



KONTAKT

+49 160 962 8 3856

WEBITE & Git Bereich

- <https://enrikbregasi.com>
- <https://github.com/riko12>
- <https://www.linkedin.com/in/enrik-bregasi-4990a715/>
- https://www.xing.com/profile/enrik_bregasi



ENRIK BREGASI

Dipl. Ing. Dipl. Programmierer Xilinx SW Designer

PROFIL

Ich bin geboren am 04.09.1061 habe in die letzten 25 Jahren als Freiberufler Software Architekt & Entwickler für Embedded Systeme, in AUTOSAR oder ohne AUTOSAR, CLASIC & ADAPTIVE, Hardware nahe und für die Systeme, PC Windows, Linux, QNX, C, C++, 3D Grafiken und Visualisierung, KI-Entwickler mit eigenen entwickelten C und C++ Module nach Standards von Heute profiliert.

🌟 Ich bringe in deine Software Projekte in der Implementierung, Geschwindigkeit, Sicherheit, Standart Lösung. Meine Erfahrung dein Vorteil.

Veröffentlichtes buch:
 "C++ Reinforcement Learning Agent applied in a chess engine"
amazon.de/dp/B0F7RKD3PM/

Projekte

Eigene Entwicklungen und produkte in KI bereich	2
Software Architekt ASIL-B Continental Villingen-Schwenningen	2
Software Architekt für Sanden International Europe in Bad-Neuheim	2
Software Architekt ASIL-B Continental Frankfurt.....	3
Software Architekt ASIL-D Hella in Lippstadt.....	3
Software Architekt ASIL-D Continental Babbenhausen BMW GEN5.....	3
Software Architekt ASIL-C Valeo Peiker Telematiks	4
Software Architekt ASIL-D generische Architektur Continental Nürnberg.....	4
Software Architekt ASIL-C mit FUSI generisch Hella Lippstadt.....	5
Software Architekt ASIL-C mit FUSI & Sicherheit Continental Markdorf.....	5
Software Ingenieur Magna Electronics Babbenhausen	6
Software Ingenieur Mayser GmbH & Co. KG	6
Software Ingenieur Hydro Building Systems	6
Software Ingenieur Harman Becker Ulm	6
Software Ingenieur Partner Unternehmer Service trace Darmstadt	7
Software Ingenieur Schuler hi-cad Systems Darmstadt	7
Software Ingenieur Institut für Unternehmens Forschung in Fürth/Nürnberg	7
Studium, Ausbildung	8

EIGENE ENTWICKLUNGEN UND PRODUKTE IN KI BEREICH

Kontinuierlich seit 2004

Zusätzliche Arbeiten in Bereich der intuitives KI weiterentwickelt, mein erster "Deep Learning" Modul damals war ein Backpropagation Netz, um die Maus Bewegungen zu lernen.

Heute habe meine eigene KI-Module, teilweise generisches KI, wie, Transformers, Reinforcement Learning und andere KI-Module mit Anwendung in C++ Implementierung dazu. Ein Vorgeschmack kann Man finden in mein erstes Buch `C++AI-Implementierung in ein Schach`. Des Weiteren zu dieser Zeit, sind, diversen Modulen zu Source Code Generieren von KI, Unit Tests generieren von KI, automatisches Debugg von KI, Tests Ausführen von KI, gegen Anforderungen evaluieren, Statische, dynamische Diagrammen die zu Ausführungszeit von der Software selbst erzeugen während der Laufzeit, lokal und im Intranet verteilt. Alle KI-Module sind in C++17, basiert mit der Anlehnung auf PC und Microcontroller Steuer Geräte `Status Q`. Mehrere bekannte Algorithmen wie zum Beispiel `Monte Carlo Baum Search`, `Upper Konfident Bound`, generische `reinforcement learning Agent`, `Deep Learning` durch `Neurale Network` in diverse Kombinationen zum Beispiel `Convolutional`, `Full connected`, RELU und mehr.

Künstliche Intelligenz

- Mehrere Entwicklungen im Bereich KI. Siehe mein Buch

Eingesetzte Technologien, Werkzeuge Methodiken

- Mehrere Entwickler Tools (Eclipse, Visual Studio, Eigene Tools) und Systeme

Systeme

- (Linux, Windows PC, Mikrocontroller von NVIDIA Systeme – Jetson AGX ORIN, Jetson NANO)

Eigene Aufgaben und Rolle in Projekt

- Entwickeln generische Software für Robotik, Industrie, Embedded, AUTOSAR (beide Adaptive, classic)

