 - 5002\mu\text{s}$, MEDIAN=5000 μs					

- Funktionale Anforderung erfüllt
- Erwarteten Werte liegen am Ausgang des SUT an
- Nicht-funktionale Zeitanforderungen erfüllt
- Senderate & Reaktionszeit sind im definierten Bereich

Leistungsanforderungen

Testfall und Ergebnisse einer Restbussimulation

T	1s	2s	3s	...	61s	62s
U	$u_1 = \text{SW_AUS}$	$u_1 = \text{HS_EIN}$	$u_1 = \text{HS_AUS}$...	$u_1 = \text{HS_EIN}$	$u_1 = \text{HS_AUS}$
Y_{soll}	$y_1 = \text{SW_AUS}$	$y_1 = \text{HS_EIN}$	$u_1 = \text{HS_AUS}$...	$u_1 = \text{HS_EIN}$	$y_1 = \text{SW_AUS}$
Y_{ist}	$y_1 = \text{SW_AUS}$	$y_1 = \text{HS_EIN}$	$y_1 = \text{SW_AUS}$...	$y_1 = \text{HS_EIN}$	$y_1 = \text{SW_AUS}$
L_{soll}	$h_1(u_1, y_1) = 500\mu\text{s}$					
$J_{L_{\text{soll}}}$	$j_{L1}(h_1) \leq 50\mu\text{s}$					
L_{ist}	$h_1(u_1, y_1) = 517 - 519\mu\text{s}$, MEDIAN=518 μs					
R_{soll}	$r_1(y_1) = 5000\mu\text{s}$					
$J_{R_{\text{soll}}}$	$j_{R1}(r_1) \leq 10\mu\text{s}$					
R_{ist}	$r_1(y_1) = 4997 - 5002\mu\text{s}$, MEDIAN=5000 μs					

- Funktionale Anforderung erfüllt
- Erwarteten Werte liegen am Ausgang des SUT an
- Nicht-funktionale Zeitanforderungen erfüllt
- Senderate & Reaktionszeit sind im definierten Bereich

Agenda

- Motivation & Einleitung
- Hintergrund RT-Ethernet & Restbussimulation
- Echtzeit Ethernet Restbussimulation
- Anwendung & Ergebnisse
- Zusammenfassung & Ausblick

1. Damit werden funktionale und nicht funktionalen Anforderungen dieser Funktion der Scheinwerfer erfüllt und der Testfall ist damit erfolgreich

1 Motivation & Einleitung

2 Hintergrund RT-Ethernet & Restbussimulation

3 Echtzeit Ethernet Restbussimulation

4 Anwendung & Ergebnisse

5 Zusammenfassung & Ausblick

Agenda

1 Motivation & Einleitung

2 Hintergrund RT-Ethernet & Restbussimulation

3 Echtzeit Ethernet Restbussimulation

4 Anwendung & Ergebnisse

5 Zusammenfassung & Ausblick

Zusammenfassung & Fazit

Restbussimulation

- Restbussimulation ist eine Testmethodik im Entwicklungsprozess
- 3 Nachrichtenklassen & Synchronisierung müssten in RT-Ethernet durchgeführt werden
- Transparentes Verhalten für das SUT erforderlich
- RT-Ethernet Restbussimulation zur Verifikation der Nachrichtenübertragung
- Spezifikation des SUT erfolgt in Form von Modellen
- Funktionale und nicht-funktionale Anforderungen durch UML-MARTE

RT-Ethernet Restbussimulation

F. Bartols

Motivation &
Einleitung

Hintergrund

Echtzeit Ethernet
Restbussimulation

Anwendung &
Ergebnisse

Zusammenfassung &
Ausblick

- Restbussimulation ist eine Testmethodik im Entwicklungsprozess
- 3 Nachrichtenklassen & Synchronisierung müssten in RT-Ethernet durchgeführt werden
- Transparentes Verhalten für das SUT erforderlich
- RT-Ethernet Restbussimulation zur Verifikation der Nachrichtenübertragung
- Spezifikation des SUT erfolgt in Form von Modellen
- Funktionale und nicht-funktionale Anforderungen durch UML-MARTE

Zusammenfassung & Fazit

Restbussimulation

- Restbussimulation ist eine Testmethodik im Entwicklungsprozess
- 3 Nachrichtenklassen & Synchronisierung müssten in RT-Ethernet durchgeführt werden
- Transparentes Verhalten für das SUT erforderlich
- RT-Ethernet Restbussimulation zur Verifikation der Nachrichtenübertragung
- Spezifikation des SUT erfolgt in Form von Modellen
- Funktionale und nicht-funktionale Anforderungen durch UML-MARTE

