Embedded Systems

1 - Introduction

© Lothar Thiele

Computer Engineering and Networks Laboratory

Michele Magno

D-ITET Center for project-based Learning

ETH

Eidgenössische Technische Hochschule Zürich

Swiss Federal Institute of Technology Zurich

Organization

WWW: https://pbl.ee.ethz.ch/education/embedded-systems.html

Lecture: Michele Magno <michele.magno@pbl.ee.ethz.ch>; Lothar Thiele, thiele@ethz.ch;

Coordination: Seonyeong Heo (ETZ D97.7) <seoheo@ethz.ch>

References:

- *P. Marwedel:* Embedded System Design, Springer, ISBN 978-3-319-85812-8/978-3-030-60909-2, 2018/2021. ***You can find it also as open access!***
- *G.C. Buttazzo:* Hard Real-Time Computing Systems. Springer Verlag, ISBN 978-1-4614-0676-1, *2011.*
- *Edward A. Lee and Sanjit A. Seshia:* Introduction to Embedded Systems, A Cyber-Physical Systems Approach, Second Edition, MIT Press, ISBN 978-0-262-53381-2*, 2017.*

Sources: The slides contain ideas and material of J. Rabaey, K. Keuzer, M. Wolf, P. Marwedel, P. Koopman, E. Lee, P. Dutta, S. Seshia, and from the above cited books.

Organization Summary

- **Lectures** are held on Mondays from 14:15 to 16:00 in ETF C1 until further notice. Life streaming and slides are available via the web page of the lecture **from the previous year (some part will be new!!).** In addition, you find audio and video recordings of most of the slides as well as recordings of this years and last years life streams on the web page of the lecture.
- **Exercises:** The exercises are made available before the date of the exercise. During an exercise, teaching assistants will summarize the lecture material required to solve the exercise questions, give hints how to approach the solution, answer your questions and discuss the correct solution at the end of the exercise. The electronic version of the exercise questions and solutions will be placed online after the exercise has been conducted. Exercises will take place from 4pm - 6pm on Wednesdays and Fridays as indicated in the timetable.
- **Laboratory:** Laboratory exercises will be conducted in teams of 2 people. Due to the large number of students, there will be two successive ES-Lab sessions at the days of the ES-Lab (see timetable): Session A from 4pm - 6pm on Wednesday and session B from 4pm - 6pm on Friday. Both sessions will take place in rooms D61.1 and D96. If you still need to sign up you can do this in the lecture (preferred) or via email to the lecture coordinator.

Lecture Organization

Autumn Semester 2022	
227-0124-00L	Embedded Systems

Data for restriction on registration

Number of places	Current registration	Priority for the primary target group until	Waiting list until
250	250	25.09.2022	02.10.2022

134

Further Material via the Web Page (under development)

Lecture Slides

All lecture slides are available for download as a bundle:

- Embedded Systems lecture slides [single page format] ⤓
- Embedded Systems lecture slides [4on1 page format] ⤓

Lecture Recordings

Life Recordings Autumn 2021

The life recordings of the lectures in Autumn Semester are available at the following link: Embedded Systems Life Recordings AS 2021.

Life Recordings Autumn 2020

The life recordings of last years lecture are available at the following links:

1. Lecture 1: Chapters 1. Introduction and 2. Software Development
2. Lecture 2: Chapters 2. Software Development and 3. Hardware-Software Interface

Audio and Videos of Selected Chapters

Some of the chapters are documented via carefully recoreded videos. They contain some of the slides as well as audio explanations.

- 1. Introduction
- 2. Software Development
- 3. Hardware Software Interface

Exercises and Laboratory

Generic Documents	
Embedded System Companion	Supplementary Material
Remote Installation Instructions	
Documents for Lab 0	
Handout	Source (code)
Slides and videos	Solution (code and handout)
Documents for Lab 1	
Handout	Source (code)
Slides and videos	Solution (code and handout)
Documents for Lab 2	
Handout	Source (code)
Slides and videos	Solution (code and handout)
Documents for Lab 3	

When and where?

