

Simplify rich applications & hard real-time



F&S Elektronik Systeme GmbH
Untere Waldplätze 23
D-70569 Stuttgart
Tel. (+49)(0)711 123722-0
Speaker: Dipl.Ing.(FH) Martin Kieper

- Founded 1990
- Privately-held company
- Specialized on Embedded Systems
 - ◆ Single-Board-Computer (SBC)
 - ◆ System-on-Module (SoM)
- Development and Production in Stuttgart (Germany)
 - ◆ Hardware and Software
 - WindowsCE / Linux
- Customer base: 60% medical, 30% industrial, 10% other

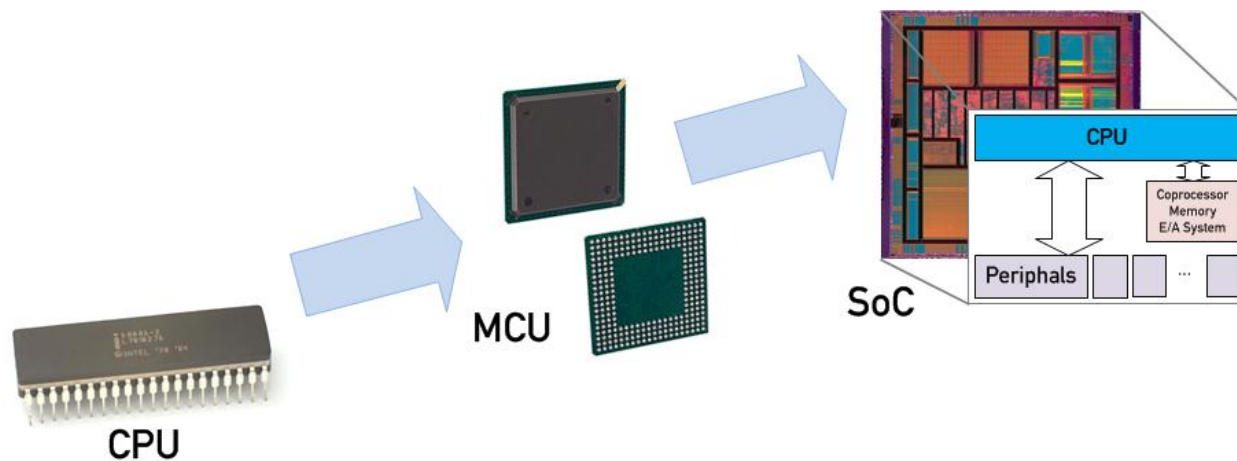


Agenda

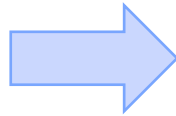
- Introduction and short history of embedded systems
- Requirements for today's embedded systems
- Current approaches to fulfill these requirements
- Alternative concepts
- Summary and future prospects

Introduction

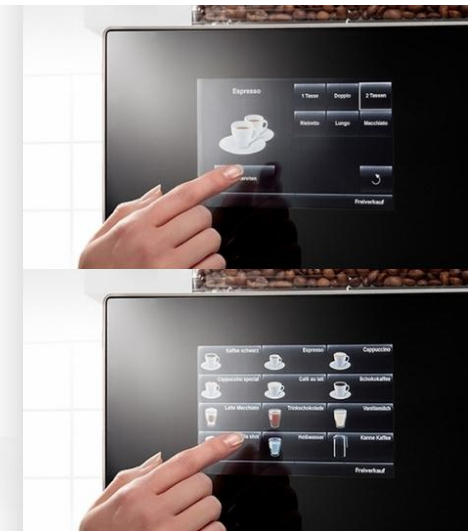
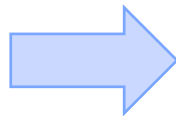
- “Special purpose system which is either used standalone or as part of a bigger system”
- In contrast: General Purpose System (e.g. PC)
- From small simple microprocessors ...
 - ◆ e.g. Motorola M6800 (8bit @2MHz)
- ... complex microcontrollers and System On Chips (SoC)
 - ◆ e.g. iMX.6 Quad (4x Cortex A9 @1GHz)



embedded devices –past and today



- Requirements to embedded devices changed dramatically
 - ◆ „look and feel“ of iPhone/iPad expected



- Basic requirements have not changed

Requirements to embedded systems

- **Human-Machine Interface**

- ◆ graphics instead of simple texts
- ◆ real-time diagrams
- ◆ visual gimmicks (animations, effects)
- ◆ touch-screen
- ◆ multimedia

