

Tagungsband

Embedded Software Engineering Kongress 2023

www.es-e-kongress.de



Embedded Software Engineering Kongress

Develop Your Software to the Highest Levels of Safety and Security



For over 40 years, world-class companies have trusted Green Hills Software's integrated software platforms, engineering services, and certification experts as the foundation to develop and deploy next-generation embedded systems with confidence to the highest levels of safety and security.

Call us on [+49 228 4330 777](tel:+492284330777) or contact us at ghs.com/go/contact

Tagungsband

Embedded Software Engineering Kongress 2023

4. bis 8. Dezember 2023

Preconference 1. Dezember 2023

Kongressbeirat 2023:

Stephan Ahrends

Jan Altenberg, Open Source Automation Development Lab (OSADL) eG

Thomas Batt, MicroConsult GmbH

Michael Bayer, The Happy Zebra Project

Maria Beyer-Fistrich, Vogel Communications Group GmbH & Co.KG

Roland Bickel, Hitex GmbH

Prof. Dr. Gert Bikker, Ostfalia Hochschule

Prof. Dr. Heinz-Peter Bürkle, Hochschule Aalen

Henning Butz

Raphael Dunker, SEW-EURODRIVE GmbH & Co. KG

Thomas Eisenbarth, Axivion GmbH

Dr. Carsten Emde, Open Source Automation Development Lab (OSADL) eG

Sebastian Gerstl, Heise Medien GmbH & Co.KG

Wolfram Gettert, Mixed Mode GmbH

Peter Gliwa, GLIWA embedded systems

Dr.-Ing. René Graf, Siemens AG

Rainer Grimm, Modernes C++

Martina Hafner, genua GmbH

Prof. Dr. Georg Hagel, Hochschule Kempten

Andreas Klinger, IT-Klinger

Maximilian Koller, intive automotive GmbH

Prof. Dr. Rainer Koschke, Universität Bremen

Caren Kresse, Open Source Automation Development Lab (OSADL) eG

Dr. Thomas Kuhn, Fraunhofer Institut IESE

Wolfgang Leimbach, SodiusWillert SAS

Prof. Dr. Jens Liebehenschel, Frankfurt University of Applied Sciences

Frank Listing, VoltStorage

Remo Markgraf, MicroConsult GmbH

Dr. Albrecht Mayer, Infineon Technologies

Prof. Dr. Jürgen Mottok, OTH Regensburg

Gudrun Neumann, SGS-TÜV Saar GmbH

Karl Nieratschker, SKT Nieratschker

Richard Oed, Richard Oed intelligent Engineering

Daniel Penning, embeff GmbH

Ingo Pohle, MicroConsult GmbH

Heiko Rießland, PLS Programmierbare Logik & Systeme GmbH

Stephan Roth, oose Innovative Informatik

Florian Schäffer, Grollmus München GmbH

Dr. Joachim Schlosser, Elektrobit Automotive GmbH

Marco Schmid, Schmid Elektronik AG

André Schmitz, Green Hills Software

Thomas Schütz, PROTOS Software GmbH

Prof. Dr. Christian Siemers, TU Clausthal

Peter Siwon, Systemisches Projektmanagement

Peter Sommerlad, Better Software

Andreas Stucki, Solcept AG

Klaus-Dieter Walter, SSV Software Systems GmbH

Siegfried Weigert, ibw industrieberatung

Johann Wiesböck

Andreas Willert, SodiusWillert SAS

Die Informationen in diesem Tagungsband werden ohne Rücksicht auf einen eventuellen Patentschutz veröffentlicht. Warennamen werden ohne Gewährleistung der freien Verwendbarkeit benutzt.

Die Texte und Abbildungen dieses Tagungsbandes wurden mit größter Sorgfalt zusammengestellt. Trotzdem können Fehler nicht vollständig ausgeschlossen werden. Verlag, Herausgeber und Autoren übernehmen für fehlerhafte Angaben und deren Folgen weder eine juristische noch eine sonstige Haftung.

Für Verbesserungsvorschläge und Hinweise sind der Verlag und Herausgeber dankbar.

Alle Rechte vorbehalten, auch die der fotomechanischen Wiedergabe und Speicherung auf elektronischen Medien. Nachdruck, digitale Verwendung jeder Art, Vervielfältigung nur mit schriftlicher Genehmigung des Verlags. Die gewerbliche Nutzung der auf der Tagung gezeigten Modelle und Präsentationen ist nicht zulässig.



Embedded Software Engineering Kongress

Copyright 2023
ELEKTRONIKPRAXIS
Vogel Communications Group GmbH & Co. KG
Max-Planck-Straße 7/9
97082 Würzburg

und

MicroConsult Microelectronics Consulting & Training GmbH
Charles-de-Gaulle-Straße 6
81737 München

Alle Rechte vorbehalten

ISBN 978-3-8343-6314-5



Analyse.
Automate.
Accelerate.
Assure Quality.



