

Virtualisierung eingebetteter Echtzeitsysteme im Mehrkernbetrieb zur Partitionierung sicherheitsrelevanter Fahrzeugsoftware

Dominik Reinhardt



Universitätsverlag Ilmenau
2016

Impressum

Bibliografische Information der Deutschen Nationalbibliothek

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Angaben sind im Internet über <http://dnb.d-nb.de> abrufbar.

Diese Arbeit hat der Fakultät Informatik und Automatisierung als Dissertation vorgelegen

Tag der Einreichung: 24. Juni 2015

1. Gutachter: Prof. Dr. Ing. Winfried Kühnhauser
(Technische Universität Ilmenau)

2. Gutachter: Prof. Dr. Uwe Baumgarten
(Technische Universität München)

3. Gutachter: Prof. Dr. Markus Kucera
(Ostbayerische Technische Hochschule Regensburg)

Tag der Verteidigung: 12. April 2016

Technische Universität Ilmenau/Universitätsbibliothek

Universitätsverlag Ilmenau

Postfach 10 05 65

98684 Ilmenau

www.tu-ilmenau.de/universitaetsverlag

Herstellung und Auslieferung

Verlagshaus Monsenstein und Vannerdat OHG

Am Hawerkamp 31

48155 Münster

www.mv-verlag.de

ISBN 978-3-86360-135-5 (Druckausgabe)

URN [urn:nbn:de:gbv:ilm1-2016000182](http://nbn:de:gbv:ilm1-2016000182)

Titelphoto: photocase.com

Inhaltsverzeichnis

1	Einleitung	1
1.1	Problemstellung	1
1.2	Zielsetzung der Arbeit	3
1.3	Lösungsweg	4
1.4	Gliederung der Arbeit	5
2	Grundlagen und Stand der Technik	7
2.1	Betriebssystemkerne und deren Architekturen	7
2.2	Grundlagen der Virtualisierung	8
2.2.1	Virtualisierte Systeme und deren Anforderungen	8
2.2.2	Hypervisortypen	9
2.3	Automotive Open System Architecture (AUTOSAR)	10
2.4	Funktionale Sicherheit für Straßenfahrzeuge	11
2.5	Der Infineon TriCore AURIX Mikrocontroller	12
3	Analyse zur Partitionierung sicherheitsrelevanter E/E-Funktionscluster	15
3.1	Herausforderung und Problemstellung	15
3.2	Zentralisierte Bordnetzarchitektur	16
3.3	Domänen-orientierte Bordnetzarchitektur	17
3.3.1	Segmentierung von Funktionen und Bildung von Funktionsclustern auf Architekturebene	18
3.3.2	Forderungen der ISO26262 an zukünftige E/E-Systeme	19
3.3.3	Softwareintegration mit gemischter Integritätseinstufung	20
3.4	Verlagerung von E/E-Funktionen	22
3.4.1	Skalierbarkeitsanforderungen an zukünftige E/E-Systeme	22
3.4.2	Identifizierte Integrationsszenarien	23
3.5	Zusammenfassung des Kapitels	25
4	Bewertung der Partitionierbarkeit sicherheitsrelevanter Softwarekomponenten	27
4.1	Herausforderung und Problemstellung	27
4.2	Verwandte Methoden und Techniken	28
4.3	Der PageRank	29
4.4	Der TrustRank	30