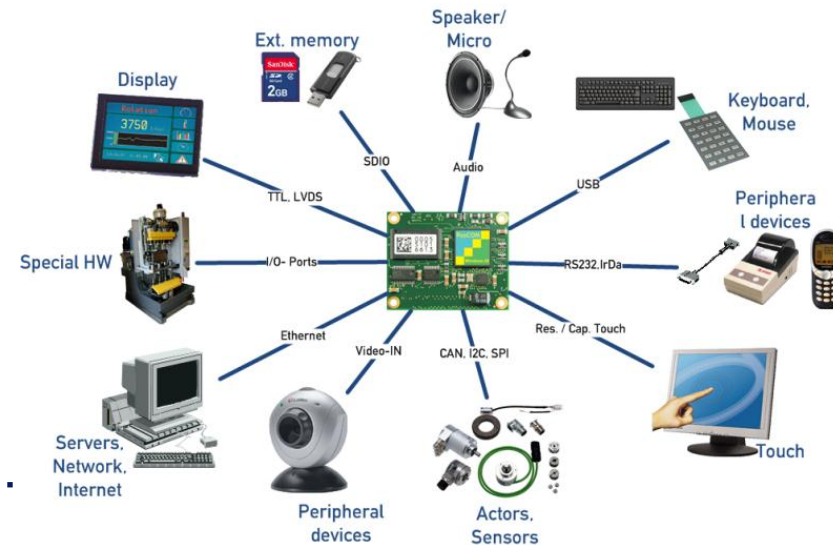


- **Flexibility and extensibility**

- ◆ additional features
- ◆ maintenance

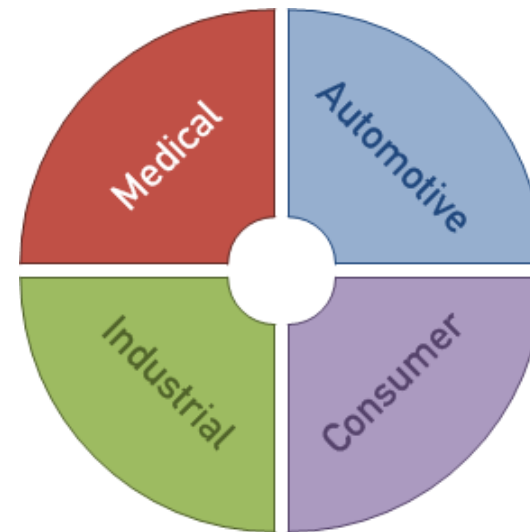
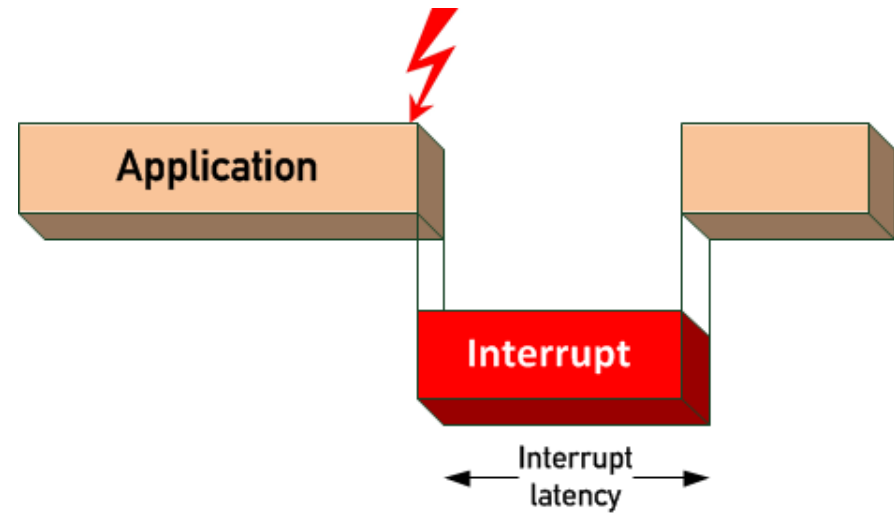
- **Connectivity**

- ◆ USB, RS232, Ethernet
- ◆ industrial busses: I2C, SPI, CAN,...
- ◆ sensors and actors

