DIPL. ING. SVEN BRANDAU

SENIOR SOFTWARE ARCHITECT/DEVELOPER

I am a graduate engineer in electrical engineering with over **25 years** of experience in the development of complex embedded systems. My focus is on the architecture, implementation, and integration of solutions based on Embedded Linux, Bare-Metal systems, and RTOS.

I manage projects holistically – from conceptual design to driver and application development, all the way through to testing and system integration. I work cross-technologically with platforms such as NXP i.MX, STM32, TI OMAP, and Nordic nRF. Understanding the underlying hardware is a given for me.

What is especially important to me is a clean, maintainable system architecture – whether it's energy-efficient low-power applications, real-time systems, or safety-critical embedded solutions. I work in a structured and independent manner and have extensive experience in remote project management.

CURRENT AVAILABILITY

I am currently available for new projects.

CONTACT

```
Sven Brandau
Rigaer Str. 18E
10247 Berlin
Germany

Email: info@brandau.biz
Tel: +49 1739960100

VAT ID: DE237816735
```

SKILLS

Architecture

- Embedded Linux

- Buildroot
- Yocto
- Bare Metal
- Event Driven
- State Machine

Programming Languages

- C/C++
- C++
- C
- C#
- Python
- Java
- Javascript
- Unix Shell Script
- Linker Scripts
- Perl
- Assembler
- ASN.1
- Visual Basic
- Matlab
- Pascal

Operating Systems

- Embedded Linux
- FreeRTOS
- VxWorks
- Nucleus
- Symbian
- Unix
- Microsoft Windows

Tools

- Git
- CMake
- Make
- J-Link
- Wireshark
- Enterprise Architect
- Doors
- SVN
- ClearCase

Libraries

- Qt
- Boost

- Protobufs
- Asterisk

Hardware Interfaces

- DMA
- PWM
- GPIO
- I²C
- SPI
- UART
- RS232
- RS485
- Modbus
- OneWire
- CAN
- LVDS
- MIPI DSI

CPU/MCU

- NXP iMX8MPlus
- NXP iMX8Nano
- NXP iMX7ULP
- NXP iMX6
- NXP RT1062
- NXP LPC43xx
- NXP LPC1347
- STM32L5
- STM32L4
- STM32WB
- STM32U3
- STM32F3
- STM32F4
- Microchip SAM70
- Microchip SAML21
- Nordic nRF9160
- Nordic nRF52850
- Renesas RZ/G2L
- TI OMAP3
- PowerPC 603e
- Cirrus EP9301
- Zilog Z80

Technologies

- CAT-M1
- NB-IoT
- AWS
- Bluetooth
- Wifi

- MPEG-4
- H.264
- MPEG-2
- AAC
- AMR
- AMR-WB
- OMA
- Device Tree
- IPTV
- Autosar
- EGPRS
- Linux Driver
- Linux Kernel

Network Protocols

- IP
- TCP
- SDP
- UDP
- Ethernet
- 3GPP
- RTP
- SOAP
- SIP
- IMS
- PoC
- MGCP
- SigComp
- 3GPP IMS

PROJECTS

Embedded Linux for a Custom Platform

Duration: March 2013 - August 2025
Position: Senior Software Architect/Developer

- **Industry:** Defense
- **Role:** System Design, Architecture, Implementation, Configuration
    - Setup, customization, and configuration of the system (TI OMAP3 / NXP i.MX6 / i.MX8)
    - Driver implementation
    - Design and implementation of the application
    - **Details:**
        - Setup of build environment (gcc for ARM)
        - Assembling the appropriate components for the Linux system:
            - Bootloader: U-Boot + X-Loader / U-Boot MLO + Secure Boot
            - Kernel: 3.0.x to 6.x + Device Tree
            - RootFS: Buildroot, Qt5