Statische Analyse- und dynamische Testing-Tools für zukunftssichere Software-Entwicklung

 Squish  Coco  Test Center  Axivion Static Code Analysis  Axivion Architecture Verification

Squish

Plattformunabhängiges funktionales GUI Testing Tool für praktisch jede Desktop-, Mobile-, Web-Oberfläche oder integrierte Systeme

Coco

Dynamische Überprüfung der Code-Abdeckung um Test-Protokolle zu optimieren und ein neues Level von Produktqualität zu erreichen

Test Center

Zentralisierte Testergebnis-Management Plattform zur Verbindung von Testautomatisierung mit dem gesamten Entwicklungsprozess

Regelkonformität

+ ISO 26262
+ ISO 14971
+ IEC 61508
+ IEC 62304
+ EN 50128 / EN 50657
+ SPICE® / ASPICE

Sicherheit

+ MISRA C/C++
+ CERT®
+ AUTOSAR C++14
+ CWE
+ ISO/IEC TS 17961
+ FDA Vorgaben

Axivion Architecture Verification

Architektur-Wiedergewinnung und -Archäologie für langfristig zielgerichtete und geplante Entwicklung von Software

Optimierte Lösungen

+ reibungslose Integration in vorhandene Tool Chain, CI/DevOps und IDE
+ individualisierbar
+ Service, Support, Coaching und Schulungen

Axivion Static Code Analysis

Umfassende statische Analyse um Software Erosion zu stoppen, potentielle Runtime Fehler zu finden und ROI zu verbessern

Inhaltsverzeichnis

KEYNOTE

- IoT-Sicherheit und vertrauenswürdige Elektronik** 20
Dr.-Ing. Matthias Hiller, Fraunhofer AISEC
- Change mich am Arsch!** 25
Axel Koch, Hochschule für angewandtes Management

ARCHITEKTUR

- Verhält sich Funktion zu Architektur wie Mikrobe zu Milieu?** 31
Andreas Willert, SodusWillert
- SysML, AUTOSAR und UML in Automotive-Anwendungen** 38
Walter von der Heiden, SodusWillert
- Embedded Software Manager Pattern** 49
Thomas Batt, MicroConsult
- Das Dreieck MBSE-SPES-SysML** 66
Wolfgang Hauck, ETAS
- Hardware-Obsoleszenz mit Architektur lösen** 76
*Dr. Jörg-Volker Müller, Systemum
Roman Koch, GMC Instruments*
- Docs-as-Code** 89
Sebastian Höller, intive
- Ein Computer-Vision-Projekt - zehn Plattformen** 94
Alexander Wirthmüller, MPSI Technologies
- Design for Change** 100
Martin Becker und Vasil Tenev, Fraunhofer IESE

ECHTZEIT

Introduction to Cyphal	110
<i>Alexander Entinger, LXRobotics</i>	
Korrektheit von Echtzeitsystemen	116
<i>Hubert B. Keller, ci-tec</i>	
Rust: Async statt RTOS	123
<i>Philipp Bormuth, awinia</i>	
Entwicklung einer Linux-Realtime-Applikation	133
<i>Andreas Klinger, IT-Klinger</i>	
Von Echtzeit- nach Mainline-Linux	144
<i>Jan Altenberg, Open Source Automation Development Lab (OSADL)</i>	
Uhrendrift in verteilten eingebetteten Systemen	150
<i>Christian Wenzel-Benner, GLIWA embedded systems</i>	
Ist mein System so Real-time wie gedacht?	156
<i>Alexander Bähr, Open Source Automation Development Lab (OSADL)</i>	
Latency Fighting	166
<i>Dr. Carsten Emde, Open Source Automation Development Lab (OSADL)</i>	

IMPLEMENTIERUNG

Von Funktionen zu Coroutinen	174
<i>Rainer Grimm, Modernes C++</i>	
Popular C++ Coding Guidelines in Automotive Software Development	181
<i>Frank van den Beuken, Perforce Software</i>	
Suchst du noch oder simulierst du schon?	187
<i>Frank Listing, VoltStorage</i>	
Beyond Protobuf	192
<i>Michael Thoma und Pierre Bayerl, HENSOLDT Sensors</i>	
Parallelization of C++ Code – Easy or Not?	198
<i>Klaas van Gend, High Tech Institute and Sioux Technologies</i>	
GraalVM on Embedded Devices	206
<i>Bruno Caballero, MicroDoc Computersysteme</i>	

Die Programmiersprache Rust	212
<i>Tobias Schmitt-Lechner, andrena objects</i>	
Einführung in C++20 Coroutinen	221
<i>Andreas Fertig, Unique Code</i>	
Python auf dem Mikrocontroller	224
<i>Frank Pilhofer, Zühlke Engineering</i>	
Was ist eigentlich Nebenläufigkeit?	230
<i>Moritz Strübe, MATHEMA</i>	
TDD und Mikrocontroller	235
<i>Daniel Penning, embef</i>	
Aha-Erlebnisse eines C++-Experten	241
<i>Dr. Timo Stripf, emmtrix Technologies</i>	

