Advanced Topics in SW Engineering – Embedded Systems

Dr. Rupert Stützle
TU München, October 25, 2021

Agenda

- Introduction
- Part I: Market and Requirements
  - Example 1: Automotive Engine Control Unit
  - Example 2: Automotive Infotainment Module
  - Embedded Systems Domains and Characteristics, Trends
  - Conclusions, Part I
- Part II: Architecture and Implementation
  - Example 1: Automotive Engine Control Unit
  - Example 2: Automotive Infotainment Module
  - Technology: general trends
  - Conclusions, Part II

What is an Embedded System?

Embedded System: Attempts at a Definition

Various definitions and points of view:

- Embedded System = real-time system, small microcontroller, little memory available, programmed in Assembler or C
- „There‘s no such thing as an Embedded System“ (Source: www.***embedded***.com)
- „Information system, that is integrated into a larger product“

Agenda

- Introduction
- Part I: Market and Requirements
  - Example 1: Automotive Engine Control Unit
  - Example 2: Automotive Infotainment Module
  - Embedded Systems Domains and Characteristics, Trends
  - Conclusions, Part I
- Part II: Architecture and Implementation
  - Example 1: Automotive Engine Control Unit
  - Example 2: Automotive Infotainment Module
  - Technology: general trends
  - Conclusions, Part II

Example 1: Engine Control Unit – Requirements

Real-time capabilities	hard - synchronized with crankshaft (ms); down to µs (for valve control, combustion chamber pressure control)

Functional Safety	min. ASIL B; up to ASIL D (according to ISO 26262) ==> critical faults are safely under control, e.g., self-acceleration, blocked traction axle

Environmental conditions	mounted in engine compartment
Temperature	-40° - 105° C
Protection class	IP6K9K
Power supply	12V network, voltage dips down to 4V

Quality	No stalled vehicles over lifetime, failures in low ppms
Life span	15 years
Operating time	8,000 hrs

Source: Bosch Diesel Gasoline Systems

Engine Control Unit – Requirements: Functionality

Engine functions

- Injection control
- Idle control
- Engine coordination
- Engine speed acquisition
- Ventilator control
- Air system: EGR control, turbocharger control
- Exhaust system: Lambda control, exhaust temperature modeling
- …

Vehicle functions

- Cruise control
- Immobilizer
- Diagnosis system
- Bus communication (CAN, FlexRay)
- …

Source: Bosch Diesel Gasoline Systems

Engine Control Unit – Requirements: Interfaces

Sensors

- Temperature: engine, air, motor oil, fuel, exhaust
- Pressure: turbocharger, ambient pressure, oil, fuel, rail pressure
- Engine speed, angle of rotation
- Accelerator pedal (angle)
- Air mass
- Oxygen concentration in exhaust (lambda sond)

Actuators

- Injectors + high-pressure pump
- Fuel pump
- Turbocharger pressure valve
- Intake manifold throttle
- Throttle valve
- Ventilator control
- EGR valve (exhaust gas recirculation)

Engine Control Unit – Requirements: Interfaces

Interfaces/physical layer

- CAN
- FlexRay

Interfaces to other units

- Immobilizer
- Transmission control unit
- ABS*/ESP** control unit

APIs to be implemented

- AUTOSAR
- Standard interfaces for off-board communication
  - Diagnosis (UDS/ODX)
  - Measurement
  - Calibration
  - Flash programming

* Anti-lock brakes   ** Electronic stability program

Engine Control Unit, non-functional Requirements (1)

Costs

- Prio A: minimizing **unit costs**
  - typical objective: reference calculation (e.g., best-of-best)
  - highly professional sourcing, high cost pressure

- Prio B: minimizing **development costs**
  - occasionally considered through DCF* calculation
  - importance increases with increasing number of variants

Similar situation for all Automotive control units

* Discounted Cash Flow

Engine Control Unit, non-functional Requirements (2)

Variants

- Variant concept and management for varying engine sizes and engine-power class, manufacturers and regions/legal requirements
- Construction kit for housing, hardware (population variants, reuse of building blocks) and software (calibration variants)
- > 500 variants overall, >50 HW variants

- **Ever-increasing number of vehicle variants**
  ➔ variant management is one of the key challenges
- **Similar situation for all Automotive control units**

SOP

Start of Production

Engine Control Unit, Requirements: Process

NON-EXHAUSTIVE

- The SOP must be met **at (almost) all cost**
- Automotive certification according to ISO/TS 16949
- Assessment of development: CMMI (level 3 min.) or Automotive SPICE
- Global localization capabilites in development and production (e.g., USA, China)
- Staged vehicle integration process with partial freezing
- Requirements freeze **one year before SOP**