- Custom hardware adjustments in the bootloader:
  - Hardware initialization: SDRAM, NAND, GPIO, LCD, ...
  - System boot from: MMC, NAND, NFS
  - Implement splash screen
- Kernel adaptations:
  - CPU frequency scaling (dynamic switching based on profile)
  - Integration of LCD display drivers (Parallel, LVDS, MIPI DSI)
  - Splash screen support
  - Power management with external IC
  - Audio support: Configure drivers for TI OMAP3
  - Custom USB device driver implementation:
    - Streaming data via DMA to the internal DSP of the TI processor
    - Implementation of custom algorithms on the internal DSP
  - Integration of WLAN drivers (RTL8188 and others)
- RootFS adaptations:
  - Configuration of NAND filesystem: UBI-FS, ECC
  - Network configuration: `ifplugd`, DHCP, DNS, static IP, WLAN (`wpa_supplicant`)
  - Hotplug system with MDEV: USB stick automount, FTDI serial-USB, USB WLAN drivers
  - Audio support: ALSA system testing and integration
- Full system features:
  - Implementation of an update mechanism:
    - Dual-boot system
    - Encryption of update package (AES256, SHA256)
    - Update of various sub-components: Detectors, Atmel ATMega via SPI
    - Update controlled via application
    - Update controlled via integrated web server (WebSockets)
- Configuration of userspace drivers for SPI, I²C, GPIO
- RTC and USB gadget drivers
- Connectivity via Bluetooth and WiFi
- Application development (Qt5, C++1x/2x)
- Build server: Jenkins
- Version control: Git
- Bug tracking: Redmine
- Remote work

- **Technologies:**
  - Embedded Linux
  - Device Tree
  - C/C++
  - Bare Metal
  - Buildroot
  - Yocto
  - Qt
  - Git
  - Linker Scripts
  - LVDS
  - MIPI DSI
  - TI OMAP3
  - NXP iMX8MPlus
  - NXP iMX6
  - NXP iMX8Nano
  - NXP iMX7ULP

- ◦ Microchip SAM70
- ◦ Protobufs
- ◦ Wireshark
- ◦ J-Link
- ◦ Bluetooth
- ◦ Wifi
- ◦ CMake
- ◦ Python
- ◦ Javascript
- ◦ Unix Shell Script
- ◦ Matlab
- ◦ Event Driven
- ◦ State Machine
- ◦ Microsoft Windows
- ◦ Unix

Embedded Development for Mass Spectrometer II

Duration: October 2014 - March 2025
Position: Senior Software Architect/Developer

- **Industry:** Electrical Engineering
- **Role:** Architecture / Implementation / Integration
  - ◦ Bare-metal application for NXP LPC43xx and NXP RT1062 SoCs
  - ◦ Software: FreeRTOS, LittleFS, CycloneTCP
  - ◦ IDE: MCUXpresso, C/C++ 20, Git
  - ◦ Hardware: NXP LPC43xx, NXP LPC1347, NXP RT1062
  - ◦ Integration of all external hardware via SPI, I²C, OneWire, UART, GPIOs; FPGA via Memory Controller
  - ◦ Remote work
- **Technologies:**
  - ◦ FreeRTOS
  - ◦ C/C++
  - ◦ Bare Metal
  - ◦ NXP RT1062
  - ◦ NXP LPC43xx
  - ◦ NXP LPC1347
  - ◦ Git
  - ◦ Linker Scripts
  - ◦ SPI
  - ◦ I²C
  - ◦ OneWire
  - ◦ UART
  - ◦ GPIO
  - ◦ J-Link
  - ◦ Protobufs
  - ◦ Wireshark
  - ◦ CMake
  - ◦ Python
  - ◦ RS485
  - ◦ Modbus
  - ◦ IP

- TCP
- Ethernet
- Unix Shell Script
- Matlab
- Event Driven
- State Machine
- Microsoft Windows
- Unix

Embedded Linux: Adjustment Display Driver

Duration: June 2023 - June 2023
Position: Senior Software Developer

- **Industry:** Electrical Engineering
- **Role:** Integration
    - Adjustment of the Display Driver
    - Implement Device Tree
    - Platform: Renesas RZ/G2L
    - OS: Yocto
- **Technologies:**
    - Embedded Linux
    - Device Tree
    - Yocto
    - Git
    - Renesas RZ/G2L
    - Unix Shell Script
    - Unix

Embedded Development in the Field of Low Power

Duration: October 2022 - June 2023
Position: Senior Software Architect/Developer

- **Industry:** Electrical Engineering
- **Role:** Architecture / Implementation / Integration
    - Bare-metal application for an STM32L5 MCU
    - QtCreator, C/C++ 20, Git
    - Interrupt-driven Low Power State Machine (power consumption in the µA range)
    - Implementation of required hardware units: DMA, PWM, GPIOs
    - Remote work
- **Technologies:**
    - FreeRTOS
    - C/C++
    - Bare Metal
    - Linker Scripts
    - Git
    - DMA
    - PWM
    - GPIO

