

Andreas Amm

B. Sc. · Senior Embedded Softwareentwickler

ASIL-D Specialist · AUTOSAR · KI-gestützte Entwicklung

Seit über 18 Jahren entwickle ich sicherheitskritische Embedded Software für Automotive und Industrie — von der Treiber- und HAL-Schicht bis zur AUTOSAR-Applikation, qualifiziert bis ASIL-D. Mein Fokus: robuster, wartbarer Code auf Mikrocontrollern wie AURIX, STM32 und NXP, eingebettet in CI/CD-gestützte Prozesse. Aktuell setze ich KI-Werkzeuge wie GitHub Copilot und Claude gezielt ein, um Qualität und Geschwindigkeit auf ein neues Level zu heben.

• Verfügbar ab Q3 2026

Remote & Onsite (Bayern / Süddeutschland)

18+

Jahre Erfahrung

7

Grossprojekte

ASIL-D

Höchstes Safety-Level

5+

MCU-Plattformen

Schwerpunkte

Hardwarenahe Softwareentwicklung in C/C++

AUTOSAR Classic & Adaptive

Treiber-, HAL- und Basissoftware-Entwicklung

Funktionale Sicherheit (ISO 26262 ASIL A–D, IEC 61508 SIL2)

Build-Automatisierung & CI/CD

KI-gestützte Entwicklung

Plattformen & Prozessoren

STM32 (U5, F4, H7)

Infineon AURIX TC27x/TC39x

NXP S32K (ARM Cortex)

TI TMS570LS (ARM Cortex-R)

ST SPC570S

Kernkompetenzen

Kommunikationsprotokolle: CAN, LIN, FlexRay, Ethernet, CANopen, UDS, HSS, SPI, I2C, UART

Bootloader- und Flash-Update-Entwicklung

Tool- und Schnittstellenentwicklung in Python und Perl

Modellbasierte Entwicklung mit Matlab/Simulink und dSpace TargetLink

Testautomatisierung und Unit-Testing

Integration, Release- und Konfigurationsmanagement

Agile Entwicklung (SCRUM, Kanban)

Betriebssysteme & Tools

RTOS / EMBEDDED

FreeRTOS, AUTOSAR Classic & Adaptive, QNX, WinCE

CI/CD & BUILD

Jenkins, CMake, Make, SCons

KONFIGURATIONS- & REQUIREMENTS MANAGEMENT

JIRA, Polarion, inStep, Doors, Git, SVN

PROGRAMMIERUMGEBUNGEN

CubelIDE, S32 Design Studio, Code Composer Studio, Eclipse, Visual Studio, VS Code

DEBUG & TEST

Lauterbach Trace32, UDE Debug, CANoe, CANape, CANalyser, EB Tresos, Geny, DaVinci Configurator & Developer

CODE ANALYSE

PC-Lint, QAC, Polyspace, Astree

MODELLIERUNG / SIMULATION

Matlab/Simulink, dSpace TargetLink

KI & ASSISTENZ

GitHub Copilot, Copilot Agents, ChatGPT, Claude

Technologien & Programmiersprachen

PROGRAMMIERSPRACHEN

C, C++, Python, Perl, Java, VBS/VBA

SCHNITTSTELLEN

SPI, I2C, UART, CAN, LIN, Ethernet, HSSL

COMPILER & BUILDSYSTEME

GreenHills, Tasking, gcc/qcc, Visual Studio, Eclipse, Make/CMake, Scons

SAFETY

FuSi ASIL A–D (ISO 26262), SIL2, MISRA C, A SPICE

Ausbildung & Abschlüsse

2012 B.Sc. Angewandte Informatik (2012)

2005 IHK-Ausbildung: Technischer Assistent für Informatik (2005)

Branchenerfahrung

Automotive (z. B. Steuergeräte, E-Mobilität, Batterie-Management-Systeme)

Industrieelektronik (z. B. Encoder, Sensorik)

Einsatzbereiche

Automotive

Industrietechnik

Nutzfahrzeuge

Sprachdialog Systeme

Elektronische Fahrwerkssysteme

HV-Batteriesysteme

Batterie-Management-Systeme (BMS)

Energiemanagement

Electric Commercial Vehicles

Seilzug-/Absolut Encoder

Soft Skills

Teamfähigkeit & Kommunikation auf Augenhöhe

Strukturierte Arbeitsweise und lösungsorientiertes Vorgehen

Erfahrung in agilen und klassischen Entwicklungsmodellen

Datenbanken

Access

SQLite

MySQL

MS SQL Server

AKTUELL

Embedded Softwareentwickler – Batterie Management Systeme (BMS)

Munich Electrification | Februar 2026 – heute | FuSi ISO 26262 ASIL-C

Entwicklung sicherheitskritischer AUTOSAR-Basissoftware für Batterie-Management-Systeme (BMS) im Bereich E-Mobilität. Verantwortung für hardwarenahe Softwarekomponenten auf zwei parallel eingesetzten Controller-Plattformen (NXP S32K und TI TMS570LS) sowie den produktiven Einsatz KI-gestützter Entwicklungs-Workflows im Team.