Similar situation for all Automotive control units

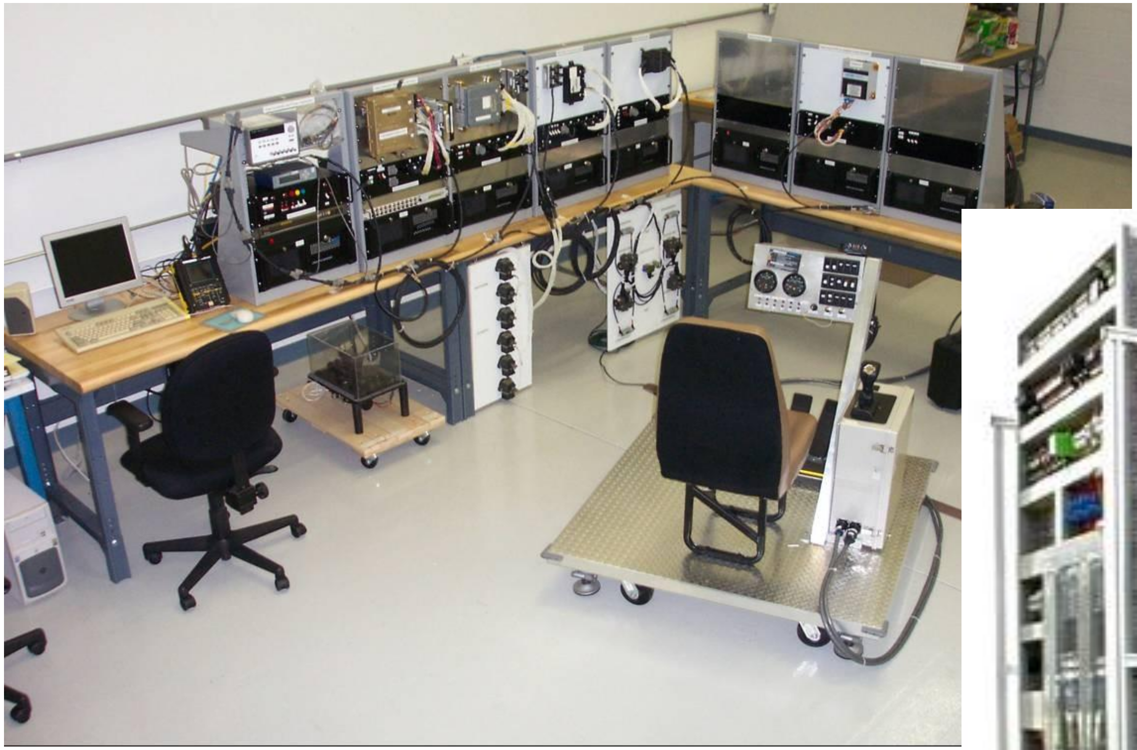
Engine Control Unit, Reqs.: Quality assurance

- Unit Tests as SIL regression tests
- Integration tests and system tests as HIL regression tests
- Integration tests and system tests on vehicles as prerequisite for series release (min. kms driven)
- Compliance with MISRA* coding standards, tested with specific tools (➔“MISRA warnings”)

- **Test automation using dedicated test systems is very important – significant portion of development effort**
- **Similar situation for all Automotive control units**

* **M**otor **I**ndustry **S**oftware **R**eliability **A**ssociation **–** coding standards for the automotive industry

Hardware-in-the-Loop (HIL): Examples



“Virtual Vehicles“

Source: Mathworks, www.elektroniknet.de

Agenda

- Introduction
- Part I: Market and Requirements
  - Example 1: Automotive Engine Control Unit
  - Example 2: Automotive Infotainment Module
  - Embedded Systems Domains and Characteristics, Trends
  - Conclusions, Part I
- Part II: Architecture and Implementation
  - Example 1: Automotive Engine Control Unit
  - Example 2: Automotive Infotainment Module
  - Technology: general trends
  - Conclusions, Part II

Example 2: Infotainment Module (Head Unit)

Real-time capabilities	no requirements but: strong requirements re. processing performance, e.g. navigation

Functional Safety	no requirements

Environmental conditions	mounted in cabin
Temperature	-40° - 85° C
Protection class	IP6K2
Power supply	12V network, voltage dips down to 4V

Quality	Failures in low ppms
Life span	15 years
Operating time	12,000 hrs

Source: Mercedes, BMW

Infotainment Module: key characteristics – summary

- User interface, challenges
  - Consistent HMI over various displays (dashboard, head unit, head-up display, …)
  - Organic user interaction
  - Variants (languages, views, transitions)
  - …
- Drivers of Architecture
  - Cooperations, 3rd party Software
  - Integration of 3rd party devices (e.g., Smartphone)
  - Usage of Platforms (e.g., Genivi)
  - Availability of second source for important suppliers
  - Reduction of license fees (e.g., for operating systems, …)
- Connected Navigation is a data source for Advanced Driver Assistance Systems (ADAS), e.g., learning speed limits and maximum curve speeds for Adaptive Cruise Control (ACC)
- Multiple collaboration models with OEMs and other suppliers (HW and/or SW)