- ◦ UART
- ◦ STM32L5
- ◦ STM32U3
- ◦ STM32L4
- ◦ STM32WB
- ◦ STM32F3
- ◦ STM32F4
- ◦ Microchip SAML21
- ◦ J-Link
- ◦ CMake
- ◦ Event Driven
- ◦ State Machine
- ◦ Microsoft Windows

NB-IoT Embedded Development

Duration: April 2018 - October 2021
Position: Senior Software Architect/Developer

- **Industry:** Electrical Engineering
- **Role:** Software Architect/Developer
  - ◦ Prototype development in the field of NB-IoT / CAT-M1
  - ◦ Software: QtCreator, GCC Toolchain, CMake, J-Link GDB, Git
  - ◦ Hardware: Nordic nRF9160 / Nordic nRF52850
  - ◦ Integration of sensors and PMIC via SPI/I²C
  - ◦ Cellular modem integration
  - ◦ Complete bare-metal application development: bootloader, linker scripts, C++ startup, Secure/Non-Secure Cortex-M
  - ◦ Connection to Amazon AWS IoT Core
- **Technologies:**
  - ◦ NB-IoT
  - ◦ CAT-M1
  - ◦ AWS
  - ◦ Linker Scripts
  - ◦ Bare Metal
  - ◦ Git
  - ◦ SPI
  - ◦ I²C
  - ◦ Nordic nRF9160
  - ◦ Nordic nRF52850
  - ◦ J-Link
  - ◦ CMake
  - ◦ Event Driven
  - ◦ Microsoft Windows

Audio Driver Development for Embedded Linux

Duration: January 2014 - March 2014
Position: Senior Software Developer

- **Industry:** Consumer Electronics
- **Role:** Implementation
  - Development of the audio driver for the STA350 audio chip
    https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git/commit/sound?id=2439ea1f0f8f4cc98dfae0d1cd5ba88f6c3ee9ad
  - Linux: Upstream Linux with Device Tree
  - Remote work
- **Technologies:**
  - Embedded Linux
  - Device Tree
  - Linux Driver
  - C
  - Linux Kernel
  - Git
  - Unix Shell Script
  - Unix

Crypto Device Development

Duration: October 2012 - February 2013
Position: Requirements Engineer

- **Industry:** Security Sector
- **Role:** System Design, Architecture
  - Development of platform software for an embedded system
  - Creation of use cases
  - Requirements engineering
  - Architecture development
  - Description of use cases using DSL (Domain Specific Language)
  - Hardware platform: XILINX Zynq SoC
- **Technologies:**
  - Embedded Linux
  - Doors
  - Enterprise Architect

Embedded Development for Mass Spectrometer I

Duration: April 2012 - September 2012
Position: Senior Software Developer

- **Industry:** Electrical Engineering
- **Role:** Architecture / Implementation / Integration
  - New software development for a controller board
  - Control of various hardware components via USB, SPI, RS232, and DIO

- ◦ Implementation of algorithms for the following components:
  - ▪ Board communication via USB from the host PC
  - ▪ Analog-to-digital converter (ADC) via SPI
  - ▪ Digital-to-analog converter (DAC) via SPI
  - ▪ Digital I/O via SPI
  - ▪ Bronkhorst IQFlow devices via RS232
  - ▪ Fan speed monitoring
  - ▪ Pump control via DIO
  - ▪ Firmware update via external SPI Flash
- ◦ Development of a Windows test application in C# to verify functionality
- ◦ Development of a firmware update tool in C# for Windows
- ◦ Programming languages: C/C++, C#
- ◦ Environment: MS Visual Studio, RedSuite, Git, ARM Compiler, Make, Eclipse
- ◦ Hardware: NXP LPC2378, NXP LPC2148, ADC, DAC, DIO, Bronkhorst IQFlow
- ◦ Remote work
- **Technologies:**
  - ◦ C/C++
  - ◦ Bare Metal
  - ◦ SPI
  - ◦ I²C
  - ◦ UART
  - ◦ C#
  - ◦ RS232
  - ◦ Git
  - ◦ Linker Scripts
  - ◦ Make
  - ◦ Microsoft Windows