SOFTWARE ARCHITEKT ASIL-B CONTINENTAL VILLINGEN-SCHWENNINGEN

06.2022 – 10.2023

Aufgabe: Kunden Projekt als leitender SW-Architekt für ein Kunden Projekt TSU (Tachograph Simulation Unit) Von Start bis zu Nominierung. Entwerfen von Komplete Architektur des Projektes basierend auf SW-Anforderungen inklusive SW-Architektur und SW Detail Design, Entwerfen von SW Tests Definition, Interfaces. Tooling: IBM Rhapsody 8.3.1. Generatoren wurden entworfen von mir für generieren der Architektur Dokumente. SW Development Team wurde von mir begleitet und unterstützt. Ein AUTOSAR DET Log Mechanismus wurde von mir konzipiert und entwickelt. Mehrere Möglichkeiten für die Log Ausgaben. Teil 2 als SW-Architekt in Tachograph Plattform teil Genomen. Architektur für mehrere Module wurde entworfen mit Detail design, Interfaces, Tests, Dokumentation generieren. Software Architekt für Sanden International Europe in Bad-Neuheim

Eingesetzte Technologien, Werkzeuge Methodiken

- Git, Version Control, IBM Rhapsody, EB Tresos, IBM DOORS, Agile Prozess mit Sprints, Programmier Sprache C, Standards AUTOSAR classic, ASIL-B Zertifizierung. ECU-Hardware war ein Traveo II von Infineon mit 2Core

Eigene Aufgaben und Rolle in Projekt:

1. Design, Entwicklung der Software Architektur, Detail Design, Unit und Integration Tests, mit den Blick auf Qualität zu Erfüllung der Software Anforderungen.
2. Leiten der der Software Architecture und Development Team, Coaching Development Team um die Architecture umzusetzen, Schulung der Entwickler, Mitwirkung die Software Anforderungen zu gestalten.
3. Plattform Entwicklung. Diverse Module, ADC, Hardware Abstraktion Schicht, vereinfachen der DLT-Modul (Design und Implementierung), Design und Implementierung einen PduR Modul mit zusätzliche Serial Kommunikation Fähigkeit. Architecture Design der Software Update Funktion in 2 Varianten 1 CORE(UNSECURE), 2 CORE(SECURE).
4. Design und Entwicklung von RGB Steuerung Back light Geräte von Texas Instruments (TLC6C5724)

Erfolgreich nominiert

Project Spezifisch: Kommunikation, Leitung und Schulung der Entwickler in unterschiedliche Zeit Zone.

Hardware: Infineon Traveo II

SOFTWARE ARCHITEKT FÜR SANDEN INTERNATIONAL EUROPE IN BAD-NEUHEIM

06.2022 – 10.2022

Aufgabe: Änderungen an das Werkprodukt nach ASPICE 2.5. Erst Version von "Konfiguration Files" und neu Generierung der Applikation Kode der korrigierte Software Implementierung parallele Diagnose. Final Version ohne Fehler und komplette Dokumentation. Erstellen die VECTOR- Konfigurationsdateien aus dem vorhandenen automatisch generierten Quellcode neu. Diagnoseanfragen vom TESTER an die Standheizung. Die CCU leitet die Diagnoseanforderungen der Zusatzheizung über SRP_S1_ACH weiter Botschaft auf H-CAN / V-CAN.Die CCU darf die Diagnosedienste nicht stoppen (Absturz), wenn parallele Diagnosen durchgeführt werden. Die CCU führt parallel zur Zusatzheizung Diagnoseanfragen vom TESTER aus. Tooling: Enterprise Architekt, Vector Davinci Configurator, MPC 560 Compiler, Vector Canoe. Software Architekt ASIL-B Continental Frankfurt.

Eingesetzte Technologien, Werkzeuge Methodiken

- Verctor Informatik Davinci Configurator, Davinci Developer

Eigene Aufgaben und Rolle in Projekt

1. Modifikation der Hardware design, der Software Architektur, Implementierung der Änderung in der Software wurde gemacht und umgesetzt in ein Fertigen bei Kunden umgesetzten Sanden Produkt.

SOFTWARE ARCHITEKT ASIL-B CONTINENTAL FRANKFURT

10.2021 – 03.2022

Aufgabe: Entwicklung eines Treibers für CUS3 Ultraschall Sensor System (Continental Ultrasonic Sensor Gen. 3)

Unterschiedlichen SW-Modulen wurden entwickelt, Coding, Tests. Für alle Module, SW Anforderungen Spezifikation (SRS) und SW Architecture Dokument (SWAD), habe ich entworfen.

SW Module Anforderungen Spezifikation (MRS).

Mehr Details in folgende Meilensteine:

Analyse von Modul-spezifik Inhalten aus SRS / SWAD, SW-Anforderungen wurden entworfen, SW MRS basiert an CUS3 Template. Verbindung SRS (Software Requirement Specification) / SWAD (Software Applikation design) unter Doors NG Tools, Review von MRS-Dokument. Entwurf von SW Module Detail Design Dokument (MDD) Mehr Details in folgende Meilensteine:

Analyse von Modul-spezifik Inhalte SRS / SWAD, Raffinement / Entwurf von SW-Detail design (high Level, Detail Level), SW MDD basierend auf CUS3 Template mit Beschreibung von Module Architecture, Integration Manual, Angeforderte Interfaces und Module Dokumentation in doxygen. Implementieren von Embedded SW Module in C.