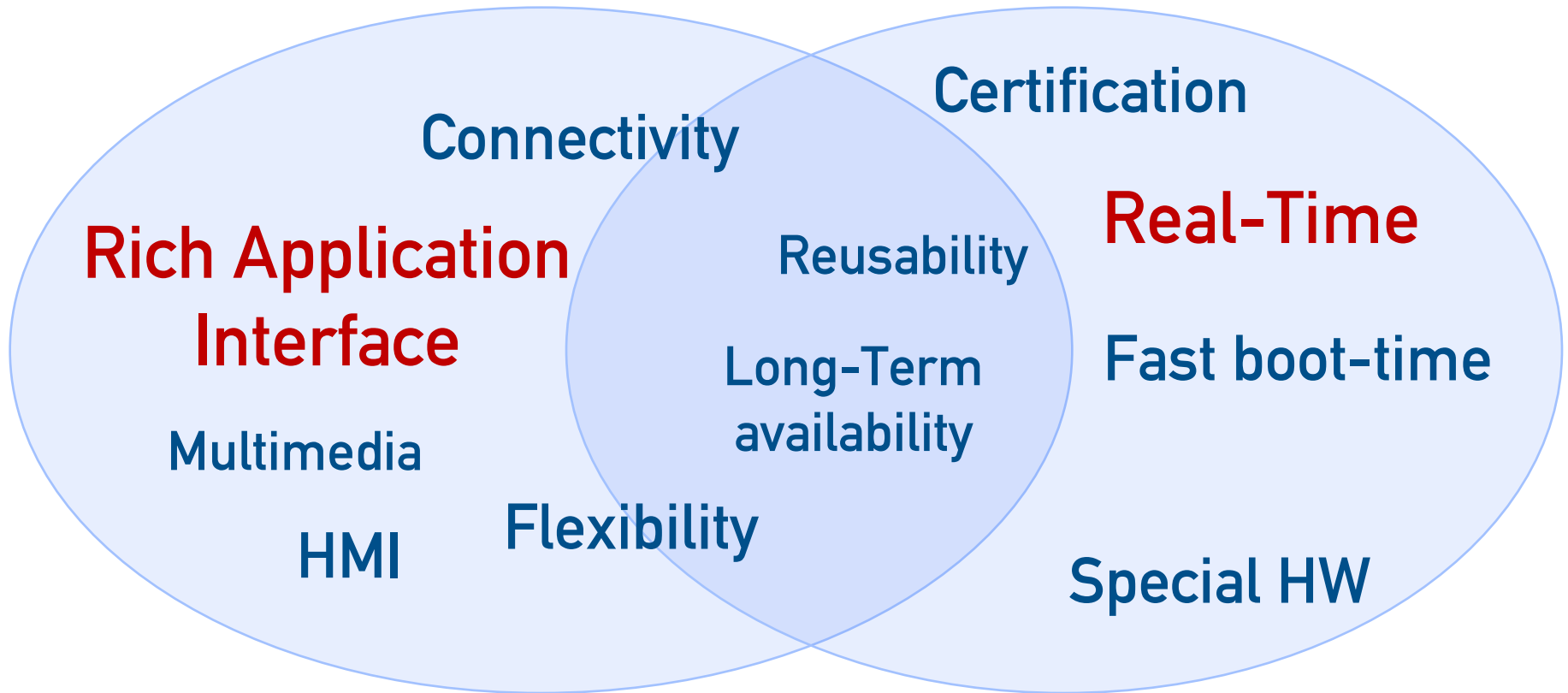


Requirements to embedded systems

- Real-Time capabilities
 - ◆ predictable response-time
- Fast boot-times
- Certification
 - ◆ medical and other life-threatening applications
 - IEC 60601-1
- Long-Term availability
- Reusability
 - ◆ from previous project
 - e.g. new product generation

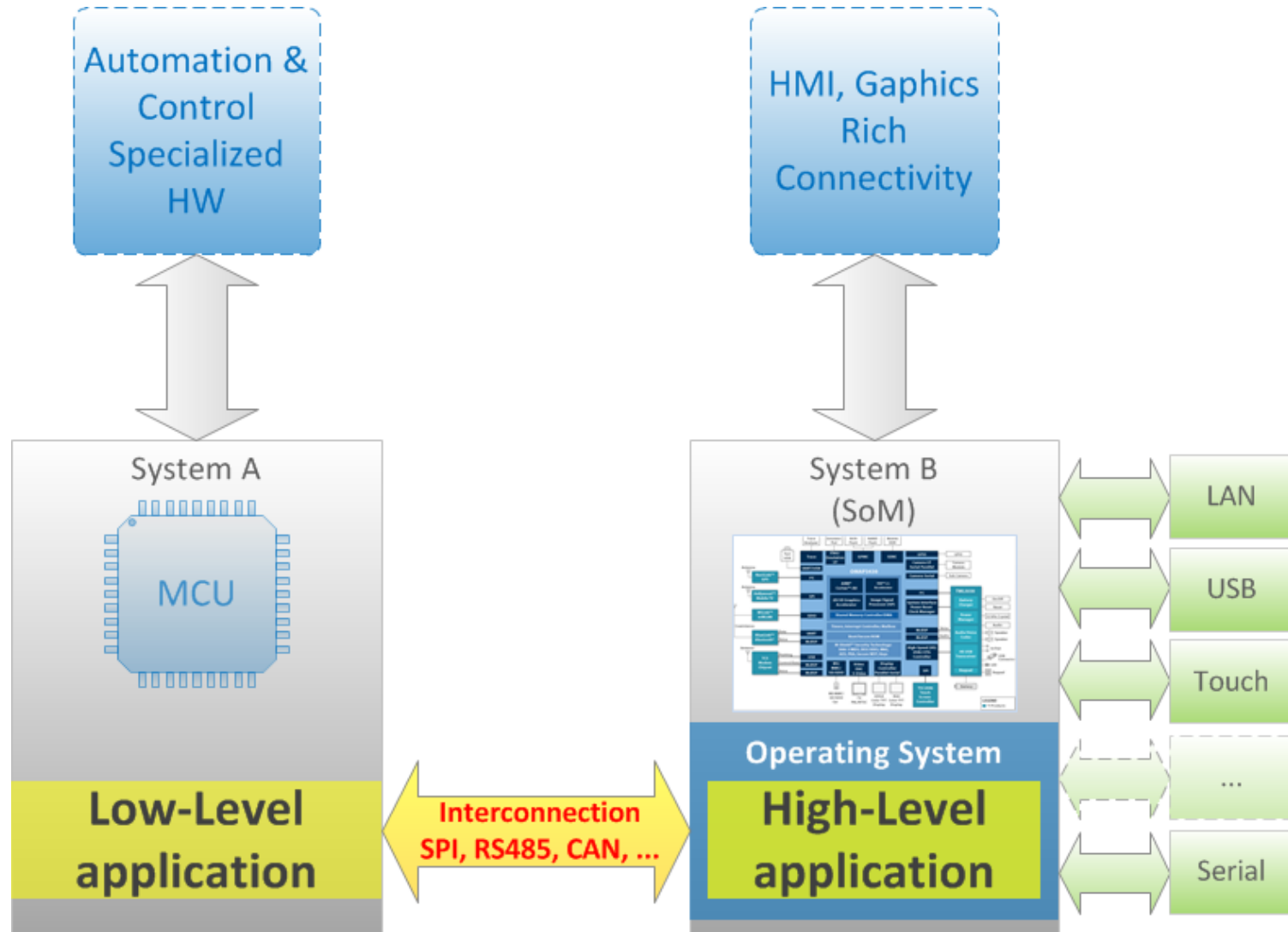


How to integrate all these challenges?