- ▶ Entwicklung und Integration von AUTOSAR-Classic-Basissoftwaremodulen (BSW/MCAL/RTE) für BMS-Steuergeräte bis ASIL-C
- ▶ Treiber- und HAL-Entwicklung in C/C++ für NXP S32K (ARM Cortex-M) und TI TMS570LS (ARM Cortex-R, Lockstep)
- ▶ Konfiguration und Tooling mit DaVinci Configurator und DaVinci Developer, Anbindung an DBC-/ARXML-basierte Kommunikationsmatrizen
- ▶ Bootloader-Entwicklung inkl. sicherem Flash-Update und UDS-Diagnoseintegration
- ▶ Produktiver Einsatz KI-gestützter Entwicklung (GitHub Copilot Agents, ChatGPT, Claude) für Code-Generierung, Refactoring und Review-Unterstützung
- ▶ Build-Automatisierung, statische Analyse und CI/CD über Jenkins und Bitbucket

C/C++ AUTOSAR S32 Design Studio Code Composer Studio VS Code CANoe DaVinci Configurator & Developer Git (Bitbucket)
JIRA NXP S32K (ARM Cortex) TI TMS570LS (ARM Cortex-R)

Embedded Softwareentwickler Industrieautomatisierung – Seilzug- und Absolut-Encoder

SIKO GmbH | Oktober 2022 – August 2025

Entwicklung hardwarenaher Firmware (Treiber, HAL, Applikationsfunktionen) für sicherheitskritische Seilzug- und Absolut-Encoder in der Industrieautomatisierung. Zielplattformen: 32-bit SPC570S (Automotive-MCU) und STM32 U575. Umsetzung nach IEC 61508 SIL 2 inkl. Test- und Releaseprozess.

- ▶ Treiber- und Funktionsentwicklung für EEPROM, I2C, SPI, UART, ADC sowie Fehlerspeicher-Handling
- ▶ Kommunikationsstack-Integration für CANopen und CAN SAE J1939
- ▶ Umsetzung sicherheitskritischer Funktionen nach IEC 61508 SIL 2 auf SPC570S und STM32 U575
- ▶ Bootloader-Entwicklung mit sicherem Flash-Update-Mechanismus
- ▶ Aufbau und Pflege der CI/CD-Pipelines (Jenkins), Build-Automatisierung und Releasemanagement
- ▶ KI-gestützte Code-Generierung, Refactoring und Unit-Test-Erstellung mit GitHub Copilot Agents

C/C++ CubelIDE (ST) Git (Bitbucket) SVN JIRA Jenkins Python UDE Debugger (Universal Debug Engine via JTAG) Make
FuSi IEC 61508 SIL2 32-bit SPC570S Automotive Microcontroller STM32 U575

Embedded Softwareentwickler AUTOSAR – Electric Commercial Vehicles

Siemens AG April 2019 – Juni 2022 FuSi ISO 26262 bis ASIL-C

AUTOSAR-Basissoftwareentwicklung für elektrische Antriebssteuergeräte in Nutzfahrzeugen (E-Busse und E-LKWs). Fokus auf Bussysteme, Diagnose-Stack und Build-/Releaseprozess auf Infineon AURIX TC27.

- ▶ Entwicklung und Integration von AUTOSAR-Classic-Basissoftwaremodulen für elektrische Antriebssysteme
- ▶ Funktionsentwicklung im Bereich Bussysteme: CAN-/LIN-Kommunikation, PDU-Router und RTE-Anbindung
- ▶ Umsetzung von UDS-Diagnose-Diensten inkl. Fehlerspeicher (DTCs), Coding und Service-Implementierung
- ▶ Integration und Konfiguration mit EB tresos, CANoe, CANape und Vector Test Studio
- ▶ Verantwortung für Build-Automatisierung und Releasemanagement über Jenkins-Pipelines (Multi-Branch)
- ▶ Toolqualifikation und Prozess-Einhaltung gemäß ISO 26262 bis ASIL-C

C/C++

Tasking v4 und v6

CANape

CANoe

VectorTestStudio

EB-Tresos

WinIDEA

AUTOSAR 4

SVN

Make

Jenkins (Pipelines)

Infineon Aurix TC27

Embedded Softwareentwickler AUTOSAR – Elektronische Fahrwerkssysteme

Continental AG Januar 2017 – Dezember 2019 FuSi ISO 26262 bis ASIL-D

Entwicklung von AUTOSAR-Basissoftware für adaptive Fahrwerkssysteme auf der VW-/Audi-Plattform MLBevo. Verantwortung für BUS-Kommunikation, Inter-Microcontroller-Kommunikation und Fahrzeuginbetriebnahme in einem bis ASIL-D qualifizierten Umfeld.