Amplifier Development (EMC Environment)

Duration: January 2011 - March 2012
Position: Senior Software Architect/Developer

- **Industry:** Electrical Engineering / Measurement Instruments
- **Role:** Architecture / Implementation / Integration
  - ◦ Development of embedded software for a modular broadband amplifier system
  - ◦ Review of the existing solution with respect to architecture, build system, and modularization
  - ◦ Architecture design using UML, especially statecharts
  - ◦ Creation of a new build system using CMake
  - ◦ Linux kernel programming: implementation of CAN bus drivers
  - ◦ Complete reimplementation of the CAN bus layers for the existing software
  - ◦ Continuous Integration: setup of the build chain with version control (Git, ClearCase), build server (Hudson), automated test coverage, and firmware generation as a fully automated process
  - ◦ Programming language: C/C++
  - ◦ Environment: Embedded Linux, Qt 4.7.x, powerpc-gcc, CMake, MS Visual Studio, VMware
  - ◦ Hardware: PowerPC 603e, x86, CAN bus, FPGA, PCI bus
  - ◦ Tools:
    - ▪ Requirements: IBM Rational DOORS
    - ▪ Bug tracking: IBM Rational ClearQuest

▪ Version control: SVN, IBM Rational ClearCase
- **Technologies:**
  ◦ Embedded Linux
  ◦ Qt
  ◦ C/C++
  ◦ CMake
  ◦ SVN
  ◦ Doors
  ◦ CAN
  ◦ CMake
  ◦ PowerPC 603e
  ◦ RS485
  ◦ Modbus
  ◦ State Machine
  ◦ Unix

Embedded Device Development

Duration: October 2009 - December 2010
Position: Senior Software Developer

- **Industry:** Security Sector (Government)
- **Role:** Implementation / Release Management / Integration
  ◦ Development of a proof-of-concept solution
  ◦ Programming of specific modules (User Interface, Self-Test, etc.) for the production version
  ◦ Implementation work for secure messaging to smartcards
  ◦ Communication and coordination with external service providers
  ◦ Acceptance testing of deliverables from external service providers
  ◦ Continuous Integration: setup of build chain with version control (SVN), build server (Bamboo), automated test coverage, and firmware generation as an automated process
      ▪ Shell scripting: Bash and Windows PowerShell
      ▪ Programming language: C/C++
      ▪ Environment: Embedded Linux, Qt 4.6.x, arm-gcc, CMake, MS Visual Studio, VMware
      ▪ Hardware: Freescale i.MX51, Cirrus EP9301, Intel Atom, Smartcards
      ▪ Special technologies: Secure Boot, SOAP, PKI, PC/SC
      ▪ Tools:
          ▪ Project management: Instep
          ▪ Bug tracking: Jira
          ▪ Static code analysis: Klocwork
          ▪ Debugging: Insight
- **Technologies:**
  ◦ Embedded Linux
  ◦ Qt
  ◦ C/C++
  ◦ CMake
  ◦ SVN
  ◦ Cirrus EP9301
  ◦ Unix

Multimedia Consulting

Duration: January 2009 - June 2010
Position: Senior Software Developer

- **Industry:** Security Sector
- **Role:** Analysis and Implementation
    - Consulting in the area of video encoding
    - Architecture and implementation of an MPEG-2 transport stream analyzer
    - Contribution to the open-source project ffmpeg
        - Programming language: C/C++
        - Environment: Windows, Fedora 11/12 x64 Linux
        - Tools: ffmpeg, VLC, mplayer
        - Multimedia: MPEG-2 Transport Stream (ISO 13181-1), MPEG-2 Video, H. 264, STANAG 4609
- **Technologies:**
    - C/C++
    - MPEG-2
    - H.264
    - Microsoft Windows

Document Control System

Duration: February 2009 - September 2009
Position: Senior Software Developer

- **Industry:** Security Sector (Government)
- **Role:** Implementation
    - Integration of hardware components into the overall system (fingerprint reader, camera, external displays)
    - Communication between components over the network
    - Programming languages: Java J2EE, C/C++
    - Environment: Eclipse, MS Visual Studio
    - Tools: C/C++, Boost, log4cxx, Cognitec SDK, Canon SDK
    - Java stack: JBoss, Hibernate, Log4j
- **Technologies:**
    - Java
    - C/C++
    - SVN
    - Boost
    - Microsoft Windows