	When	Where
Lectures	Monday 14:15 - 16:00	ETF C1
Exercises	Wednesday 16:15 - 18:00 Friday 16:15 - 18:00	ETF E1 ETF E1
Labs	Wednesday 16:15 - 18:00 Friday 16:15 - 18:00	ETZ D61.1, ETZ D96.1 ETZ D61.1, ETZ D96.1

Date	Lecture	Exercise	Lab
26.09.2022	1. Introduction 2. Software Development		
28.09./30.09.2022			0. Prelab [SH]
03.10.2022	2. Software Development 3. Hardware-Software Interface (Processor vs Microcontroller)		

What will you learn?

- Theoretical foundations and principles of the analysis and design of embedded systems.
- Practical aspects of embedded system design, mainly software design.

The course has three components:

  - *Lecture:* Communicate principles and practical aspects of embedded systems.
  - *Exercise*: Use paper and pencil to deepen your understanding of analysis and design principles . **EXAMS** ☺
  - *Laboratory (ES-Lab)*: Introduction into practical aspects of embedded systems design. Use of state-of-the-art hardware and design tools.

Please read carfully!!

- https://pbl.ee.ethz.ch/education/embedded-systems.html

> Exercises and Laboratory
>
> We urgently ask all students to install required software for the labs on their own hardware. You can find the installation instructions on GitLab. We have tested this setup on PCs and Laptops with an USB port that run Windows 11, as well as Linux Mint and Linux Ubuntu 20.04 and 22.04. You are not allowed to enter ETZ D61.1 or ETZ D96.1 during the laboratory hours if you do not have an allocated slot.

Practical part of this course.

Discovery kit for IoT node with STM32U5 series

- Ultra-low-power STM32U585AII6Q microcontroller based on the **Arm® Cortex®-M33 core** with Arm® TrustZone®, 2 Mbytes of Flash memory and 786 Kbytes of SRAM, and SMPS in UFBGA169 package
- 512-Mbit Quad-SPI Flash memory, 64-Mbit Octo-SPI PSRAM, 256-Kbit I^2C EEPROM
- USB FS, Sink and Source power, 2.5 W power capability
- 802.11 b/g/n compliant Wi-Fi® module from MXCHIP
- Bluetooth® Low Energy from STMicroelectronics
- **MEMS sensors** from STMicroelectronics
  - 2 digital microphones
  - Relative humidity and temperature sensor
  - 3-axis magnetometer
  - 3D accelerometer and 3D gyroscope
  - Pressure sensor, 260-1260 hPa absolute digital output barometer
  - Time-of-flight and gesture-detection sensor
- Ambient-light sensor
- Authentication and security for peripherals and IoT devices from STMicroelectronics
- 2 user LEDs
- User push-button
- Reset push-button
- Board connectors
  - USB Type-C®
  - **ARDUINO® Uno V3 expansion connectors**
  - Camera module expansion connector
  - 2× STMod+ expansion connectors
  - Pmod™ expansion connector
- **Flexible power-supply options: ST-LINK USB V_{BUS}, USB connector, or external sources**
- On-board STLINK-V3E debugger/programmer with USB re-enumeration capability: mass storage, Virtual COM port, and debug port
- Comprehensive free software libraries and examples available with the STM32CubeU5 MCU Package
- Support of a wide choice of Integrated Development Environments (IDEs) including IAR Embedded Workbench®, MDK-ARM, and STM32CubeIDE

Be careful and please do not …

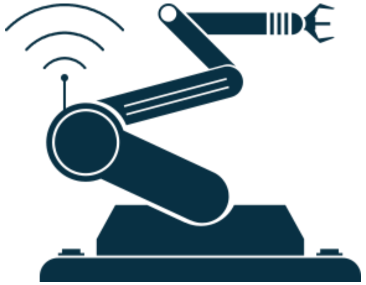
You have to return the board at the end!