Mehr Details in folgende Meilensteine:

Entwurf / Implementierung Funktionen mit zusätzlich das entsprechende doxygen Template. Source Coding, Static Test, Test von Source code in Embedded-Umgebung. GIT als Development Version Kontrolle.

Eingesetzte Technologien, Werkzeuge Methodiken

- Git, Confluence tool, CMake, Programmier Sprache C

Eigene Aufgaben und Rolle in Projekt

1. Detail Design und Entwicklung eines Ultraschall Sensor Modul Applikation hinten im Auto. Entwicklung der MCAL-Schicht und Hardware Abstraktion Schicht in C. Design und Entwerfen der CMake Toolchain um diese Anwendung zu kompilieren. Zusätzlich Entwerfen die notwendigen Software Anforderungen, Architektur, Detail Design, Implementierung der Source code, und Unit Tests.

SOFTWARE ARCHITEKT ASIL-D HELLA IN LIPPSTADT

09.2021 – 05.2022

Aufgabe: AUDI DC DC entwerfen von eine verbundene mit Anforderungen in DOORS von SW-Architektur und Module design konform ASPICE 2.1 mit Rhapsody als Input für Da Vinci Konfigurator und Da Vinci Developer. Projekt AUDI DC/DC 48 V 2.5, Entwicklung der Software-Architektur, Erfüllung aller ISO 26262 Anforderungen bis zu ASIL D (Alle Sicherheitstabellen der ISO, die für die Software-Architektur gelten, müssen berücksichtigt werden), Statische & dynamische Architektur, Timing-Verhalten der sicherheitsrelevanten Pfade,

Einschließlich Rückverfolgbarkeit zu Software-Anforderungen, Behandlung der Interaktion mit der Systemarchitektur, Simulation der Architektur nach ISO 26262, Interaktion mit AUTOSAR-Modellierung und 3rd Party SW,

Zusammenspiel mit detailliertem SW-Design (handschriftlich & Matlab/Simulink/TargetLink-basiert). Tooling: IBM Rhapsody, Akzeptanz-Kriterien: Review bei Hella abgeschlossen (einschließlich Hella-Safety und Hella QA).

Eingesetzte Technologien, Werkzeuge Methodiken

- IBM Rhapsody, DOORS, IBM MKS, FUSI-Standard ISO 26262, Safety Relevantes Projekt, Vector Informatik, Davinci Developer

Eigene Aufgaben und Rolle in Projekt

1. Entwerfen der Software Architektur und Detail Design für ein Licht Anwendung vorne im Auto in AUDI-Projekt in Zusammenarbeit mit FUSI-Manager (26262).

Projekt wurde erfolgreich Nominiert

SOFTWARE ARCHITEKT ASIL-D CONTINENTAL BABBENHAUSEN BMW GEN5

12.2019-06.2020

Aufgabe 1: Design der FPD Architecture, BMW-Begriff für Diagnose Stack auf Embedded Linux System. Es wurde nicht nur die Architektur Design, aber es wurde auch implementiert, von Entwicklung Teams, die Entwicklungen wieder rum wurden von mir geleitet, und unterstützt. Die Architektur wurde entworfen mit Enterprise Architekt.

Aufgabe 2: Entwerfen der Architektur für Security Concept für Instrumenten Cluster von Continental.

Die Umgebung war eine, auf SCRUM Basis Team orientiertes Arbeiten. Programmierung C++.

File System, Versions Verwaltung GIT, aufgelegt auf Conti und BMW online Plattformen. Die Hardware ist eine Komposition von ein Grafik Prozessor und Renesas µController. Die Software: Ein Stark erweitertes AUTOSAR System und Embedded Linux.

Die Architektur wurde entworfen mit Enterprise Architekt.

Eingesetzte Technologien, Werkzeuge Methodiken

- IBM Rhapsody, MKS, Green hills Compiler, Agile Process mit Sprints

Eigene Aufgaben und Rolle in Projekt

1. Entwerfen der Software Architektur, Detail Design, Coach Implementierung von Developers Team in Portugal für "Feature Protocol Dispatcher".
2. Implementierung Anpassung der Software basierend an der Live Leitung von FPD
3. Entwerfen der Architecture, Detail Design, für einen AUTOSAR Security Module und Integration von ein 3.rd Party Verschlüsselung Modul, basierend an TARA-Anforderungen.