NGN - New Generation Networks / VoIP (HiQ)

Duration: March 2008 - December 2008
Position: Senior Software Developer

- **Industry:** Telecommunications
- **Role:** Analysis and Implementation
    - Implementation of various services in a VoIP server (Soft-PBX)

- Services: Lawful Interception, Parallel Ringing, Serial Ringing, Music On Hold
    - Software installation
    - Handling of change requests
    - Creation of test scenarios
    - Programming language: C/C++
    - Environment: Solaris, Linux, ClearCase, ClearQuest, TestDirector, make
    - Protocols: SIP, SDP, MGCP, RTP, 3GPP IMS, Mobile Centrix, SOAP
- **Technologies:**
    - C/C++
    - ClearCase
    - Make
    - SIP
    - SDP
    - MGCP
    - RTP
    - 3GPP IMS
    - SOAP
    - Microsoft Windows

Video Telephony (H.324M) with Asterisk

Duration: February 2008 - May 2008
Position: Senior Software Developer

- **Industry:** Telecommunications
- **Role:** Analysis and Implementation
    - Analysis of an existing video telephony solution
    - Implementation work within the Asterisk PBX for H.324M
    - Bug fixing of the H.324M stack
    - Delivery of patches for the Fontventa H.324M stack
    - Testing with Nokia N73 and Motorola K3
    - Programming languages: C, Java, Perl
    - Environment: Linux, gcc, Eclipse
- **Technologies:**
    - C
    - Java
    - Perl
    - Asterisk

System Architecture for IPTV

Duration: November 2007 - February 2008
Position: System Engineering

- **Industry:** Telecommunications
- **Role:** System Engineering
    - Development of various system architectures for an IPTV system
    - Description of client architecture and backend architecture (CMS) as well as the CDN (Content Delivery Network)
    - Evaluation of different IPTV technologies: Multicast, Unicast, Peer-to-Peer

- ◦ Evaluation of various IPTV platforms: Microsoft Windows Media / MSTV, Adobe Flash, RealNetworks
- **Technologies:**
  - ◦ IPTV

Implementation Work for a Document Control System

Duration: July 2007 - November 2007
Position: Senior Software Developer

- **Industry:** Security sector (government environment)
- **Role:** Implementation
  - ◦ Programming a GUI for an image processing framework: GUI logic, mainframe, and various dialogs
  - ◦ Using the Qt GUI framework from Trolltech
  - ◦ Controlling hardware: camera, lighting, positioner
  - ◦ Data storage in an SQL database (MySQL)
  - ◦ Reading configuration data using DOM XML parser
  - ◦ Output of statistical data via XML
  - ◦ Protection against unauthorized copying using a dongle from WiBu-Systems
  - ◦ System installation
    - ▪ Programming language: C++
    - ▪ Libraries: Qt 4.3.x (Trolltech), Boost
    - ▪ Environment: Windows XP, MS Visual Studio 2005
- **Technologies:**
  - ◦ C/C++
  - ◦ Qt
  - ◦ Boost
  - ◦ SVN
  - ◦ Microsoft Windows

IMS Videosharing/VoIP

Duration: April 2006 - July 2007
Position: System Engineering and Implementation

- **Industry:** IMS Videosharing/VoIP
- **Role:** Implementation
  - ◦ Development of use cases, requirements, specifications, architecture, and implementation for a system enabling real-time audio/video streaming between mobile devices (full duplex).
  - ◦ **Environment:**
    - ▪ Programming Languages: C/C++, C#, Visual Basic
    - ▪ Environment: 3GPP IMS, IPv6
    - ▪ Codecs: Video – MPEG-4, H.264; Audio – AMR, AMR-WB
    - ▪ Development Platform: Windows Mobile 5.x, Intel XScale PXA27x
- **Technologies:**
  - ◦ C++
  - ◦ C#
  - ◦ MPEG-4