Embedded Systems - Impact

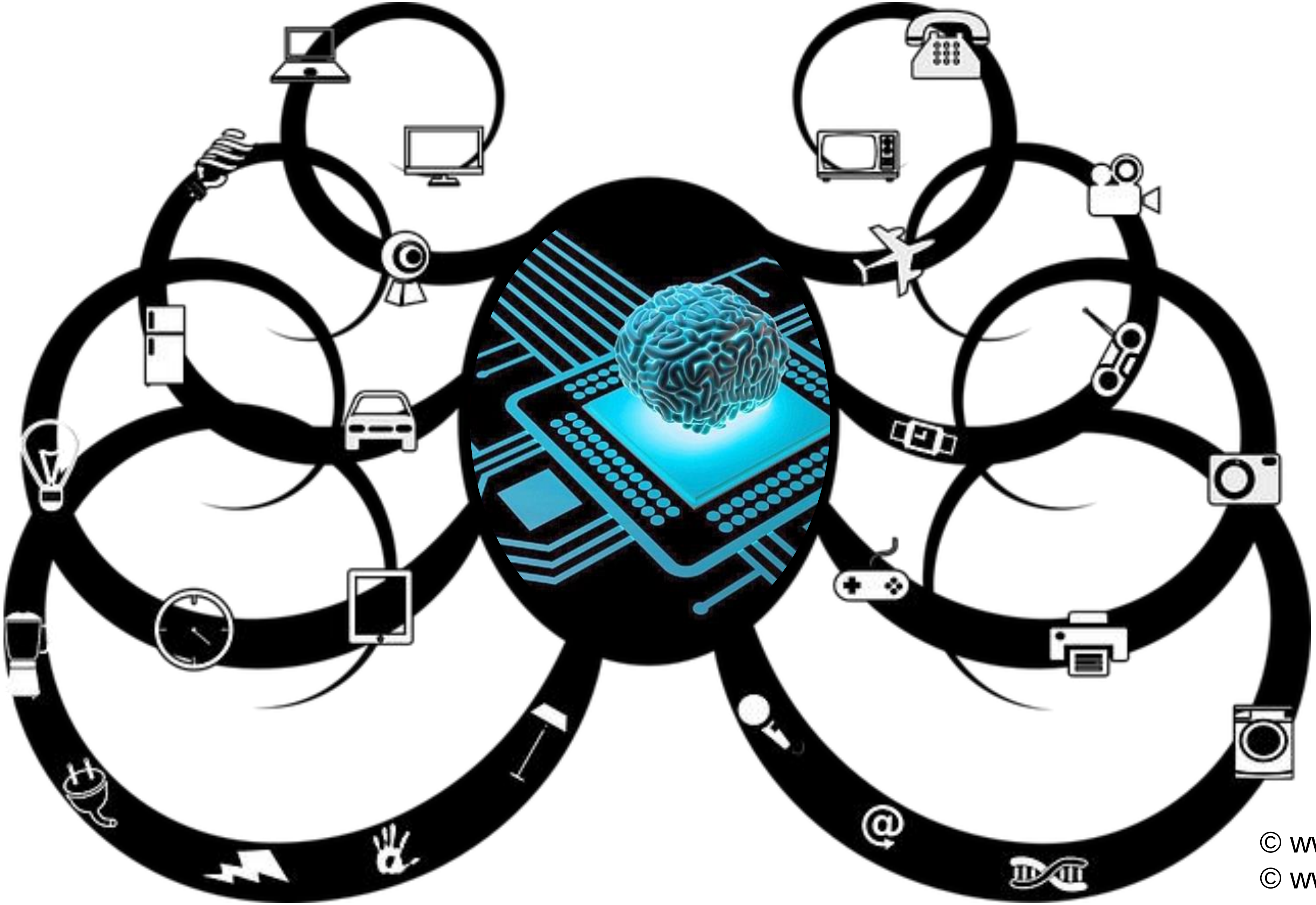
Embedded Systems

Embedded systems (ES) = **information processing systems embedded into a larger product**