Agenda

- Introduction
- Part I: Market and Requirements
  - Example 1: Automotive Engine Control Unit
  - Example 2: Automotive Infotainment Module
  - Embedded Systems Domains and Characteristics, Trends
  - Conclusions, Part I
- Part II: Architecture and Implementation
  - Example 1: Automotive Engine Control Unit
  - Example 2: Automotive Infotainment Module
  - Technology: general trends
  - Conclusions, Part II

Embedded Systems: Domain Overview

NON-EXHAUSTIVE

- Embedded Systems
  - Consumer Products
    - Consumer Electronics
      - Smart phones,Tablets
      - TV sets, music players
      - Cameras
      - Gaming consoles
    - Appliances
  - Home auto-mation
    - Air conditioning
    - Lighting
    - Security systems
  - Transport
    - Automotive
    - Railway technology
    - Aviation, Aerospace
  - Industry
    - Machine controls
    - Robotics
  - Medical Engineering
  - Defence technology

Automotive: Sub-Domains

- Embedded Systems
  - Consumer Products
    - Consumer Electronics
      - Smart phones,Tablets
      - TV sets, music players
      - Cameras
      - Gaming consoles
    - Appliances
  - Home auto-mation
    - Air conditioning
    - Lighting
    - Security systems
  - Transport
    - Automotive
      - Vehicle motion — *Engine (ICE, E), transmission, ABS, ESP, …*
      - Driver Assistance/ Automated Driving — *ACC, FCW, Warn-Brake-Evade, Lane Assist, ACA, …*
      - Body — *Body Controller, Energy management,…*
      - Infotainment, Connectivity — *Instrument cluster, Head unit, Rear seat Entertainment, …*
    - Railway technology
    - Aviation, Aerospace
  - Industry
    - Machine controls
    - Robotics
  - Medical Engineering
  - Defence technology

Domains: relative importance of optimization criteria

Relative assessment –
within domain, view might differ

Embedded Systems	Cost	Time to market	Quality
Consumer Products	!!!	!!!	0
Home automation	!!!	!! ↗	!
Automotive	!!!	!!	!!
Railway technology	!!	!	!!
Aviation, aerospace	!! - !	! - 0	!!!
Industry	!!	! ↗	!!
Medical engineering	!	! ↗	!!!
Defence technology	0	0	!!!

Embedded Systems: Domains and Trends

- Embedded Systems
  - Consumer Products
  - Home automation
  - Transport
    - Automotive
    - Railway tech.
    - Aviation, Aerospace
  - Industry
  - Medical engineering
  - Defence technology

Internet of Things

Industry 4.0

Industry 4.0 (development & production proc.)

Agenda

- Introduction
- Part I: Market and Requirements
  - Example 1: Automotive Engine Control Unit
  - Example 2: Automotive Infotainment Module
  - Embedded Systems Domains and Characteristics, Trends
  - Conclusions, Part I
- Part II: Architecture and Implementation
  - Example 1: Automotive Engine Control Unit
  - Example 2: Automotive Infotainment Module
  - Technology: general trends
  - Conclusions, Part II

Embedded Systems: Conclusions, Part I (1/2)

- Wide range of embedded systems, (non-)functional requirements varying widely between (sub-)domains (➔ there's no such thing as **the** embedded system)
- Commonalities:
  - Information System is integrated into a physical thing
  - Functionality is specific to the thing/its environment
  - Strong interaction with the physical environment through specific hardware and interfaces
- Optional characteristics:
  - Real-time capabilities, control of physical processes
  - High reliability

Embedded Systems: Conclusions, Part 1 (2/2)

With the Internet of Things and Industry 4.0 as two of the most important driving forces of technological and commercial progress today

- Embedded systems are gaining importance
- Back-end connectivity is key
- To provide new services, it is necessary to merge the worlds of Embedded (front-end) and IT (back-end)