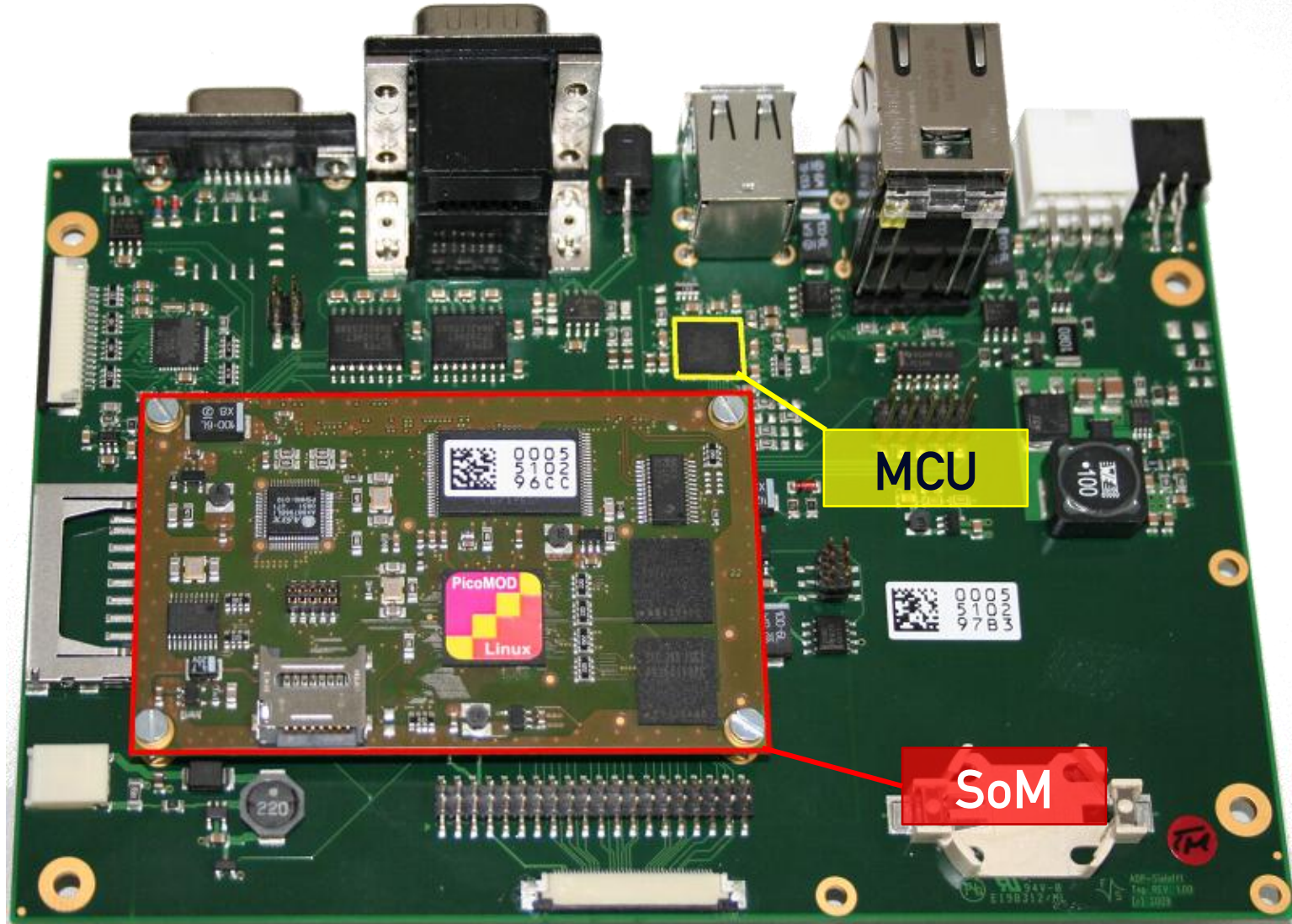


One single system is not able to meet all these requirements

Current approach

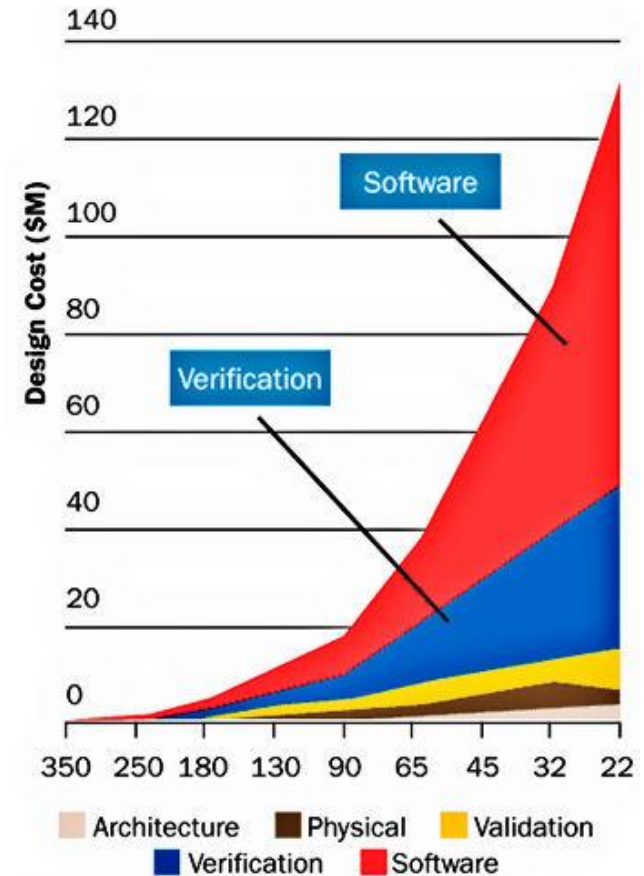


Current approaches in complex embedded systems



Downside of modular solution

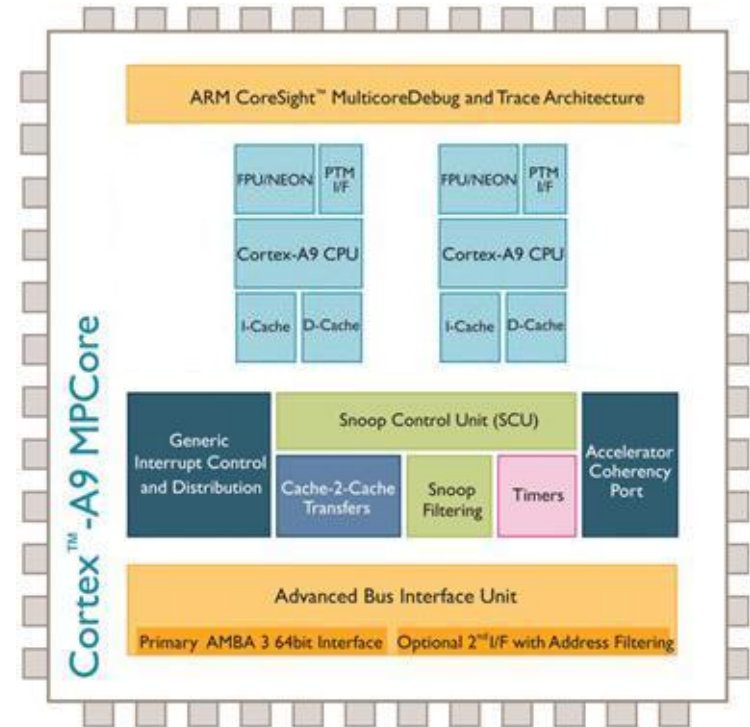
- 2 Systems \Rightarrow 2 sources of error
- Longevity ?
- Increasing in complexity
 - ◆ single system & Integration
 - ◆ different development environments
- Slower time to market
- Increased price
- Compactness
- Limited interconnection



SOURCE: IBS2009

Alternative system design?

- Use 2 MCUs on one PCB
 - ◆ still complex
 - ◆ unsuitable for small budgets