TEST & QUALITÄT

MISRA Compliance with AI Support	248
<i>Marcin Żwawa, Parasoft</i>	
Software-in-the-Loop-Testen leicht gemacht	256
<i>Markus Helmling und Patrick Welz, Vector Informatik</i>	
100% Coverage und doch nicht alles getestet?	263
<i>Michael Wittner, Razorcat Development</i>	
Wie sich Software-Fehler wie von alleine beheben	273
<i>André Schmitz, Green Hills Software</i>	
Code Coverage für Fortgeschrittene	279
<i>Frank Büchner, Hitex</i>	
Towards Autonomous Testing	287
<i>Alexej Popovič and Maximilian Blochberger, Qt Group</i>	
MISRA C:2023 System-Level Guidelines	292
<i>Michal Rozenau, Parasoft</i>	
Model-Based-Testing für Embedded-Systeme	299
<i>Thomas Schütz, PROTOS Software</i>	



Empowering Continuous Embedded Software Development and Testing

DevOps for System Testing

Set up your DevOps environment using Vector's virtual execution environments. Vector provides ready-to-run solutions for software-in-the-loop testing. Build the test platform based on open and partly free tooling without vendor lock-in. Depending on the use case, choose the appropriate building blocks from Vector and 3rd-party to test interactively or automated in virtual environments.

Increasing Software Quality, Minimize Technical Debt

The more complex software becomes, the more important it is to monitor its quality to identify weaknesses and take countermeasures in good time. Vector offers tools to analyze and test embedded software, with and without safety requirements.

Embedded Software

Vector is shaping the future of the Software-Defined Vehicle with an open DevOps toolchain and modular software platform. OEMs and suppliers worldwide benefit from first-class basic software tools and solutions for the development of embedded systems. The Vector Software Factory supports and automates workflows to implement your SDV project quickly and effectively.

KI & MACHINE LEARNING

Enhancing Traceability	305
<i>Sruthi Radhakrishnan, itemis</i>	
Spieglein an der Wand, welches ist das beste Modell im ganzen Land?	308
<i>Dr. Stefano Signoriello und Dr. Thomas Kittler, Infoteam Software</i>	
Mehr Effizienz und Qualität im Requirements Engineering mit KI	313
<i>Tobias Sommerfeld, Schaeffler Technologies</i>	
<i>Vincent Bertram, Simon Dehn und Viktor Slawik, RWTH Aachen University</i>	
<i>Dr. Dirk Fleischer und Julian Chander, BMW Group</i>	
<i>Andreas W. Müller, Schaeffler AG</i>	
<i>Nikola Kostov, Schaeffler Technologies</i>	
<i>Jan Niklas Schmitz, CSE Aachen</i>	
KI: Wunsch und Wirklichkeit	322
<i>Simon Duque Antón, comlet Verteilte Systeme</i>	

OPEN SOURCE

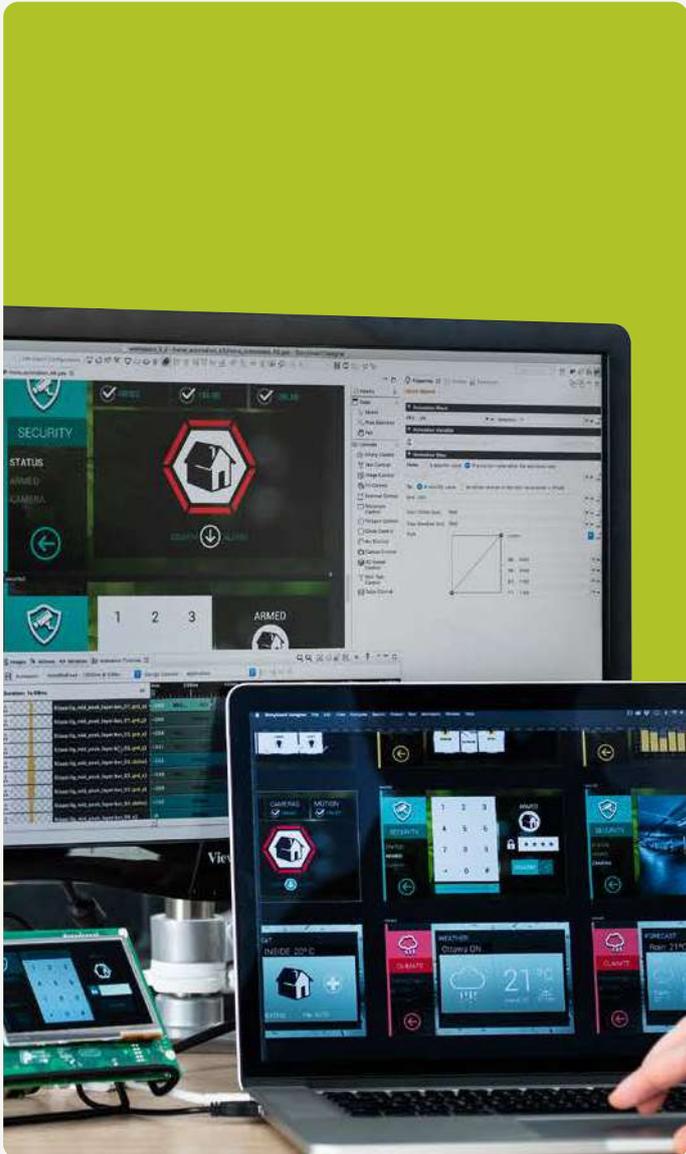
Multiprozess-Realtime-Systeme unter Linux	328
<i>Martin Steih, Lachmann & Rink</i>	
(Un)gelöste Herausforderungen bei Embedded-Linux-Updates	339
<i>Joschka Seydell, Zühlke Engineering</i>	
Build-Systeme für die effiziente Embedded-Linux-Entwicklung	348
<i>Bastian Krause, Pengutronix</i>	
TuxLayers	352
<i>Daniel Wenske, Avnet Silica</i>	