- ◦ H.264
- ◦ AMR
- ◦ AMR-WB
- ◦ Symbian
- ◦ VxWorks

IMS Services in the Telecommunications Environment

Duration: November 2006 - June 2007
Position: Senior Software Developer

- **Industry:** Telecommunications
- **Role:** Implementation and Testing
  - ◦ Implementation of SIP services on the Nokia Siemens Networks Advantage platform
  - ◦ Implementation of a charging interface in a 3GPP IMS Application Server / B2B User-Agent
  - ◦ Development of the test environment for the 3GPP IMS Application Server / B2B User-Agent
  - ◦ Parsing ASN.1 data using Perl scripts for automatic documentation generation
  - ◦ Use of regular expressions in Perl and Java
  - ◦ Remote debugging with Eclipse (Solaris/Windows XP)
  - ◦ Creation of a test system:
    - ▪ Defining test script syntax
    - ▪ Designing a script interpreter (Java)
    - ▪ Controlling the test system with Perl scripts
  - ◦ **Environment:**
    - ▪ Programming language: Java 1.5 (Java SE 6)
    - ▪ Protocols: SIP, SDP, RTP, ASN.1
    - ▪ Environment: Solaris 10, Eclipse, Perl, codegen
- **Technologies:**
  - ◦ Java
  - ◦ Perl
  - ◦ SIP
  - ◦ SDP
  - ◦ RTP
  - ◦ ASN.1

MPEG-2 Transport Stream Multiplexer

Duration: June 2006 - October 2006
Position: Senior Software Developer

- **Industry:** TV Broadcasting
- **Role:** Implementation
  - ◦ Design and implementation of an MPEG-2 Transport Stream multiplexer.
  - ◦ Processing of MPEG-4/AVC or H.264 video streams and MPEG-4 AAC HE audio streams.
  - ◦ Testing with set-top boxes from Pace, Amino, and Stino.
  - ◦ **Environment:**
    - ▪ Programming Language: C++

- ▪ Protocols: MPEG-2 Systems
- ▪ Tools: VLC, mplayer, MS DirectShow
- ▪ Development Environment: MS Visual Studio .NET 2003, Cygwin, GCC
- **Technologies:**
  - ◦ C++
  - ◦ MPEG-2
  - ◦ MPEG-4
  - ◦ H.264
  - ◦ AAC

IMS Services

Duration: August 2005 - March 2006
Position: Systems Architect

- **Industry:** Telecommunications
- **Role:** Systems Engineering
  - ◦ Development of system specifications for IMS services. Work focused on Push-To-Talk, Video Sharing, Presence, and other IMS-related services. Creation of use cases, requirements, and overall system architecture.
- **Technologies:**
  - ◦ IMS
  - ◦ PoC

SigComp (IETF)

Duration: May 2005 - December 2005
Position: Systems Architect

- **Industry:** Telecommunications
- **Role:** Systems Engineering and Implementation
  - ◦ Design, architecture, and implementation of the Signal Compression Layer (SigComp). Implementation in C for an embedded platform.
- **Technologies:**
  - ◦ SigComp

AUTOSAR

Duration: July 2005 - July 2005
Position: Systems Architect

- **Industry:** Automotive
- **Role:** Systems Engineering
  - ◦ Work in the automotive domain: review of AUTOSAR specifications, development of solutions within the AUTOSAR standardization process.
- **Technologies:**
  - ◦ Autosar

IMS Video Sharing

Duration: March 2005 - June 2005
Position: Systems Architect

- **Industry:** Telecommunications
- **Role:** Systems Engineering
  - Architecture design for 3GPP IP Multimedia calls in the IMS.
  - Creation of call flows, definition of requirements, and design of architecture components for mobile phone clients.
- **Technologies:**
  - 3GPP
  - IMS

Push-To-Talk (PoC)

Duration: August 2004 - June 2005
Position: Systems Architect

- **Industry:** Telecommunications
- **Role:** Systems Engineering
  - Software architecture design and planning for PoC and video streaming in the mobile domain on the client side. Design specifications for embedded platforms (Apoxi, embedded Linux). Representation of the client at the Open Mobile Alliance (OMA).
- **Technologies:**
  - OMA
  - PoC
  - Embedded Linux