Projekt wurde erfolgreich Nominert

SOFTWARE ARCHITEKT ASIL-C VALEO PEIKER TELEMATIKS

10.2017-12.2019

Project: Applikation + Module AUTOSAR Adaptive R17-10

Programmiert Sprachen: C (Renasas Mikro Controller) und C++ unter Linux embedded (NAD-Mikroprozessor 2Kerne)
 Aufgabe: Evaluierung Project in Adaptive AUTOSAR Technologies, Adaptive Services Adaptive Funktion, Adaptive APIs in der Valeo-Peiker Telematik Module. Machbarkeit Forschung Evaluierung. Es wurden Entwicklungs-Methoden und Vorgehensweisen definiert um Valeo-Peiker Telematik Software, basierend auf AUTOSAR Adaptive auf Linux-Embedded Systems entwickeln zu können. Ein gesamt Implementierung Manual wurde erzeugt. Auch ein Valeo-Peiker Projekt wurde zusammengestellt als

Vorentwicklung-Projekt für zukünftige Valeo-Peiker Telematik-Applikationen basierend auf AUTOSAR Adaptive mit COM, DIAG, ARA-API/EXEC, REST, APD-Services und Funktionen. Als Build Umgebung wurde YOCTO Project benutzt um das Sourcen des Projektes, inklusive Embedded Linux System (C++) zu bauen. Für die SRC, AUTOSAR Generatoren wurden angewendet (vorgegeben von AUTOSAR Konsortium). Ab 07.2018-12.2019 Software Engineered, Software Architekt für Valeo Peiker Telematik, Projekt: RNMA (Renault Nissan Mitsubishi Alliance) IVC1 und IVC2, Components: FOTA (Firmware Over The Air), SomelP Modem internal Status und ENDS (Ethernet Netzwerk Design Spezifikation) mit SomelP, Dolp, Telematik + Automotive Projekt. Aufgaben:

Software Architektur für FOTA abhängig von RNMA-Anforderungen. Service Verantwortlicher für "Modem-Status Management" mit SomelP über Ethernet. Service Verantwortlicher für ENDS-Komponente als Lead verantwortlich für Design und Implementierung abhängig von RNMA-Anforderungen. Operating System: Embedded-Linux für Telematik-ECU, OSEC-AUTOSAR 4.2 für den MCU, Hardware: NAD 9x28 Qualcomm als Telematik-TCU und Renesas RH850 F1KM-S4 MCU µC. Bus und Communication: CAN-V, CAN-M, USB, ETH, SPI (RH850<->NAD), WLAN, LAN, BT Tooling: Enterprise Architekt (v.13), DOORS, Core-ALM (Version-Verwaltung, Dokumentation, Monitoring)

Eingesetzte Technologien, Werkzeuge Methodiken

- Enterprise Architekt für die Architektur, Koordination mit internationalem Team in Rumänien, Tschechei, Deutschland

Eigene Aufgaben und Rolle in Projekt

1. Entwerfen der Software Architecture für Telematik Module (Valeo-Peiker) AUTSAR classic
2. Migrieren das Valeo Peiker Telematik Module Projekt von AUTOSAR Classic in AUTOSAR Adaptive in Version R2017.

AUTOSAR Classic Version der Software wurde Nominert

SOFTWARE ARCHITEKT ASIL-D GENERISCHE ARCHITEKTUR CONTINENTAL NÜRNBERG

10.2016-12.2017

Automotive Projekt: BR223 (eSuspension- Daimler)

Safety: ASIL -C, Aufgaben: Erstellung und Mitwirkung der Software Architektur für Data Model/Kommunikation zwischen Komponenten durch RTE, BSW + SWC. Komponente für Timing, Konfiguration für SW Memory Mappings, Komponenten und Feature List Zusammenstellung, Hardware: Aurix TC 27x (Infineon) 1xFlexRay,

4xPSI5 v2.1, 4xPSI5 v1.3, 1xADC für Druck Messung, 4xCAN, Operating System: OSEK-OS, AUTOSAR 3.2, 4.1-4.2, Tools: Rhapsody 8.3, DOORS 9.5, PTC Integrity Client 10 (MKS). Versionierung: PTC Integrity Client 10 als Version Kontrolle und Dokument Verwaltung. Bus: CAN-Bus, FlexRay, SPI, PWM für Kontrolle und abfragen der Sensor Werte (aktive, passive).

Ab 04.2017 bis 12.2017 Software Ingenieur, Software Architekt für generische Architektur bei Continental.