Secure back-end connectivity as future commonality

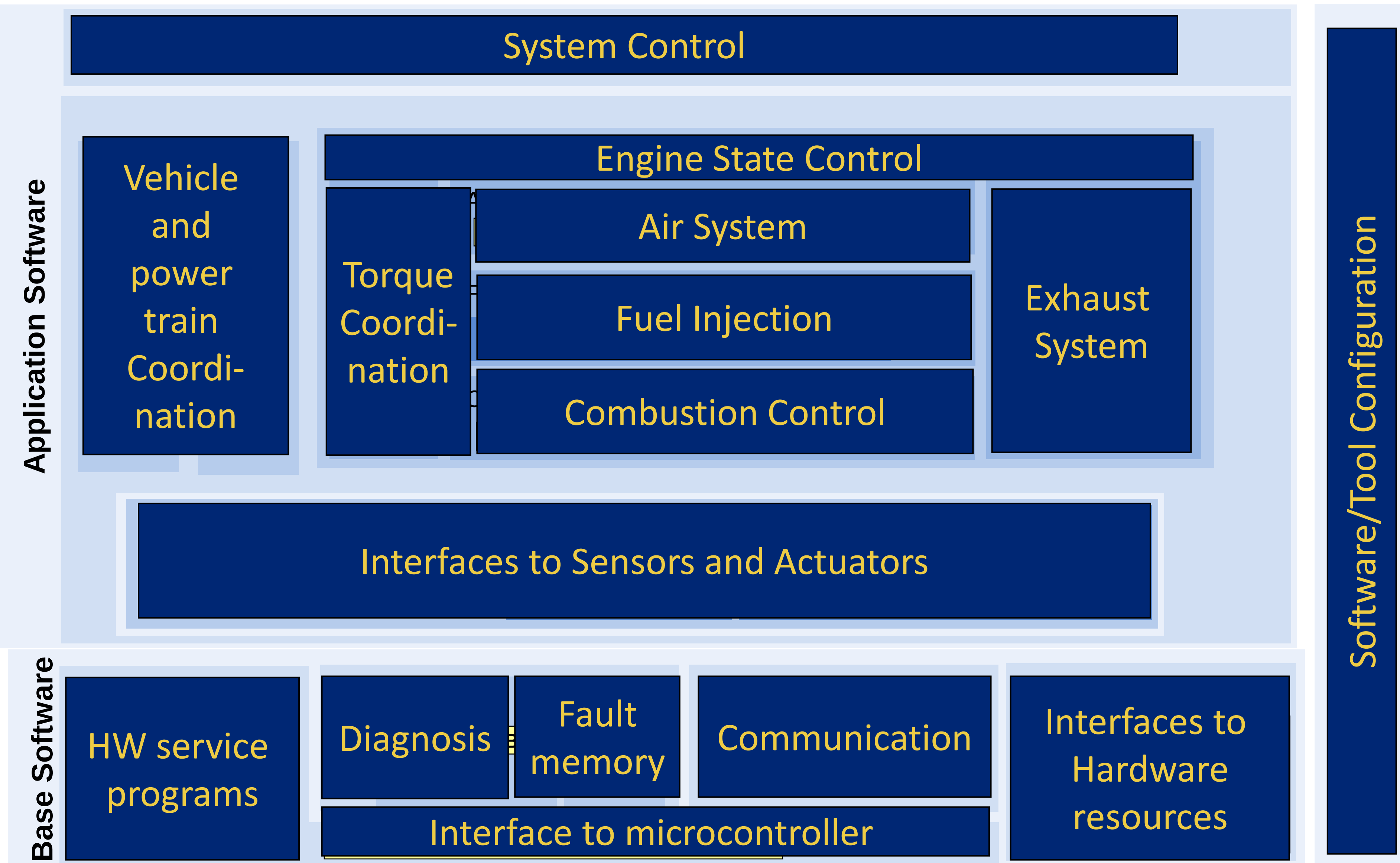
Agenda

- Introduction
- Part I: Market and Requirements
  - Example 1: Automotive Engine Control Unit
  - Example 2: Automotive Infotainment Module
  - Embedded Systems Domains and Characteristics, Trends
  - Conclusions, Part I
- Part II: Architecture and Implementation
  - Example 1: Automotive Engine Control Unit
  - Example 2: Automotive Infotainment Module
  - Technology: general trends
  - Conclusions, Part II

Engine Control Unit: HW, Operating System

- Infineon (IFX) Or ST multi-core controller
- Security module for
  - Tuning protection (secure authentication of object code image)
  - Immobilizer authentication (activated by electronic key module)
- Bus systems
  - Up to 5 CAN bus interfaces
  - FlexRay for real-time applications
  - Future: Ethernet
- Real-time OSEK/AUTOSAR-OS, task scheduling with message concept, static configuration
- AUTOSAR RTE (Run-Time Environment) as HW abstraction layer

Engine Control Unit: Software Architecture (1/2)