SAFETY

Misra C++ 2023	357
<i>Peter Sommerlad, Better Software</i>	
Qualifying a C Library	365
<i>Gerard Vink, TASKING</i>	
Safety and Security in Systems Programming	372
<i>Kris van Rens, High Tech Institute</i>	



Make Embedded GUI Development Effortless With Crank Storyboard



Embedded Software Engineering Kongress

December 4th - 8th, 2023

Visit our Booth at ESE Kongress to Learn More!

Kenne deine Abhängigkeiten	377
<i>Dr. Martin Becker, The MathWorks</i>	
Introduction to SOTIF: Implications for HPC Systems	384
<i>Abdelillah Ymlahi Ouazzani, Elektrobit</i>	
Proven in Use vs. SIL by Design	392
<i>Matthias Spranz, Hitex</i>	
SafMidWare: A Configurable Middleware for Future Safety-Critical Vehicle Applications	399
<i>Vinod Mohan Prabhu, Dr.-Ing. Günter Kessler und Dr.-Ing. Marco Jentges, FEV.io</i>	
<i>Prof. Dr.-Ing. Stefan Kowalewski, Informatik 11 Embedded Software, RWTH Aachen University</i>	
Towards Software-Defined Vehicles	413
<i>Nikola Velinov, Green Hills Software</i>	

SECURITY

Dann machen wir halt schnell eine digitale Signatur dran	422
<i>David Kauschke und Dr.-Ing. Markus Wamser, Ingenics Digital</i>	
Wie bereiten wir uns auf den Cyber Resilience Act vor?	430
<i>Dr. Markus Fockel, Fraunhofer IEM</i>	
The Power of Formal Methods Integrated in Regular Development Workflows	436
<i>Olivier Korach, TrustInSoft</i>	
Die Kraft der SBOM: Erschließung der Sicherheit der Software-Lieferkette im Ökosystem des vernetzten Fahrzeugs	450
<i>Gregor Knappik, VicOne</i>	
Reducing Supply Chain Risks	459
<i>Cris Sinnott and Dr. Aditya Deshpande, BlackBerry QNX</i>	
<i>Matthew Moser, Florian Antony and Jonathan Mohring, Itemis</i>	
<i>Gerhard Steininger, Itemis Consulting</i>	
End-to-end Derivation of Cybersecurity Requirements in Different Levels of the V-Model Using TARA Methods	477
<i>Dr. Thomas Liedtke, Magility Cyber Security</i>	
<i>Dr. Richard Messnarz, I.S.C.N.</i>	
Automatisierte Überprüfung der Sicherheitsaspekte von Softwarearchitekturen	484
<i>Sebastian Krings, Qt Group</i>	
Embedded Secure by Design	491
<i>Alois Cavelti, Solcept</i>	

AUTOMOTIVE

Software-Defined Vehicle	496
<i>Morten Huber und Thorsten Gerke, Dassault Systèmes Deutschland</i>	
Domänenspezifische Middleware-Lösungen für ADAS- und AD-Systeme	504
<i>Stjepan Dujmovic, Robert Bosch</i> <i>Uwe Maier, ETAS</i>	
Firmware Security Module	510
<i>Philipp Jungklass und Randolph Barg, Ingenieurgesellschaft Auto und Verkehr</i>	
ASPICE im Kontext von SAFe – Ein Erfahrungsbericht	519
<i>Christian Böttcher und Vikram Yadav Krishnamurthy, CARIAD</i>	
Software-Testmanagement im Automotive-Projekt	535
<i>Wolfgang Rohé, Razorcat Development</i>	
V2X mit dem Smartphone	559
<i>Daniel Lux, Lars Kelm, Florian Pramme, Ostfalia-Hochschule für angewandte Wissenschaften</i>	
Scalable Automotive Software Factory	566
<i>Matthias Wernicke, Vector Informatik</i> <i>Dr. Patrick Bartsch, Amazon Web Services</i>	
Safety Challenges of Highly Automated Systems	573
<i>Iwo Kurzidem, Fraunhofer IKS</i>	