Automotive Projekt: 3KPlus (Porsche Panamera) Safety: ASIL-D, AUTOSAR 3.2.2 Aufgaben: Architecture Rhapsody Model BSW and SWCs, Analyse von Data Kommunikation zwischen Komponente durch die RTE Timing-Konfiguration, SW Memory Mappings, SW-Komponente und feature List, Analyse Stakeholder Anforderungen für Diagnose Entwicklung der Architecture für ERH (Error Handler ASIL-D) Entwicklung der Architecture für SysStateHdl (System State Handler ASIL-D) Entwicklung der Architecture für Odm mit AUTOSAR v3.2.2 (Operation Data Monitor QM) Hardware: Aurix TC297 (Infineon 8MB, 300MHz) HAWS (Hydraulische Roll Stabilisation (Hydraulics Wanken Stabilizer)), 1xFlexray, 8xPSI5 v1.3, 1xCAN, 4xSENT, 4xPWM, 2xLin, 6x Air Suspension Valve für spring rate ctrl, 1x Ambient Valve, 4x Damper Valve für vertikal Dynamik Control Tools:

IBM Rhapsody 8.0.4, Enterprise Architekt v10, IMS-Versionierung, Development Tools CATCH Control & CESSAR, Vector Candela Studio 8.5 für die Diagnose, Inchron für Timing Simulationen und log Analysen, AUTOSAR Builder 2017 x64 für Entwerfen von AUTOSAR Konfiguration Dateien verwendet für die source Generierung.

Eingesetzte Technologien, Werkzeuge Methodiken

- IBM Rhapsody, typisches V-Model Prozess, Inchron tool.

Eigene Aufgaben und Rolle in Projekt

1. Erstellung und Mitwirkung der bereits Bestehende Software Architektur für Data Model/Kommunikation zwischen Komponenten durch RTE, BSW + SWC.

Projekt wurde Nominiert

SOFTWARE ARCHITEKT ASIL-C MIT FUSI GENERISCH HELLA LIPPSTADT

06.2015-08.2016

Design, Entwicklung der Generische Software Architektur mit Funktionale Sicherheit eines Licht Steuergerätes.

Aufgaben: Erstellung und Mitwirkung generischer Architektur für WDG, Renesas F1M/F1H auf AUTOSAR 4.x, basierend auf existierende Architektur für AUTOSAR 3.2 mit FUSI. System Behavioral: Watchdog Konzeption (External Watchdog on SBC), (Cyclic-Wakeup) und Konzepte über Multicore-Umgebung. Erarbeitung von Konzepten (System Integrity, SSI, System Behavioral) Adaption/Aktualisierung von AUTOSAR 3.2 nach AUTOSAR 4.x, Erstellung und Mitwirkung einer generischen Architektur für AUTOSAR 4.x im Bereich SSI-1 (RamTst/FlashTst, ECC, DataROM, MPU) mit FUSI und Konzepte über Multicore-Umgebung. Erstellung und Mitwirkung einer generischen, Architektur für AUTOSAR 4.x im Bereich SSI-2 (CPU-Core, INT/Exception, DMA, µC Mode-Manager) mit FUSI und Konzepte über Multicore-Umgebung. Erstellung und Mitwirkung einer generischen, Architektur für AUTOSAR 4.x im Bereich SSI-3 (Error Handler, PWR-Supply, Clock (PLL) mit FUSI und Konzepte über Multicore Umgebung. Hardware: Mikrokontroller RH 850/F1M/F1H/F1K von Renesas, Microcontroller AURIX - Infineon.

Technology: ISO 26262 Norm, "Road Vehicle – Funktionale Sicherheit"

Operating System: OSEK-OS, AUTOSAR 3.2, 4.x Tooling: Rhapsody 8.3, DOORS 9.5, PTC Integrity Client 10 (MKS) Versionierung: PTC Integrity Client 10 (früher MKS) als Version Kontrolle und Dokument Verwaltung. Bus: CAN-Bus, LIN-Bus. Die Ergebnisse der Aufgaben wurden mit Berücksichtigung auf SEoC, in DOORS eingetragen und dementsprechend verlinkt, um nachfolgend die Auswertung ASIL-B Klassifizierung erreichen zu können.

Eingesetzte Technologien, Werkzeuge Methodiken

- IBM Rhapsody, DOORS, IBM MKS, FUSI-Standard ISO 26262, Funktionale Sicherheit Relevantes Projekt, Vector Informatik, Davinci Developer

Eigene Aufgaben und Rolle in Projekt

2. Entwerfen der Generische Software Architektur und Detail Design für ein Licht Anwendung vorne im Auto in AUDI-Projekt in Zusammenarbeit mit FUSI-Manager (26262).

Project wurde erfolgreich nominiert

SOFTWARE ARCHITEKT ASIL-C MIT FUSI & SICHERHEIT CONTINENTAL MARKDORF

09.2014-06.2015

Design, Entwicklung und Einführung der Software für die Funktionale Sicherheit eines Steuergerätes in ein AUDI.