Source: Bosch Diesel Gasoline Systems

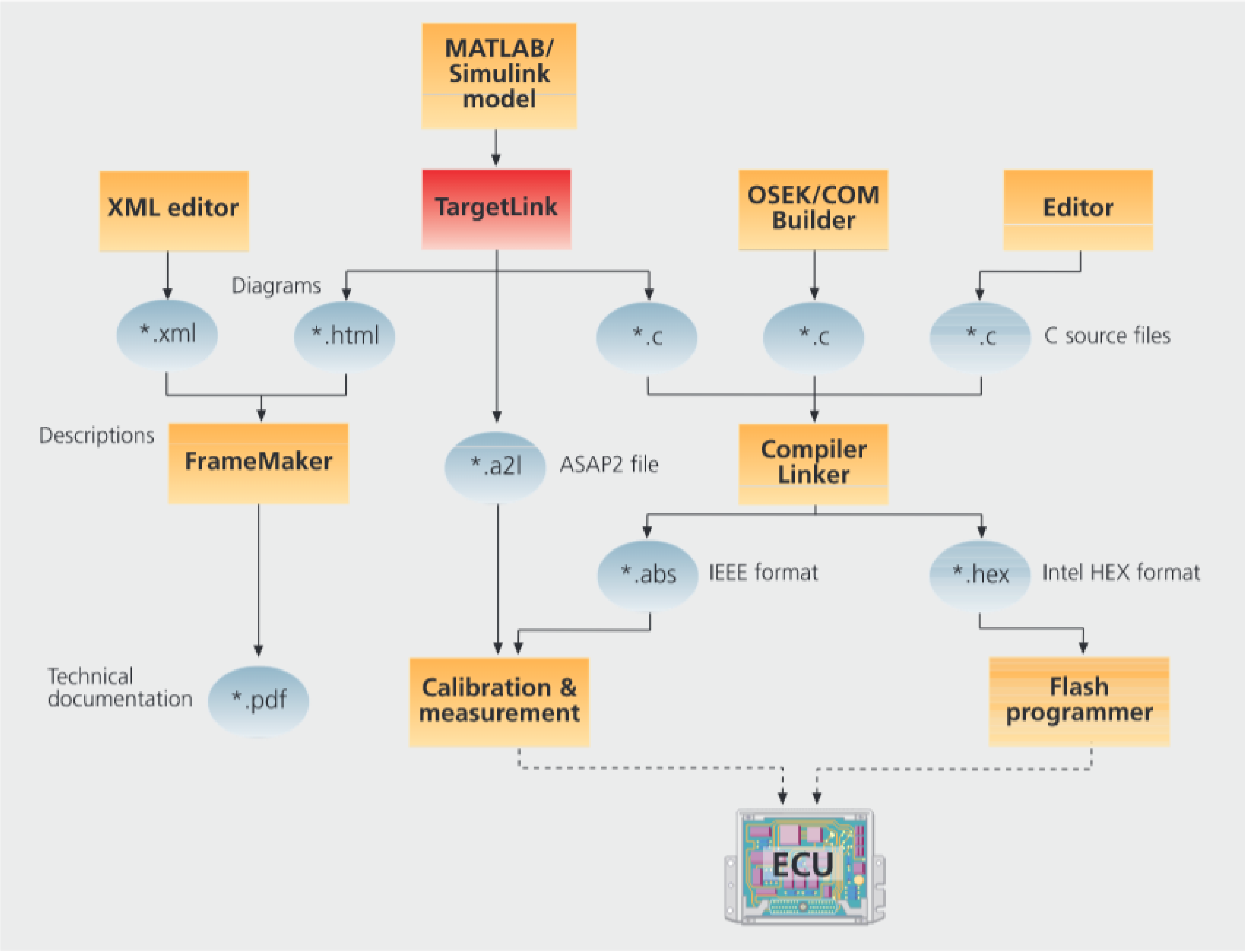
Engine Control Unit: Architecture

- Layered Architecture with Hardware Abstraction Layer
- Implementation of AUTOSAR-API
- Specific (“complex”) drivers for peripherals with real-time constraints
- “Torque path” as functional model
- Functional safety architecture: 3-level monitoring concept
  - Level 1: plausibility checks of sensor values and actuator condition
  - Level 2: redundant calculation of maximum torque and checking of current fuel quantity + check of zero fuel quantity when coasting
  - Level 3: monitoring of microcontroller with watchdog or redundant controller core in lock-step mode
- > 50,000 parameters for vehicle application/calibration

Engine Control Unit: Implementation, Tools

- Programming language: C, in the future also C++
- **Model-based code generation** for control functions, using tools like Matlab/Simulink (de-facto standard) and TargetLink or ASCET
- Variants generated at compile time through selective compilation (“compiler switches”)
- SW functionality is adapted to specific vehicle and powertrain by **tool-based calibration** (setting of parameters)
- SW provisioning to the final product: **end-of-line programming** (“flashing”) in the factory; specific tool chain for software logistics

Engine Control Unit: Model-Based Code Generation



Source:www.dspace.com

Agenda

- Introduction
- Part I: Market and Requirements
  - Example 1: Automotive Engine Control Unit
  - Example 2: Automotive Infotainment Module
  - Embedded Systems Domains and Characteristics, Trends
  - Conclusions, Part I
- Part II: Architecture and Implementation
  - Example 1: Automotive Engine Control Unit
  - Example 2: Automotive Infotainment Module
  - Technology: general trends
  - Conclusions, Part II