- ▶ Funktionsentwicklung BUS: Kommunikation zwischen FlexRay, SubCAN und RTE inkl. Gateway-Funktionalität
- ▶ Umsetzung von UDS-Diagnosediensten (DTCs, Coding, Routine Control)
- ▶ Entwicklung der Microcontroller-to-Microcontroller-Kommunikation über HSSL inkl. Treiberentwicklung im MCAL
- ▶ Integration und Inbetriebnahme der Funktionen am Fahrzeug (HiL und reales Fahrzeug)
- ▶ Umsetzung nach ISO 26262 bis ASIL-D inklusive MISRA-C-Konformität und statischer Analyse
- ▶ Build-Automatisierung mit Jenkins und SCons, Toolchain auf Basis Tasking für Infineon AURIX TC39

C

Tasking

CANape

CANoe

CANalyzer

Davinci

Geny

EB-Tresos

AUTOSAR 3 und 4 + Adaptive Autosar

Lauterbach Trace32

Scons

Jenkins (Pipelines)

PTC Integrity / SVN / Git

Infineon Aurix TC39

Embedded Softwareentwicklung Energiemanagement – AUTOSAR MLBevo

AUDI AG / BFFT GmbH März 2013 – Dezember 2016

Entwicklung der Diagnosefunktionen für das Niedervolt-Energiemanagement-Steuergerät der VW-/Audi-Plattform MLBevo. Modellbasierte Funktionsentwicklung (Matlab/Simulink, dSpace TargetLink) mit AUTOSAR-Integration und Fahrzeug-Inbetriebnahme.

- ▶ Funktionsentwicklung Diagnose: Messwerte, Fehlerspeicher, Anpassung, Kodierung, EEPROM-Handling, Historienspeicher, Grundeinstellung, Routine Control
- ▶ Modellbasierte Entwicklung mit Matlab/Simulink und dSpace TargetLink, Code-Generierung für AUTOSAR 3
- ▶ Integration der generierten Komponenten in die Steuergeräte-Basissoftware
- ▶ Inbetriebnahme und Absicherung der Funktionen am Fahrzeug (HiL und Erprobungsfahrzeug)
- ▶ Entwicklung von Python-Skripten zur Testautomatisierung und Auswertung
- ▶ Build-Automatisierung mit Make/Jenkins auf Basis Greenhills-Compiler

C/C++

Python

Matlab Simulink / dSpace Targetlink

PTC Integrity / SVN

Lauterbach Trace32

Greenhills

CANape

CANoe

AUTOSAR 3

Make

Jenkins

Embedded Softwareentwicklung Energiemanagement für alternative Antriebe

Audi Electronics Venture (100%-Tochter der AUDI AG) Juni 2012 – März 2013

Modellbasierte Entwicklung von Steuergeräte-Software zur Ansteuerung von 48-V-Verbrauchern in Vorentwicklungs-Prototypen alternativer Antriebskonzepte.

- ▶ Modellbasierte Funktionsentwicklung (Matlab/Simulink, dSpace TargetLink) zur Ansteuerung der 48-V-Verbraucher
- ▶ Integration der generierten Komponenten in AUTOSAR-Basissoftware
- ▶ Inbetriebnahme der Prototypen auf Laborträger und im Versuchsfahrzeug
- ▶ Aufbau der Build-Automatisierung mit Jenkins und Python (Greenhills-Toolchain)

C/C++

Matlab Simulink

dSpace Targetlink

PTC Integrity

Greenhills

CANape

CANoe

Make

AUTOSAR

Embedded Softwareentwicklung sprachbedienbarer Infotainment-Systeme

Elektrobit Automotive GmbH März 2008 – Dezember 2011

Entwicklung und Integration von Sprachdialogsystemen für Automotive-Infotainment-Plattformen auf WindowsCE und QNX/Linux (ARM9). Verantwortung für Funktionsimplementierung im Bereich Sprachausgabe sowie Aufbau und Betrieb der Software-Integrations-, Build- und Test-Infrastruktur.

- ▶ Funktionsimplementierung in C/C++ im Bereich Sprachausgabe (TTS) und Dialogsteuerung
- ▶ Anbindung und Integration von Nuance Recognizer und SVOX TTS Engine
- ▶ Aufsetzen und Pflege der Software-Integrations- und Buildumgebung (Make/CMake, Hudson/CruiseControl)
- ▶ Automatisierung des Buildprozesses sowie Software-Konfigurations- und Releasemanagement (SVN)
- ▶ Testfallerstellung, Testdurchführung, Testautomatisierung und Testauswertung
- ▶ Schulung neuer Mitarbeiter in SW-Integration und SW-Konfigurationsmanagement

WinCE 5.0

QNX 6.x (ARM9)

Visual Studio

Hudson/Cruise Control

Atlassian JIRA

Subversion SVN

Make / CMake

Nuance Recognizer

SVOX TTS Engine