Aufgaben: Architektur, Design, Entwicklung und Programmierung, Unit und Integrationstest für die folgende Technische Sicherheit relevante ASIL-B Module: E2E, Safe RTE, MPU („Memory Protektion Unit“ mit Hardware Unterstützung) und MPU-Treiber, Fsm (Functional Safety Manager SW), Safe WatchDog, Basierend auf AUTOSAR V4.0 Spezifikation. Hardware: Mikrokontroller RH 850/F1L und RH850/F1M von Renesas.

Technologien: ISO 26262 Norm, "Road Vehicle, Funktional Safety" Programmiersprachen: C Operating System: OSEK-OS Tools: Eclipse Entwicklungsumgebung für die Programmierung, Cesar und Spark Konfiguration und Generator Tool. Green Hills Compiler. SWATT Tool für Unit Tests und Test Spezifikation. IBM Rational Rhapsody in Version V8.6.0 für die Erstellung von Architektur und Design. WinIDEA Version 9.12.224 als Debugger/Emulator.

Rhapsody- Gateway für den Export von Architektur nach DOORS und Trace Ability. DOORS V. 9.6.3 für Software Architektur und Software Detail Design Dokumentierung. Canoe V. 8.2 für die Untersuchungen, Überprüfungen der Funktionalität und Rest- Bussimulation. Versionierung: IMS (MKS) System als Version Kontrolle und Source code Verwaltung. Bus: CAN-Bus Die Aufgaben wurden orientiert und gelöst konform Funktionale Sicherheitsmaßnahmen, beschrieben in Technical Safety Concept (TSC in DOORS) konform Pflichten-Heft in DOORS (Anforderungen). Die Ergebnisse der Aufgaben wurden in Safety Case Dokument eingetragen um nachfolgend die Auswertung und ASIL-B Klassifizierung erreichen zu können. Die Software wurde in Mai 2015 erfolgreich und integriert und an AUDI übergeben.

Eingesetzte Technologien, Werkzeuge Methodiken

- IBM Rhapsody, DOORS, IBM MKS, FUSI-Standard ISO 26262, Safety Relevantes Projekt

Eigene Aufgaben und Rolle in Projekt

1. Entwerfen der Generische Software Architektur und Detail Design für ein Licht Anwendung vorne im Auto in AUDI-Lichter hinten Projekt in Zusammenarbeit mit FUSI-Manager (26262).
2. Entwicklung der Architektur, Detail Design, Implementierung MPU (Memory Protection Unit) basierend an ISO 26262. Zusätzlich Entwicklung der Unit Tests dazu und entwerfen und Implementierung der Injektion Tests in MPU von außen (Hardware Beteiligt in der Implementierung SPI-Leitung, SPI externe Geräte um in die MPU einzudringen und Kompromittieren).

Project wurde erfolgreich nominiert

SOFTWARE INGENIEUR MAGNA ELECTRONICS BABBENHAUSEN

06.2012-03.2014

Softwareentwicklung der Applikation, RTE-Schicht, Treiber Schicht für Anwendung Video Sorraund System mit Multikamera Funktion für FORD. Programmiersprachen: Standard C, auch mit C99 Erweiterungsdiaklekt für MPC56... Serie Mikrokontroller, mit Green Hills Compiler. Aufgaben: Entwicklung von Diagnostik Modul und Steuerung durch Ausführen von Routinen, Voltage Range Monitor Modul, Trailer Aid Control Modul, Trailer Aid Backup Modul, mit Entwicklung von FPGA-Manager, SPI-Modul, Signal Manager, Entwicklung von Salbstest Modul und andere. AUTOSAR 3.2 wurde als Basis Voraussetzung vorgezogen und die Entwicklungen der Module wurden so gerichtet. Des Weiteren Debugg Aufgaben, viele Analysen, Fehler Behebung und Qualitätsverbesserung wurden durchgeführt. Zahlreiche FORD-Anforderungen wurden analysiert, Konflikte wurden festgestellt, Änderungen wurden vorgeschlagen und dann Anforderungskonflikte wurden durch Softwareimplementierungen gelöst. In der Architektur Phase in V Model, habe ich sehr oft mit FORD in Meetings und Analysen teilgenommen. Mehrere Oszilloskop Signal-Analysen wurden ausgeführt und dokumentiert. In Dezember 2012, ich habe eine Ausbildung für Zynq Systeme bei Xilinx absolviert als ZYNQ Software Designer. Tools: Eklipse Entwicklungsumgebung mit C Compiler von Green Hills, WinIDEA wurde zu debugg Zwecke verwendet. Canoe, Vektor Informatik 8.0 für die Restbussimulation CAN, Canoe 8.1 mit LIN-Option. Diagnostik Engineering Tool von FORD. Video Grubber Software, MKS-System als Version Kontrolle und Source Kode Verwaltung.

Eingesetzte Technologien, Werkzeuge Methodiken

- Eclipse, C für die Entwicklung, Xilinx Toolchain für die FPGA-Konfiguration, Green Hills Compiler.