Infotainment Module: Implementation Choices

- **Linux operating system – drivers:**
  - functionality, innovation
  - quality, security
- **Usage of state-of-the-art processes and toolchain** (mostly open-source), e.g.
  - Continuous integration with **Jenkins**
  - Distributed version control with **git**
- **Separate domain for safety-critical applications** to increase IT security
- **Sandbox** for entertainment applications

- **Open-source software is widely used to ensure speed and functionality**
- **IT security is main concern to protect safety-critical functions**

Infotainment Module: key characteristics - summary

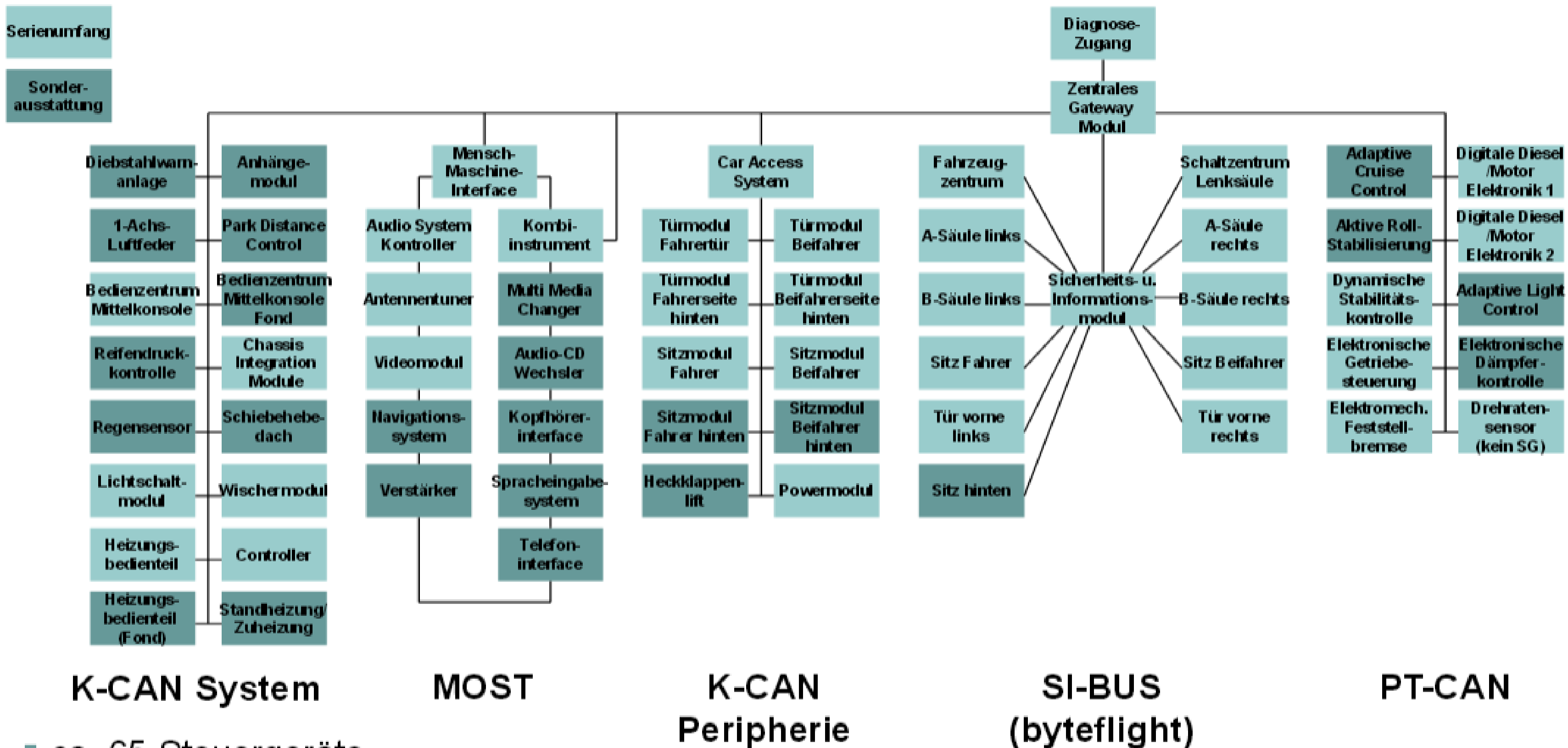
- Future partitioning in two domains:
  - Automotive: key objective is safety (e.g., ADAS[1], FOTA[2], Diagnosis)
  - Internet of Things: key objective is connectivity (e.g., media player, connectivity)
- Linux as platform ➔ functionality, innovation, quality and security
- Firewall between domains to ensure IT security
  - Inter-Process Communication via proxies
  - Dedicated computational resources allocated to Automotive domain to ensure functionality even if there is malfunction in IoT domain
- Back-end with secure access is indispensable for asynchronous feature updates and integration of services from Web
- Software statistics
  - 3rd party share of code growing much faster than in-house code
  - Share of Assembler code <2%, C code ~50%, C++ 14%, rest Java, Python, Perl, Vala, …
- Agile process with state-of-the-art tools (git, Jenkins, …)