Zusammenfassung & Fazit

- Entwicklung des abstrakten Testfallmodells zur Überprüfung von nicht-funktionalen Anforderungen
- Verwendung eines Abstrakten Testfallmodells für zwei Aufgaben:
 - Verifikation von funktionalen und nicht-funktionalen Anforderungen
 - Verwendung als Simulationsmodell zur Verhaltensmodellierung
- RT-Ethernet Restbussimulation zur Etablierung als Fahrzeugnetzwerk erforderlich

1. Die Restbussimulation ist eine Testmethodik innerhalb des modellbasierten Entwicklungsprozesses im Automobil
2. Bei einer RT-Ethernet Restbussimulation müssen die Nachrichtenklassen sowie die Synchronisierungsmechanismen unterstützt werden
3. Das Verhalten der Restbussimulation muss transparent für das System-under-Test erfolgen und darf keine Unterschiede zum realen System aufweisen
4. Die RT-Ethernet Restbussimulation wird in dieser Arbeit zur Verifikation der Nachrichtenübertragung verwendet
5. Dabei müssen Spezifikationen in Form von Modellen entwickelt

- Entwicklung des abstrakten Testfallmodells zur Überprüfung von nicht-funktionalen Anforderungen
- Verwendung eines Abstrakten Testfallmodells für zwei Aufgaben:
 - Verifikation von funktionalen und nicht-funktionalen Anforderungen
 - Verwendung als Simulationsmodell zur Verhaltensmodellierung
- RT-Ethernet Restbusssimulation zur Etablierung als Fahrzeugnetzwerk erforderlich

- Entwicklung des abstrakten Testfallmodells zur Überprüfung von nicht-funktionalen Anforderungen
- Verwendung eines Abstrakten Testfallmodells für zwei Aufgaben:
 - Verifikation von funktionalen und nicht-funktionalen Anforderungen
 - Verwendung als Simulationsmodell zur Verhaltensmodellierung
- RT-Ethernet Restbussimulation zur Etablierung als Fahrzeugnetzwerk erforderlich

- Vereinfachung der Systemarchitektur auf eine gemeinsame Hardwarebasis
- Umsetzung eines Konzepts zur Modellierung von reaktiven Verhalten
- Automatische Analyse des aufgezeichneten Verkehrs

1. Damit auch die nicht-funktionalen Anforderungen getestet werden können muss ein abstraktes Testfallmodell diese Informationen darstellen
2. Das Testfallmodell wird anschließend auf der Restbussimulation ausgeführt sodass es sich für zwei Aufgaben eignet
3. Zur Verifikation von funktionalen und nicht-funktionalen Anforderungen
4. Und als Simulationsmodell zur Beschreibung der Restbussimulation
5. Die Entwicklung einer Restbussimulation für RT-Ethernet ermöglicht es RT-Ethernet als Fahrzeugnetzwerk zu etablieren
6. Es zählen nicht nur die Eigenschaften einer neuen Technologie

- Vereinfachung der Systemarchitektur auf eine gemeinsame Hardwarebasis
- Umsetzung eines Konzepts zur Modellierung von reaktiven Verhalten
- Automatische Analyse des aufgezeichneten Verkehrs

- Vereinfachung der Systemarchitektur auf eine gemeinsame Hardwarebasis
- Umsetzung eines Konzepts zur Modellierung von reaktiven Verhalten
- Automatische Analyse des aufgezeichneten Verkehrs



1. Vereinfachung der Systemarchitektur auf eine Hardware-Plattform mit nur einer Komponente
2. Portierung des Konzepts auf eine Plattform mit RT-Ethernet fähiger Netzwerkkarte
3. Das Verhalten der Restbussimulation ist statisch und muss durch Testfälle modelliert werden Reaktives Verhalten ist auf diese Weise nicht möglich umzusetzen
4. Für Reaktives Verhalten muss ein Konzept entwickelt werden, dass reaktiv auf bestimmte Bedingungen eingegangen werden kann
5. Der aufgezeichnete Datenverkehr bei der Restbussimulation kann maschinell analysiert werden

Vielen Dank für eure
Aufmerksamkeit



RT-Ethernet
Restbussimulation
F. Bartols

Motivation &
Einleitung

Hintergrund

Echtzeit Ethernet
Restbussimulation

Anwendung &
Ergebnisse

Zusammenfassung &
Ausblick

Vielen Dank für eure
Aufmerksamkeit



Vielen Dank für eure
Aufmerksamkeit