IOT & INDUSTRIE 4.0

Integrationsmuster: Migrationspfad für die Industrie 4.0	574
<i>Frank Schnicke, Fraunhofer IESE</i>	
IoT im Miniformat	575
<i>Detlef Vollmann, vollmann engineering</i>	
Echtzeit-OT-IT Konvergenz für moderne Maschinen mit IoT Readiness	584
<i>Robert Schachner, embedded ocean</i>	
Sicherheit in dezentralen Energiesystemen	587
<i>Martin Lautenbacher, Ingenics Digital</i>	

TECHNOLOGIE – FORSCHUNG – INNOVATION

Software-Verifikation für Quantencomputer	594
<i>Marc Maußner, infoteam Software</i>	
Entfesselte Kreativität und Effizienz im mentalen Bootcamp	603
<i>Marco Schmid, Schmid Elektronik</i>	
How Autonomous Systems Become Reality	621
<i>Gereon Weiß, Fraunhofer Institute for Cognitive Systems IKS</i>	
Abschalten? Geht es noch?	627
<i>Dr. Jasmin S. A. Link, Universität Hamburg</i>	

AGILE TRANSFORMATION

CI/CD: Tests und Reports aus einem Guss für alle Stakeholder	632
<i>Hermann Bayala, Rohde & Schwarz</i>	
Eigenes Agile Framework vs. Blueprint: Which Way to Go?	635
<i>Christoph Schmiedinger, borisgloger consulting</i>	
Agile und konforme Prozessgestaltung in Projektteams	640
<i>Ralf Bürger, Systematic Software Engineering (SSE), Martin Becker, Fraunhofer IESE</i>	
Agile Entwicklung, Softwarearchitektur, Funktionale Sicherheit – drei Sichten einer Systemherausforderung	646
<i>Dr. Joachim Schlosser, Christoph Patzelt, Dr. Ulrich Kirchmaier und Dr. Viktor Zeißler, Elektrobit Automotive</i>	

SOFTWARE ENGINEERING MANAGEMENT

Where the Hell Are my Semantics?	659
<i>Albrecht Schwarz, ETAS</i>	
Das Digitale Warehouse	681
<i>Christian Wehebrink und Dr. Andreas Achtzehn, Robert Bosch</i>	
Unbewusste Fehler bei der Embedded-Systementwicklung	686
<i>Thomas Weber, Zühlke Engineering</i>	
Secure Software Development Lifecycle (SSDLC) für jedermann	693
<i>Jürgen Messerer, bbv Software Services</i>	

MENSCH – TEAM – MANAGEMENT

Barcamp mit Lösungspotential: Open Space	694
<i>Andreas Stucki, Solcept</i>	
Mitarbeiter nachhaltig motivieren	700
<i>Frank Benkert, FRB Computersysteme</i>	
Rock the Transformation	704
<i>Tobias Sommerfeld, Schaeffler Technologies; Sarah Stettin, Schaeffler</i>	
What the hack!	709
<i>Dr. René Graf, Siemens AG</i>	
Die Zukunftsorganisation	717
<i>Michael Bayer und Florian Schäffer, Zebra Kollektiv</i>	
Die Wiedergeburt des klassischen Führungsstils	723
<i>Horst Kostal, Process Fellows</i>	
Ingenieurmäßige Organisationsentwicklung	727
<i>Matthias Künzi, visuellklar</i>	
Erfolgreiche Unternehmensveränderung	737
<i>Daniel Westermayr, Colenet</i>	

PROJECT STORIES

Nicht wachstumsfähige Strukturen im Maschinenbau aufbrechen	740
<i>Dr.-Ing. Jan Pinkowski und Jan Schroeder, Jungheinrich Norderstedt Daniel Penning, embef</i>	
Ein neuer Stern am Himmel der HMI-Werkzeuge?	745
<i>Andy Walter, André Wengert und Felix Marek, Cloudflight Germany</i>	
Der lange Weg zur Security	754
<i>Willi Flühmann, Noser Engineering</i>	
Avoiding Spaghetti Code in Large Python Projects	761
<i>Tal Avidan, Aurora Labs</i>	

TIPPS – TRICKS – LÖSUNGEN – VORTRAG-ABSTRACTS DER PRECONFERENCE

Methods to Improve Software Security	762
<i>Green Hills Software</i>	
Popular C++ Coding Guidelines in Automotive Software Development	762
<i>Perforce</i>	
Robustheitstests für Embedded Systems	763
<i>PROTOS Software</i>	
Statische Analyse und dynamisches Testing	764
<i>Qt Group</i>	
Enabling Future Software Defined Vehicles	764
<i>Vector Informatik</i>	
Delivering Safe and Secure Code with Formal Methods Based Static Code Analyzers	764
<i>TrustInSoft</i>	
Architektur gestalten gemäß arc42 mit IBM Rhapsody	765
<i>SodiusWillert</i>	
Auswahl des passenden Safety-Controllers für IEC61508 Anwendungen	766
<i>Hitex</i>	
Ausführbare Architekturen für Embedded-Systeme	766
<i>PROTOS Software</i>	