4.5	Der Software Component Rank	31
4.5.1	Bewertung von Fahrzeugsoftwarekomponenten	32
4.5.2	Komplexität und Konvergenz	34
4.5.3	Berücksichtigung sicherheitsrelevanter Softwarekomponenten	34
4.5.4	Gewichtete Beziehungen zwischen Softwarekomponenten	36
4.6	Analyse von ECU internen Softwareabhängigkeiten	37
4.7	Bewertung und Einsatzgebiete	39
4.8	Zusammenfassung des Kapitels	40
5	Ableitung von Kernelarchitekturen aus nicht-funktionalen Eigenschaften	43
5.1	Herausforderung und Problemstellung	43
5.2	Verwandte Methoden und Techniken	44
5.3	Attribute der Systemzuverlässigkeit	46
5.3.1	Verlässlichkeit	46
5.3.2	Verfügbarkeit	47
5.3.3	Wartbarkeit	47
5.3.4	Funktionale Sicherheit	48
5.4	Weitere nicht-funktionale Eigenschaften zur Systembewertung	49
5.4.1	Effizienz	49
5.4.2	Skalierbarkeit	50
5.5	Methode zur Bestimmung geeigneter Kernelarchitekturen	51
5.5.1	Grundstruktur eines NFE-Baumes	51
5.5.2	Graphenbasierte Darstellung und Gewichtung	54
5.5.3	Einführung von Querbeziehungen zur Reduktion der Komplexität	56
5.5.4	Gegensätzliche Zielstellungen nicht-funktionaler Eigenschaften	57
5.6	Ableitung Technischer Maßnahmen und Kernelarchitekturen am Beispiel funktionaler Sicherheit	59
5.6.1	Fehlerpotentiale	60
5.6.2	Technische Maßnahmen	60
5.6.3	Auswertung der gewichteten Ergebnisse	65
5.7	Zusammenfassung des Kapitels	70
6	Kostenanalyse zur Softwareintegration sicherheitsrelevanter Fahrzeugfunktionen	73
6.1	Herausforderung und Problemstellung	73
6.2	Verwandte Methoden und Techniken	75
6.3	Monolithische Systemintegration	76
6.3.1	Portabilität und Wartbarkeit	77
6.3.2	Funktionale Sicherheit	78
6.3.3	Skalierbarkeit	79
6.4	Einsatz eines Mikrokerns	81
6.4.1	Portabilität und Wartbarkeit	81
6.4.2	Funktionale Sicherheit	82
6.4.3	Skalierbarkeit	83

6.5	Systemvirtualisierung	83
6.5.1	Portabilität und Wartbarkeit	83
6.5.2	Funktionale Sicherheit	86
6.5.3	Skalierbarkeit	87
6.6	Analyse einer exemplarischen Hypervisorlösung	88
6.6.1	Richtlinien zur Migration virtualisierter Plattformen	88
6.6.2	Grundlegender Aufbau des Hypervisors	89
6.6.3	Ausnahmebehandlungen von Traps und Interrupts	91
6.6.4	Systemzustände und Sicherheitsmodell des Hypervisors	93
6.6.5	Systemperformanz	95
6.6.6	Portabilität und Wartbarkeit	102
6.6.7	Funktionale Sicherheit	104
6.6.8	Skalierbarkeit	107
6.7	Systembewertung	111
6.7.1	Effizienz	111
6.7.2	Portabilität	113
6.7.3	Wartbarkeit	113
6.7.4	Funktionale Sicherheit	114
6.7.5	Skalierbarkeit	115
6.8	Zusammenfassung des Kapitels	116
7	Kommunikationskanäle für virtuelle Steuergeräte	119
7.1	Herausforderung und Problemstellung	119
7.2	Verwandte Methoden und Techniken	120
7.3	Anbindung paravirtualisierter Hardwaretreiber	121
7.3.1	Realisierung des Zugriffsschutzes von Hardwareressourcen	122
7.3.2	Integrationsansätze von AUTOSAR Treibermodulen	124
7.3.3	Beurteilung dargestellter Integrationsansätze	128
7.4	Informationsaustausch zwischen virtuellen Maschinen	129
7.4.1	Realisierung eines generischen Kommunikationskanals	129
7.4.2	Schnittstellenbeschreibung zur domänenübergreifenden Interaktion	132
7.4.3	Routing von Informationen unter Verwendung einer privilegierten Domäne	132
7.5	Systemanalyse und Beurteilung der Messwerte	134
7.5.1	Antwortzeit in Relation zur Nachrichtengröße	136
7.5.2	Durchsatzmessungen in Relation zur Nachrichtengröße	138
7.6	Zusammenfassung des Kapitels	141
8	Demonstratoraufbau zur Konsolidierung von virtuellen AUTOSAR-Softwaresystemen	143
8.1	Herausforderung und Problemstellung	143
8.2	Methoden zum Informationsaustausch zwischen virtualisierten AUTOSAR Partitionen	144
8.3	Aufbau eines Hochintegrationsdemonstrators	148

8.3.1	Softwarearchitektur und portierte Fahrzeugsoftware	148
8.3.2	Informationsfluss zwischen virtuellen Maschinen	150
8.3.3	Initialisierung der Hardware	151
8.4	Untersuchung des Demonstratoraufbaus	153
8.4.1	Analyse des Rechenbedarfs	153
8.4.2	Funktionsbeanspruchung innerhalb des Hypervisors	154
8.4.3	Beurteilung der Messergebnisse	155
8.5	Zusammenfassung des Kapitels	156
9	Zusammenfassung und Ausblick	159
9.1	Ergebnisse der Arbeit	159
9.2	Weiterführende Arbeiten	161
A	Abkürzungen	163
B	Abbildungen	167