- Dual/Quad core SoC
 - ◆ 1 Core for HMI application
 - ◆ 1 Core for RT application
 - ◆ already available
 - ◆ same architecture
- Heterogeneous CPUs in one Chip
 - ◆ application processor + microcontroller in one chip
 - ◆ high performance for HMI
 - ◆ reduced complexity for RT

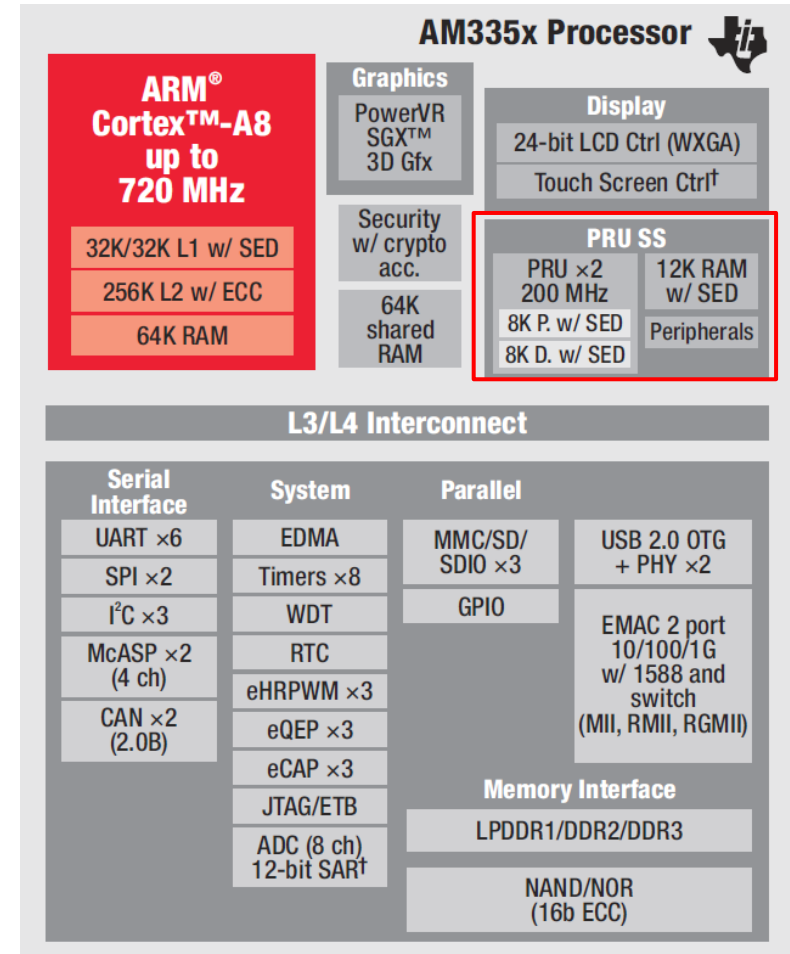
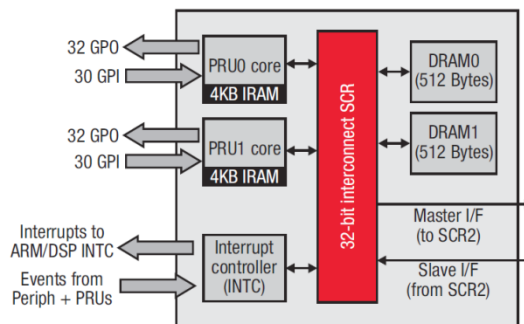


Heterogeneous CPUs in one Chip

- TI Sitara

- ◆ e.g. AM335x
- ◆ Cortex A8
- ◆ Programmable Real-Time Unit and Industrial Communication Subsystem (PRU-ICSS)

- 2x 32bit RISC cores
- special interface implementations
- EtherCAT, ProfiNET, PROFIBUS, ...