1) Advanced Driver Assistance Systems 2) Firmware Over-the-Air Update

Agenda

- Introduction
- Part I: Market and Requirements
  - Example 1: Automotive Engine Control Unit
  - Example 2: Automotive Infotainment Module
  - Embedded Systems Domains and Characteristics, Trends
  - Conclusions, Part I
- Part II: Architecture and Implementation
  - Example 1: Automotive Engine Control Unit
  - Example 2: Automotive Infotainment Module
  - Technology: general trends
  - Conclusions, Part II

Traditional: distributed E/E Architecture (Automotive)



- ca. 65 Steuergeräte
- ca. 115 MByte, davon 80% Infotainment Software Umfang
- ca. 900 Funktionen

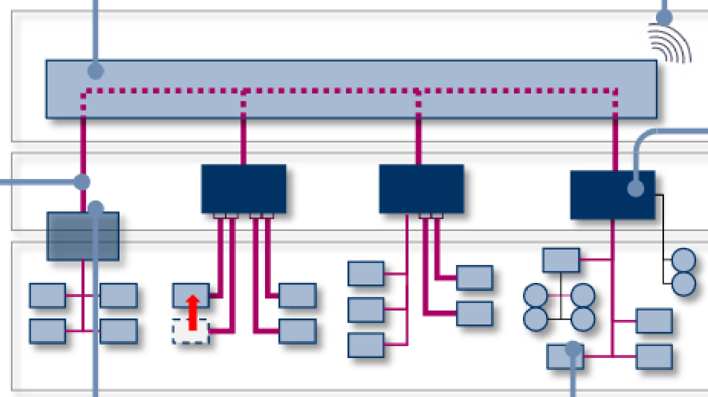
Quelle: BMW Group, 2005

Source: www.ifr.ing.tu-bs.de

New: domain controller architecture (Automotive)

Advanced Gateways
Increased routing complexity and communication throughput.

Connected Car
Open E/E architecture with external communication, secure connection and SW download.
(FOTA/SOTA)



Communication Network
Higher bandwidth, flexible communication mechanism.
(Gbit Ethernet)

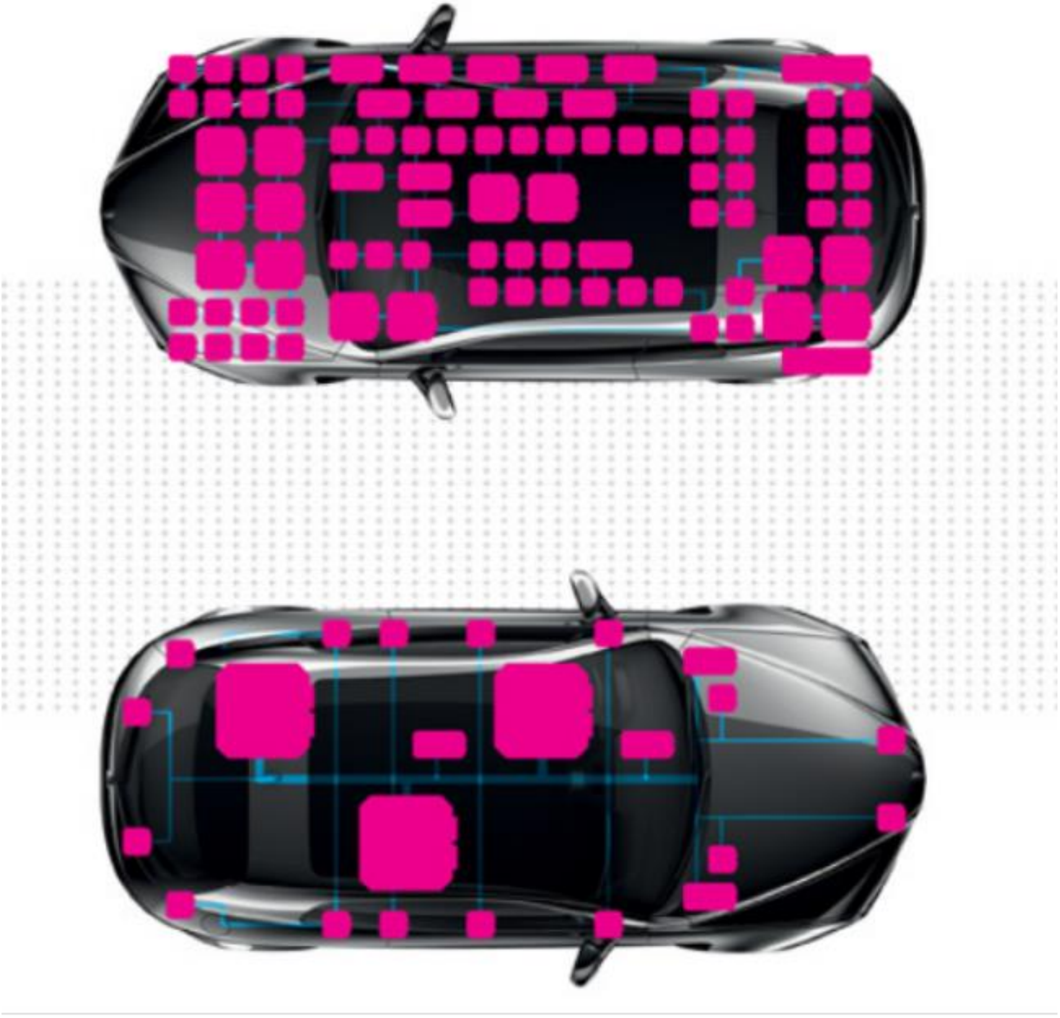
Domain ECUs
Powerful controller as integration platform for cross domain software components, high performance platform.
(non-automotive µP)