Data Intelligence	766
<i>Vector Austria</i>	
Medizintechnik: Wie eine einzigartige Software-Architektur eine schnelle Zertifizierung ermöglicht	767
<i>Crank AMETEK</i>	
Bug Location Using Code Coverage	767
<i>Qt Group</i>	
A Toolchain for Safe and Secure Embedded Software Development	767
<i>Green Hills Software</i>	
MISRA-Konformität mit Unterstützung von KI	768
<i>Parasoft</i>	
How to Streamline Embedded Development and Reduce Time- to-Market with Cloud Hardware Targets	768
<i>BlackBerry QNX</i>	
The GUI Landscape in Rust	769
<i>SixtyFPS</i>	
ISO 26262 Teil 6 automatisieren mit Hilfe von Understand von SciTools	769
<i>Emenda</i>	
Firmenverzeichnis: Alle Goldsponsoren und Eventpartner von A bis Z	771



Digitale Lösungen für die Smart Industry.

Kundenspezifische Softwareentwicklung
für die Industrielle Interoperabilität.

Wir sind Ihr verlässlicher Partner für Smart Industrial Solutions. Als Südwestfalens größtes Beratungs- und Entwicklungszentrum für Industriesoftware realisieren wir individuelle Lösungen des Maschinen- und Anlagenbaus sowie des produzierenden Gewerbes. Wir machen Industrie smart und intelligent, verleihen Maschinen neue Funktionalitäten, vernetzen sie, optimieren Prozesse und kreieren innovative Geschäftsmodelle.

Mit Stolz verhelfen wir Technologie- und Weltmarktführern durch die Entwicklung von Maschinensoftware, IIoT, Embedded, IT/OT, Cloud-Diensten, Digitalen Zwillingen u.v.m. zu ihrem Projekterfolg.

- ✓ Über 100 Entwickler an zwei Standorten
- ✓ Eigenes großes Hardware- und Maschinenlabor
- ✓ Seit 40 Jahren zuverlässiger Entwicklungsprofi
- ✓ Seit 1998 durchgängig DIN EN ISO 9001 zertifiziert
- ✓ Mitglied u. a. in der OPC Foundation, EtherCAT Technology Group, VDMA
- ✓ Consulting, Software, Embedded aus einer Hand
- ✓ Unterstützung bei Ressourcen-Engpässen
- ✓ Über 100 Kunden aus rund 50 Branchen
- ✓ Breites, industrielles Branchenwissen und Umsetzungs-Know-how



IoT-Sicherheit und vertrauenswürdige Elektronik

Eine Perspektive aus der Forschung zu aktuellen Risiken und zukünftigen Technologien

Dr.-Ing. Matthias Hiller, Fraunhofer AISEC¹

Innovationen im Bereich der Hardware erweitern ständig die Grenzen von Embedded-Systemen in Bezug auf Rechenleistung, Kommunikation und Effizienz. Dies ermöglicht ihren Einsatz in immer neuen Anwendungen, z.B. in den Bereichen Internet der Dinge (IoT), Edge Computing oder eingebettete KI. Allerdings vergrößert sich dadurch auch die Angriffsfläche – und damit das Sicherheitsrisiko für das Gerät an sich, die verarbeiteten Daten, und die darauf ausgeführte Anwendung.

Eine Bestandsaufnahme: Embedded Devices im IoT aus Hardware-Sicht

Eingebettete Systeme und insbesondere vernetzte Systeme im IoT interagieren mit ihrer Umwelt durch eine Anwendung, die als Applikationssoftware auf dem System ausgeführt wird. Darunter liegen das Betriebssystem, die Systemsoftware, wie Firmware und Treiber, und schließlich auf unterster Ebene die Hardware. Fehler und Sicherheitslücken in der Applikations- und Systemsoftware können dazu führen, dass eine große Anzahl von Geräten durch remote Schnittstellen kompromittiert wird. Diese Fehler können durch Updates behoben werden. Die darunterliegende Hardware verhält sich aufgrund mehrerer Aspekte anders: sie bildet den Vertrauensanker auf dem alle weiteren Sicherheitsfunktionen, und damit die Sicherheit und Vertrauenswürdigkeit des Gesamtsystems aufbauen, und kann ohne Hardwareaustausch im Allgemeinen nicht verändert werden.

Es gibt eine Vielzahl von Gründen, warum die Hardware unter Umständen nicht ausreichenden Schutz für eine Anwendung und die verarbeiteten Daten bieten kann, wie schlichtweg das Fehlen der nötigen Features, unzureichende oder fehlerhafte Spezifikation und Implementierung, kompromittierte oder fehlerhafte Hardware, oder Angriffe durch Dritte oder auch den eigentlichen Nutzer. Um die Sicherheit eines Gerätes über die ganze Lebensdauer hinweg zu gewährleisten, ist es wichtig, dass zu Beginn sichere Hardware ausgewählt oder entwickelt wird und vertrauenswürdige Komponenten in den Systemen verbaut werden.