Heterogeneous CPUs in one Chip

- Freescale Vybrid (*VFxx*)

- ◆ Cortex A5

- ◆ Cortex M4

- scalable ultra-low-power mixed-signal MCU
 - widely used in embedded devices

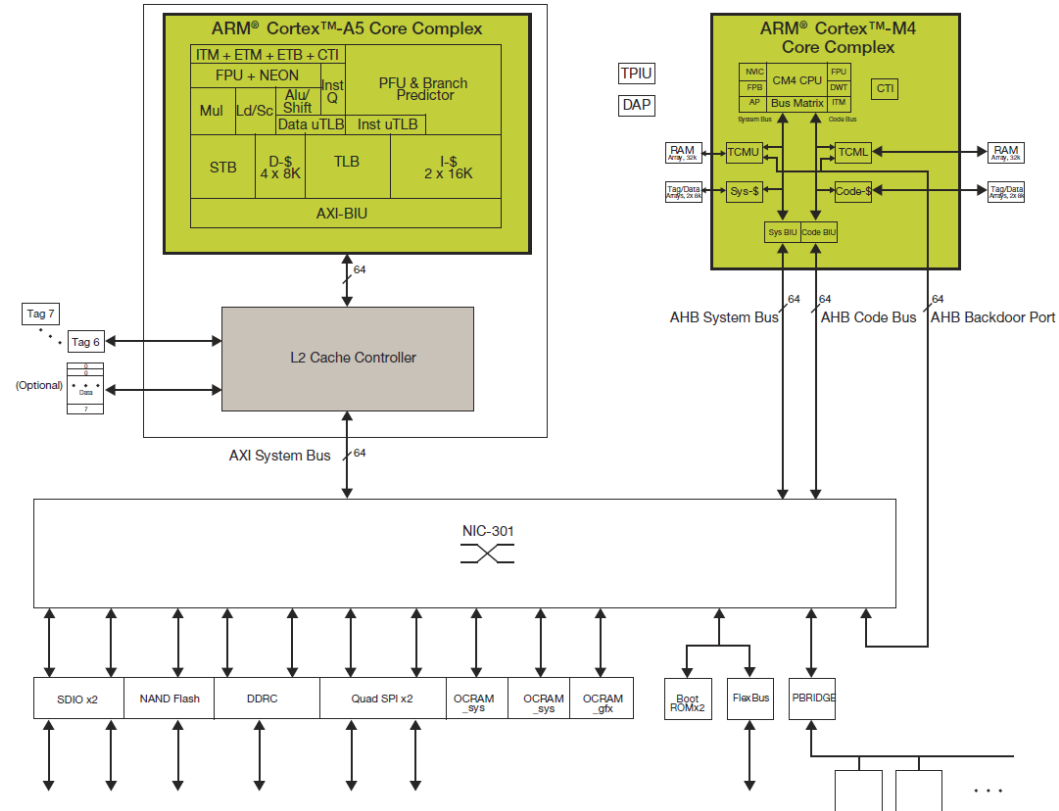
- OpenVG GPU

- Integrated ...