Integration Platform
Integration platform for superposed software components.
(Hypervisor)

Embedded System ECUs
Focusing on deep system specific functionalities.

Source: www.asam.net, Bosch Automotive Electronics

Old vs. new architecture: example Volkswagen



Source: carIT/Automobilproduktion IAA Sonderedition 2019, Interview with Rolf Zöller

Embedded Systems Technology: General Trends

Hardware Platforms: Microcontrollers with ARM core and specific hardware features

- ARM core microcontrollers as de-facto standard for many applications
- Increasing specificity of solutions to their application, e.g., specific controller modules with dedicated hardware (e.g., GPS, modems, smart card) and software frameworks
- Security features have become standard (e.g., secure boot, cryptography)
- Crypto controllers/Smart Cards are widely used for fast execution of asymmetric cryptography operations (en-/decryption, signing and authenticating)

Embedded Systems Technology: General Trends

System stacks: well-known from the IT world

- Open Source system stacks and tools are widely used
- Operating Systems: Linux and derivatives (especially Android) are widespread
- Back-End connectivity: REST
- Secure communication: TLS
- Specific libraries for IT Security, e.g., Bouncy Castle

Embedded Systems Technology: General Trends

Implementation and Tools: carry over from IT world

- High-level languages like Java are increasingly used
- Tools and processes do not differ from classic IT:
  - Agile processes using advanced tools, e.g., JIRA
  - Continuous integration, e.g., Jenkins
  - Test automation, e.g. using DSLs like groovy

Agenda

- Introduction
- Part I: Market and Requirements
  - Example 1: Automotive Engine Control Unit
  - Example 2: Automotive Infotainment Module
  - Embedded Systems Domains and Characteristics, Trends
  - Conclusions, Part I
- Part II: Architecture and Implementation
  - Example 1: Automotive Engine Control Unit
  - Example 2: Automotive Infotainment Module
  - Technology: general trends
  - Conclusions, Part II

Embedded Systems: Conclusions, Part II (1/2)

Architecture, implementation technologies and processes used in Embedded Software Engineering are increasingly similar to those used in classical Software Engineering

- State-of-the-art development overall
- Increased efficiency
  - Standard libraries, frameworks and tools (especially Open Source)
  - Agile development processes
- Optimal quality and process control
  - Continuous integration
  - Test automation

The boundaries are blurring – no world apart

Embedded Systems: Conclusions, Part II (2/2)

Implementation of connectivity and IT security requirements plays an important role in architecture and implementation

- Prevent hackers from accessing critical functions
- Allow end users to access and use their own valued devices and back-end services in all environments
- Minimize implementation overhead, control complexity
  - Usage of standards such as TLS
  - Reuse of open-source frameworks and libraries

Secure connectivity is key in architecture

Embedded System: Definition

Information system, that is integrated into a **physical product**, and has the following common properties:

- Functionality is specific to the product/its environment
- Strong interaction with the physical environment through specific hardware (sensors, actuators, user interfaces) and interfaces
- Important, but not common properties in certain domains:
  - Real-time constraints
  - Safety and reliability requirements
  - Severely limited resources (processor, memory)
  - Energy efficiency requirements (especially mobile systems)

Finally: Embedded ist about…

Physical products

… and their services

Contact information

If you have questions, comments or feedback (highly welcome!), please send me eMail:

rupert.stuetzle@gmail.com