Eigene Aufgaben und Rolle in Projekt

1. Entwicklung von Software Module wie FPGA-Manager, SPI MANAGER, Diagnose Stack, Wurde Manuel implementier und nicht generiert.
2. Entwicklung der TAD (Trailer Aid System) als SWC-Applikation auf der RTE-Schicht.
3. Als Haupt Entwickler hatte ich auch die Aufgabe bei Kunden in den USA (Ford) die Software in AUTO zu testen, konfigurieren zu integrieren

Project wurde erfolgreich nominiert

SOFTWARE INGENIEUR MAYSER GMBH & CO. KG

01.2012-05.2012

Softwareentwicklung Sensor Applikation für power Gate System in General Motors Autos. Aufgabe: Entwicklung der Applikation, basierend auf das HALL-Prinzip, Steuerung, LIN-Treiber mit EEPROM Speicher Funktion. Entwicklung der LIN J2602 Treiber, Rest Bus Simulation für Steuerung, Protokollierung von Sensor Verhalten und Eigenschaften.

Eingesetzte Technologien, Werkzeuge Methodiken

- Programmiersprachen: Standard C für MSP430 Mikrokontroller, IAR-Entwicklungsumgebung für die Software, Canoe, Vektor Informatik und CAPL-Programmierung für Canoe und Canister.

Eigene Aufgaben und Rolle in Projekt

1. Entwicklung der ECU-Software basierend auf LIN Protocol für ein Ford Projekt, um eine Motorisierte Kofferraum zu steuern.

SOFTWARE INGENIEUR HYDRO BUILDING SYSTEMS

11.2009-01.2012

Softwareentwicklung in 3D Bereich. C++ Weiterentwicklung der WICTOP Software. Analyse, Design, Konzeption, Entwicklung und Test bis zur Einführung bei Kunden. Aufgabe: Dynamische Dimensionierung, Intelligente Artikel. Entwicklung eigener STEP File Importer in WICTOP Interner Format und Rendern auf 2D(GDI) und 3D(OpenGL).

Programmiersprachen: Windows Forms in C++ und C#

Database: Fast Objects von Versandt-Version 11

Technologien: OpenGL auf Windows-Forms und MFC-Forms, Open-Cascade wurde verwendet als Framework zu Unterstützung für den Importer. Plattform: PC mit Windows 7, Windows XP.

Eingesetzte Technologien, Werkzeuge Methodiken

- Windows System und Visual Studio

Eigene Aufgaben und Rolle in Projekt

1. Softwareentwicklung in 3D Bereich. C++ Weiterentwicklung der WICTOP Software

Projekt war bereit bei der Kunde

SOFTWARE INGENIEUR HARMAN BECKER ULM

02.2007-04.2009

Softwareentwicklung in Plattform Infotainment Bereich C++.

Ich habe erfolgreich entwickelt eine "Data Base Stack Server" basiert auf SQLite.

Laden und Speicher und laden von Data Lists, in und von Head Unit und Mobile Phone.

Entwicklung der A2DP (AVRCP V1.0-V1.4)

Einen Kontroller zwischen Bluetooth Service und QNX Multimedia Engine.

Mikroprozessor was SH4, ARM, WINDOWS, ARIS.

Betriebssystem: QNX V4. Kunde: BMW, Audi, VW, Porsche, Mercedes in ihre Infotainments Systeme.

10.2008-04.2009 Auch habe ich erfolgreich entwickelt in Plattform eine Multimedia Engine in QNX verantwortlich für die Kommunikation zwischen QNX- Multimedia Engine und mobile Phone verbunden durch Bluetooth.

Eingesetzte Technologien, Werkzeuge Methodiken

- Eclipse C++, MKS, V-Model Prozess

Eigene Aufgaben und Rolle in Projekt

1. Entwicklung der Diagnose Stack als SWC-Anwendung unter QNX-Plattform für das Head Unit.
2. Entwicklung der Multimedia Manager in der QNX-Plattform und Kommunikation Stacks

SOFTWARE INGENIEUR PARTNER UNTERNEHMER SERVICE TRACE DARMSTADT

09.2004-02.2007

Web Server Design und Entwicklung, Windows, Web Client Design und Entwicklung (Plattform unabhängig) Service und Objekt Orientiert Reporting System: Crystal Reports und Microsoft Reporting Softwareentwicklung von Service trace in Darmstadt.

Aufgabe: Entwicklung von Client-Server Softwaresystem zu optimieren der IT- Infrastruktur und Performance

Technologien: Web Server Design und Entwicklung, Windows Server, Web Client Design und Entwicklung (Plattform unabhängig), Service und Objekt Orientiert. Einführung bei Kunden: TSystems- Bamberg, NEXIO-Mainz, Entega-Darmstadt, usw. Design Tools: Visio- UML, Enterprise Architekt 4- UML. Entwicklung Tools: MS Visual Studio 2003, 2005.