- ◆ power management
 - ◆ security and data integrity features

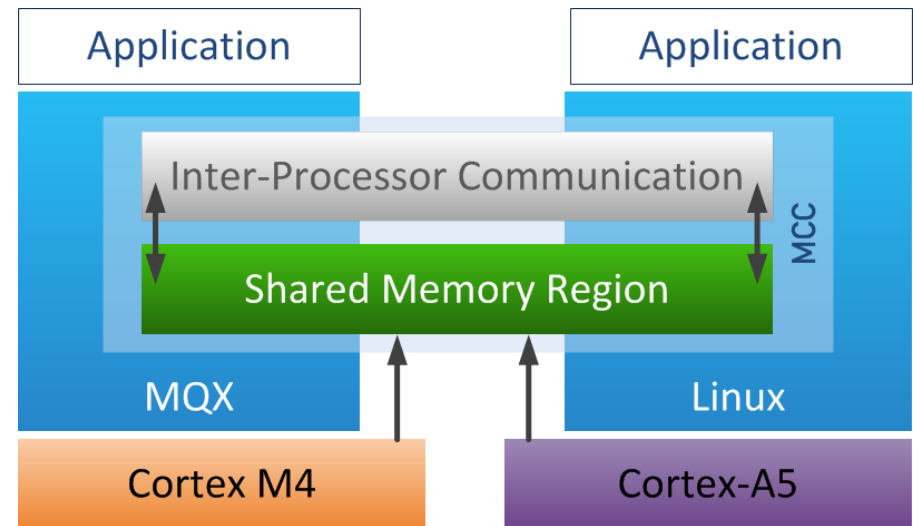
- >15 years availability

- Designed to combine HMI + RT explicitly

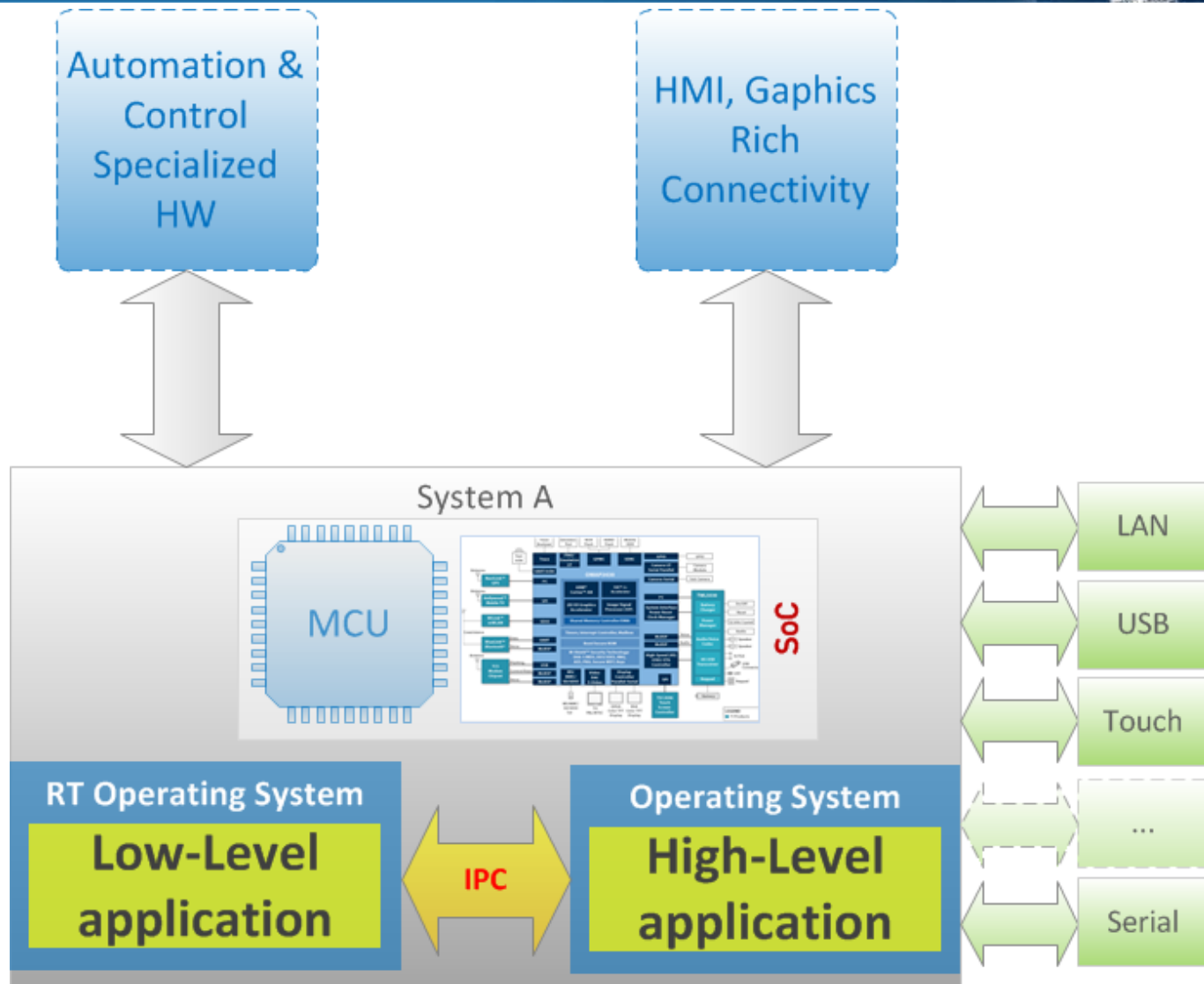


Software development

- **Direct usage**
 - ◆ RT app. on Cortex M4
 - ◆ HMI on Cortex A5
 - ◆ shared memory region
- **MQX & Linux**
 - ◆ hard real-time OS
 - ◆ certified (IEC 60601-1 / CFR 820.30)
 - ◆ extendable
 - ◆ small footprint and code size
 - ◆ integrated 3rd party SW (e.g. CanOpen)
 - ◆ Multicore Communication (MCC)
 - ◆ common development environment
 - DS-5 / Codewarrior
 - Debugging, Streamline Analyzer