IMTC Conformance Testing

Duration: March 2004 - July 2004
Position: Software Developer

- **Industry:** Telecommunications
- **Role:** Implementation and Testing
  - Representation of the client at the IMTC
  - Management and execution of conformance tests for video/audio codecs and the 3GPP file format within the IMTC organization.
  - **Environment:**
    - Programming Languages: C/C++, Perl, Visual Basic
    - Protocols: MPEG-4/H.263, AMR, AAC
    - Environment: ARM Toolchain, Visual Studio, gcc
- **Technologies:**
  - C/C++
  - Perl
  - Visual Basic
  - MPEG-4
  - AMR
  - AAC

Multimedia Framework for Mobile Devices

Duration: August 2003 - December 2003
Position: System Engineer

- **Industry:** Telecommunications
- **Role:** Systems Engineering and Implementation
    - Development of the architecture and components for a middleware multimedia framework (similar to DirectShow) for mobile devices (Nucleus/APOXI).
- **Technologies:**
    - Nucleus

Transmission of Multimedia Data over Mobile Networks

Duration: August 2003 - December 2003
Position: Research Associate

- **Industry:** Telecommunications
- **Role:** Systems Engineering and Implementation
    - Setup and development of a simulation environment for optimized video data transmission using EGPRS and WLAN. Protocols used: RTP, UDP, IP, EGPRS, and IEEE 802.11b. Video standards used were MPEG-4 and H.264 (AVC).
- **Technologies:**
    - RTP
    - UDP
    - IP
    - EGPRS
    - MPEG-4
    - H.264
    - C++
    - Perl
    - Matlab

MPEG-4 Content Management

Duration: January 2001 - June 2001
Position: Research Associate

- **Industry:** Telecommunications
- **Role:** Implementation
    - Analysis and development related to the storage of multimedia content in the MPEG-4 file format.
- **Technologies:**
    - MPEG-4

Virtual 3D Video Conferencing Systems

Duration: January 1999 - December 2000
Position: Research Associate

- **Industry:** Telecommunications
- **Role:** Systems Engineering and Implementation
  - Development of the foundations for future 3D video conferencing systems using multiple camera setups.
- **Technologies:**
  - MPEG-4

Contribution to the Development of the MPEG-4 Video Standard

Duration: May 1998 - December 1998
Position: Research Associate

- **Industry:** Telecommunications
- **Role:** Systems Engineering and Implementation
  - Temporary supervision of the EU project MoMuSys as well as the development and implementation of own algorithms for MPEG-4.
- **Technologies:**
  - MPEG-4

Fast Motion Estimation in MPEG Encoding

Duration: May 1998 - December 1998
Position: Research Associate

- **Industry:** Telecommunications
- **Role:** Implementation
  - Implementation of fast motion estimation methods in MPEG-2 and MPEG-4 video compression. Development in Intel assembler using MMX and SSE extensions.
- **Technologies:**
  - MPEG-4
  - MPEG-2
  - Assembler

Fractal Image Coding

Duration: January 1997 - December 1997
Position: Research Associate

- **Industry:** Telecommunications
- **Role:** Systems Engineering and Implementation
  - Development of an image coding system using fractal methods and algorithms. Implementation using object-oriented methods in C++.

- **Technologies:**
  - C++
  - Pascal

DIY Project: Z80 PC

Duration: September 1986 - December 1988
Position: Apprentice

- **Industry:** IT
- **Role:** Implementation
  - Construction of a Z80 PC, including creating the circuit boards, soldering, and troubleshooting using an oscilloscope.
  - Development of a keyboard controller with Z8 CPU
  - Development of a floppy disk controller
  - Implementation of the BIOS in assembler
  - Adaptation of CP/M
- **Technologies:**
  - Zilog Z80
  - Assembler

PUBLICATIONS

Video Transmission in Packet-Oriented Networks Using a Priority-Controlled Buffer

Date: April 2008
pending

Combination of Link Layer ARQ and Unequal Error Protection at the Application Layer

Date: June 2003
https://priorartdatabase.com/IPCOM/000012491

Graphical Presentation of Position Data from Multiple Mobile Devices on the Respective Involved Devices

Date: July 2006
https://priorartdatabase.com/IPCOM/000137933

EDUCATION

Technische Universität Berlin

Duration: October 1990 - December 1996
Degree: Graduate Engineer in Electrical Engineering

Specialization in Telecommunications and Networks

Lufttechnische Anlagen Berlin

Duration: September 1985 - February 1988
Degree: Toolmaker