In einer Studie für das Bundesamt für Sicherheit in der Informationstechnik (BSI) wurden z.B. am Fraunhofer AISEC Mikrocontroller im IoT Umfeld auf ihre Sicherheit gegen

¹ Teile der vorgestellten Forschungsarbeiten wurden vom Bayerischen Staatsministerium für Wirtschaft, Landesentwicklung und Energie durch das Projekt Trusted Electronik Center Bayern gefördert.

Seitenkanal- und Fehlerangriffe untersucht und es wurden für verschiedene Chips Möglichkeiten für Angriffe gezeigt².

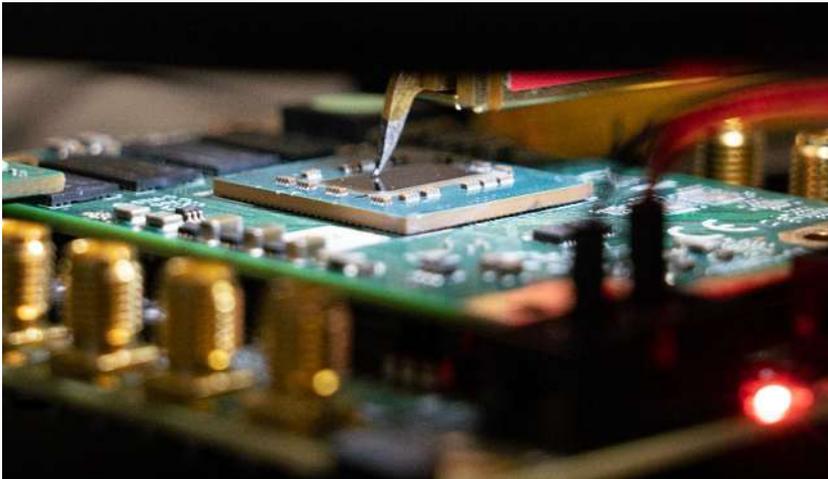


Abbildung 1: Seitenkanalangriff durch lokale EM-Abstrahlung von ICs

Durch Risikoanalysen können über alle Geräte und Anwendungen hinweg die möglichen Angriffe und dazu nötigen Schutzmaßnahmen bewertet, und so ein anwendungsspezifisches Sicherheitskonzept erstellt werden.

Technologien für resiliente Embedded-Systeme und die zugehörige Software

Zentrale Merkmale sind sichere Schlüsselspeicher, eine Umgebung zur sicheren Ausführung von Code sowie eine Implementierung kryptografischer Algorithmen, z.B. als Software-Bibliothek oder über Hardware-Beschleuniger sowie ein Zufallszahlengenerator. Wie z.B. im Projekt Keystone³ gezeigt wurde, ist es mit diesem minimalen Set an Features möglich, grundlegende Sicherheitsfunktionalitäten bereitzustellen. Tiefer im System verankerte sichere Ausführungsumgebungen (Trusted Execution Environments) verbinden eine stärkere Separierung auf Hardware-Ebene und spielen eine immer wichtigere Rolle.

Attestierung ermöglicht, die Eigenschaften des Systems und seine Unversehrtheit gegenüber der auf dem System ausgeführten Software und nach Außen gegenüber Dritten nachzuweisen. Secure Boot und secure Update sind essenziell, um sicherzustellen, dass das System nur authentischen Code ausführt und dieser nicht verändert wurde. Weiter ist secure Update notwendig, um die Sicherheit auch langfristig zu gewährleisten. Neben sicheren Designs sind Tool-basierte Analysen und praktische Penetrationstests eine Voraussetzung, um die Effektivität der eingesetzten Maßnahmen sicherzustellen.

² https://www.bsi.bund.de/DE/Service-Navi/Publikationen/Studien/Hardware-Angriffe/Hardware-Angriffe_node.html

³ <https://keystone-enclave.org/>

Ausblick Forschungsthemen

Um Systeme aufbauend auf der Hardware auch langfristig abzusichern, werden in der Forschungscommunity neue Sicherheitstechnologien erforscht, die über einen Transfer in die Industrie auch nach und nach in Produkten verwendet werden. Einige Beispiele aus der aktuellen Forschung sind:

Post-Quanten-Kryptografie (PQC)

Asymmetrische kryptografische Verfahren wie RSA und ECC bilden die Grundlage unserer heutigen Kommunikation. Die Komplexität der darunter liegenden mathematischen Probleme ist für einen Quantencomputer drastisch reduziert, sodass Kommunikation, die z.B. durch ECC und RSA gesichert wurde, ggfs. durch einen Quantencomputer entschlüsselt werden kann. In den vergangenen 10 Jahren wurden deshalb neue Verfahren entwickelt und werden derzeit durch verschiedene Sicherheitsbehörden, wie NIST, standardisiert⁴. Sind Daten langfristig relevant oder Systeme über Jahrzehnte im Feld, ist es wichtig, jetzt den Wechsel zu PQC-Algorithmen vorzubereiten und zu vollziehen. So wird derzeit z.B. ein Branch der Krypto-Bibliothek Botan im Auftrag des BSI um PQC-Algorithmen erweitert⁵.