HMI + RT embedded system design



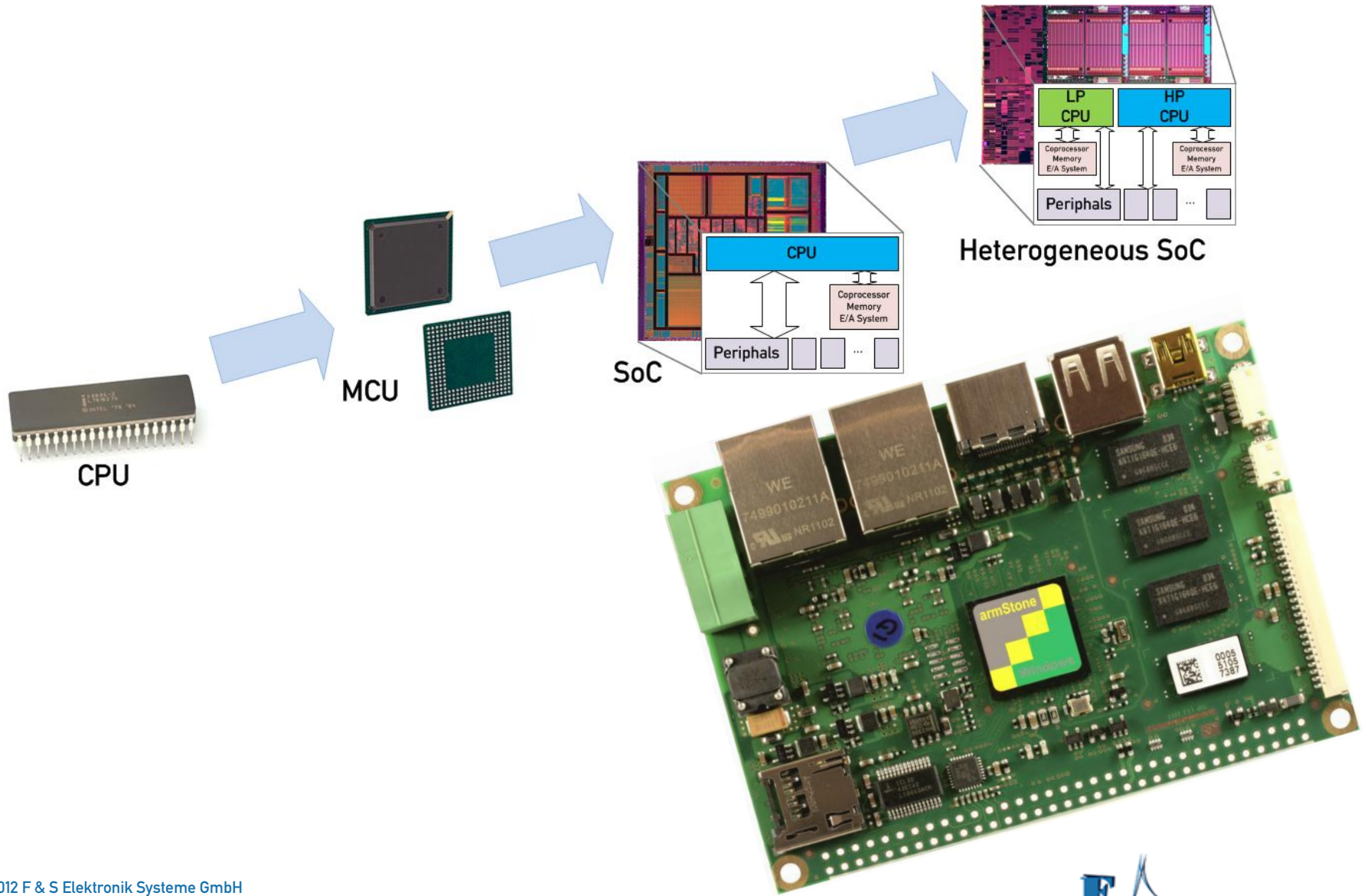
Modular vs. Integrated design



Modular design (MCU+SoM)	Integrated design (SBC with one SoC)
scalable, exchangeable	compactness
more flexible (no limitations in functionality)	easier to handle (SW and HW development)
	TTM, costs
	one supplier only (longevity, support)
high performance	power management

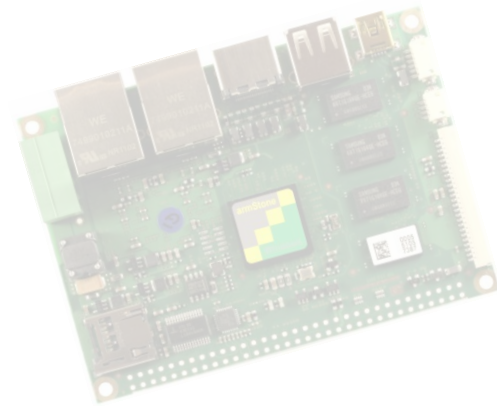
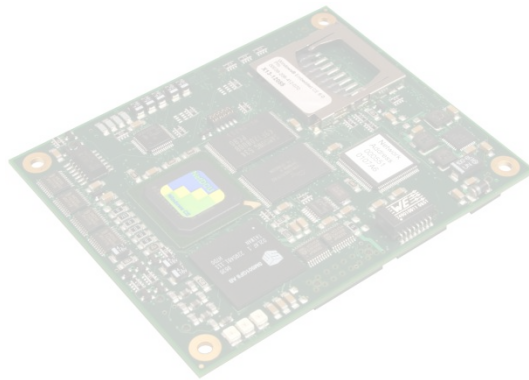
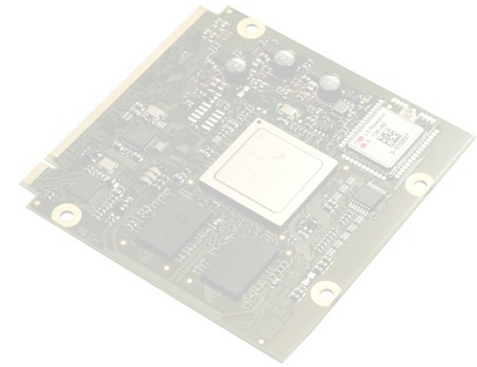
- Many embedded solutions require combination of rich **HMI + Real-Time capabilities**
- Modular design is today's solution
 - ◆ getting more complex
 - ◆ disadvantages in costs and TTM
 - ◆ still suitable for most systems
- Use Multiprocessor Core SoC
 - ◆ high performance
- SBC with heterogeneous CPUs in SoC (e.g. CortexA+CortexM)
 - ◆ compact and low powered design possible
 - ◆ real-time and fast boot-time
 - ◆ simplified development
 - ◆ quick and easy data sharing (shared memory)

Further step in embedded evolution?



Thank you

Questions?



kiepfer@fs-net.de