Programmierung: C++, C#, ASP.NET, SQL (2003, 2005), WCF, WPF, XML-Patterns, JavaScript und Ajax (MS) Komponenten.

System: IIS, .NET, MSSQL- Server, TCP/IP, XML. AutoIT.

Eingesetzte Technologien, Werkzeuge Methodiken

- Viele Werkzeuge in unterschiedliche Programmierer Sprachen, SQL und weitere Frameworks

Eigene Aufgaben und Rolle in Projekt

1. Entwickeln eines Kontrolle System zum Beobachten, Überwachen, Reporten an Manager von Netzwerk Aktivitäten.

SOFTWARE INGENIEUR SCHULER HI-CAD SYSTEMS DARMSTADT

05.2000-12.2003

Aufgabe: Konzeption, Design, Entwicklung, Testen, Einsetzen und Betreuen von Softwarebausteinen unter Windows im Rahmen der Schuler hi-cad Systems-Gesamtlösung.

Eigenständige Erstellung von Schnittstellenmodulen für kaufmännische Anwendungsprogramme unter Windows.

Analyse, Synthese und Implementierung der

Entwicklung von Projekten für den Datenaustausch zwischen mehreren Softwareprodukten und Softwarehäusern.

Inbetriebnahme beim Kunden.

Testen, Einschulung von Anwendern und Administratoren.

Entwicklung von Komponenten im Bereich B2B und Einsetzen von XML-Dokumenten als Medium für Datenaustausch. Design und Implementierung der Windows Oberfläche im Bereich Windows, zum Beispiel Registry, Multithreading, MSMQ-Access (MDB), DB, MSSQL.

Analyse, Konzeption, Entwicklung eines Client-Server-Systems.

Einführung von BMECAT als Medium in Projekten für den Datenaustausch.

Entwicklung von Programmiermodulen zur Datenbankkonvertierung nach EPL und PPS.

Forschung nach Lösungen und Umstellungen einer Anwendung, Umstieg in das Internet. Programme: Delphi, C++,

JavaScript, XML, MSSQLServer7, VB-Script, HTML, DHTML, ACCESS Datenbank: MSSQL Server7, MS DBASE, Paradox, Text DB.

Eingesetzte Technologien, Werkzeuge Methodiken

- Visual Studio, mehrere Tools

Eigene Aufgaben und Rolle in Projekt

1. Software Entwickler für HI-CAD Anwendung

SOFTWARE INGENIEUR INSTITUT FÜR UNTERNEHMENS FORSCHUNG IN FÜRTH/NÜRNBERG

08.1998-05.2000

Aufgabe: Konzeption, Analyse, Design, Implementierung, Entwicklung, Testen, Einführung, Einschulung, Betreuung der Anwendung, Terminplanung für Therapeuten, Ärzte, Patienten und Räume für deutsche Krankenhäuser.

Dialoge zur Steuerung der Anwendung Terminplanung.

Projekt-Aufgaben:

Datenübermittlung Konzeption, Analyse, Design und Implementierung der Entwicklung, Testen, Einführung, Einschulung, Betreuung der Anwendung

Datenübermittlung und Abrechnung nach § 302 SGB V für deutsche Krankenhäuser und der Anwendung

Datenübermittlung und Abrechnung nach § 301 SGB V für deutsche Krankenhäuser. Design und Implementierung der Entwicklung, von Komponenten und Oberflächen zur Steuerung der Anwendung „Datenübermittlung Krankenkasse-Krankenhaus.

Design und Implementierung der Entwicklung von Tools und Modulen zur Stammdatenerfassung.

Archivierung von Datenbeständen im Rahmen der IFU-Gesamtlösung.

Anpassung der Programme und Ausarbeitung, um die Zulassung der Software für den bundesweiten Einsatz zu erhalten.

Betriebssysteme: Windows 98, Windows NT

Programmierung: Delphi, C++

Datenbanken: BTRIEVE 4. Datenaustausch mit Text Format.

Eingesetzte Technologien, Werkzeuge Methodiken

- Delphi

Eigene Aufgaben und Rolle in Projekt

1. Dialog Entwickler für Health System in Krankenhaus.

STUDIUM, AUSBILDUNG

2012	Ausbildung Zynq -7000 Software Designer
1994	Studium Programmierer Darmstadt. Diplomarbeit über eine Fallstudie für Dynamische Datenstrukturen Textverarbeitung, Grafiken und Funktionen.
1992-1994	Studium Informatiker Hagen (Unterbrochen 2 Jahre)
1981-1986	Studium, Abschluss Dipl. Ing. für Transport und Landwirtschaft Maschinen in Mechanik und Electrotechnik-Tirana / Albanien