Durch Unterschiede in den mathematischen Verfahren, ist es nötig, Wissen und Erfahrung zu PQC zu transferieren und zu nutzen, und andererseits die grundlegenden Unterschiede weiter zu erforschen.



Abbildung 2: Arbeiten an einem Lasermessplatz für Fehlerangriffe auf eingebettete Systeme und darauf ausgeführte kryptografische Implementierungen

⁴ <https://esrc.nist.gov/Projects/post-quantum-cryptography>

⁵ https://www.bsi.bund.de/DE/Themen/Unternehmen-und-Organisationen/Informationen-und-Empfehlungen/Kryptografie/Kryptobibliothek-Botan/kryptobibliothek-botan_node.html

RISC-V

Der offene Befehlssatz RISC-V wurde in den letzten Jahren immer weiterentwickelt, um eine Alternative zu bestehenden Befehlssätzen bereitzustellen⁶. Der weitere Ausbau des Befehlssatzes und des Tool- und Software-Ökosystems wird an vielen Stellen vorangetrieben, um unterstützende Prozessoren in System-on-Chip Systemen, aber auch die eigentlichen Applikationsprozessoren zu ermöglichen. Dies ermöglicht, leichter Anpassungen an Prozessoren durchzuführen und so angepasste Chip zu fertigen.

Open-Source HW

Open-Source Software ist inzwischen ein essenzieller Baustein der IT-Landschaft geworden (z.B. Linux). RISC-V und andere offene Implementierungen stehen an der Schwelle, ähnliche Rollen einzunehmen. Das Hardware-Ökosystem stellt jedoch andere Herausforderungen, sodass sowohl die Technik als auch ihre Nutzung vorangetrieben werden müssen.

Sichere Lieferketten und Vertrauenswürdige Elektronik

Elektronik ist nur sicher, wenn sie frei von unbeabsichtigten Schwachstellen und bewusst eingebrachten Hintertüren ist, und außerdem korrekt gefertigt wurde. In der international vernetzten Lieferkette ist es derzeit unmöglich, diese Vertrauenswürdigkeit herzustellen und nachzuweisen. Initiativen wie der EU Chips Act haben das Ziel, die europäische Souveränität zu stärken und bieten Möglichkeiten für vertrauenswürdiger Lieferketten⁷. Dies setzt aber auch die Verfügbarkeit und technische Reife der eingesetzten Technologien voraus⁸. Dazu sind z.B. neue Vertrauensmerkmale - wie Hardware Fingerprinting -nötig, um einzelne Chips und Geräte über die Lieferkette nachverfolgbar zu machen und ihre Authentizität nachzuweisen. Effiziente Analysetechniken sind erforderlich, um nachzuweisen, dass Geräte frei von Trojanern und Backdoors sind; sowie Tools, um nachzuweisen, dass Tools, Designs und Software frei von Schwachstellen und Hintertüren sind.

Zusammenfassung und Ausblick

Die Sicherheit der Hardware ist Voraussetzung für die Sicherheit von Eingebetteten Systemen, aller darauf ausgeführter Software und der verarbeiteten Daten. Über die letzten Jahre wurden immer mehr Sicherheitsfeatures in Chips und Geräten implementiert, sodass das Thema Sicherheit einen breiteren Stellenwert einnimmt. Wichtig ist, dass die Features korrekt umgesetzt und verwendet werden. Dazu werden konstant neue Sicherheitsfeatures und Analysemethoden entwickelt, die ermöglichen die Sicherheit von eingebetteten Systemen und ihrer Lieferketten zu erhöhen.

Wichtige Trends dabei sind z.B. die Migration zu Post-Quanten Kryptografie, RISC-V Prozessoren und sicherer Hardware.

⁶ <https://riscv.org/>

⁷ <https://digital-strategy.ec.europa.eu/de/policies/european-chips-act>

⁸ <https://www.velektronik.de/en/referenzpapier-vertrauenswuerdige-elektronik/>

Autor

Dr.-Ing. Matthias Hiller leitet die Abteilung Hardware Security am Fraunhofer AISEC in Garching bei München. Vor seiner Tätigkeit am Fraunhofer AISEC studierte Matthias Hiller an den Universitäten Ulm und Portland State University, und arbeitete als Wissenschaftlicher Mitarbeiter an der Technischen Universität München.

In seiner Forschung beschäftigt er sich mit sicheren eingebetteten Systemen und vertrauenswürdiger Elektronik.