Examples:

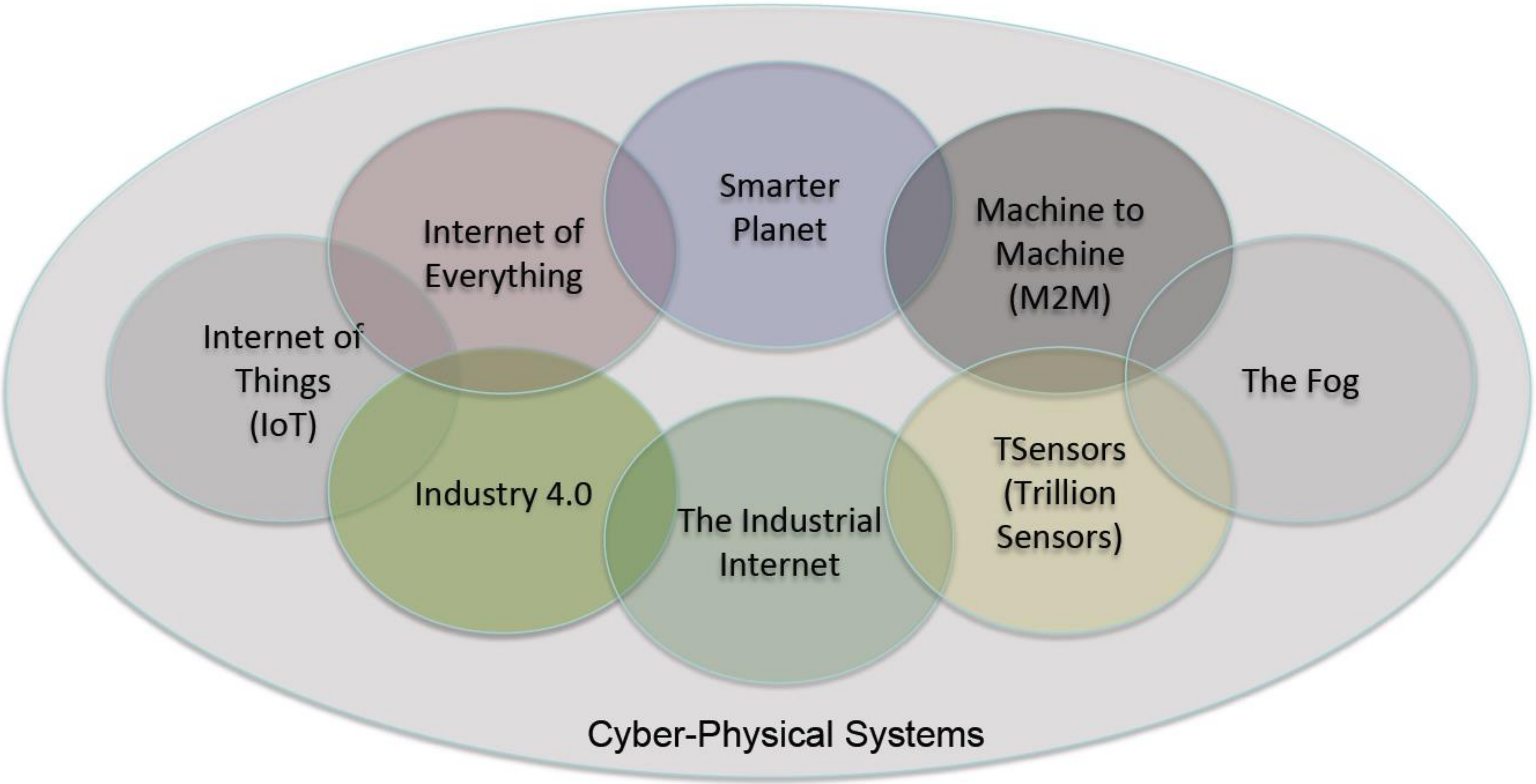


Often, the main reason for buying is not information processing



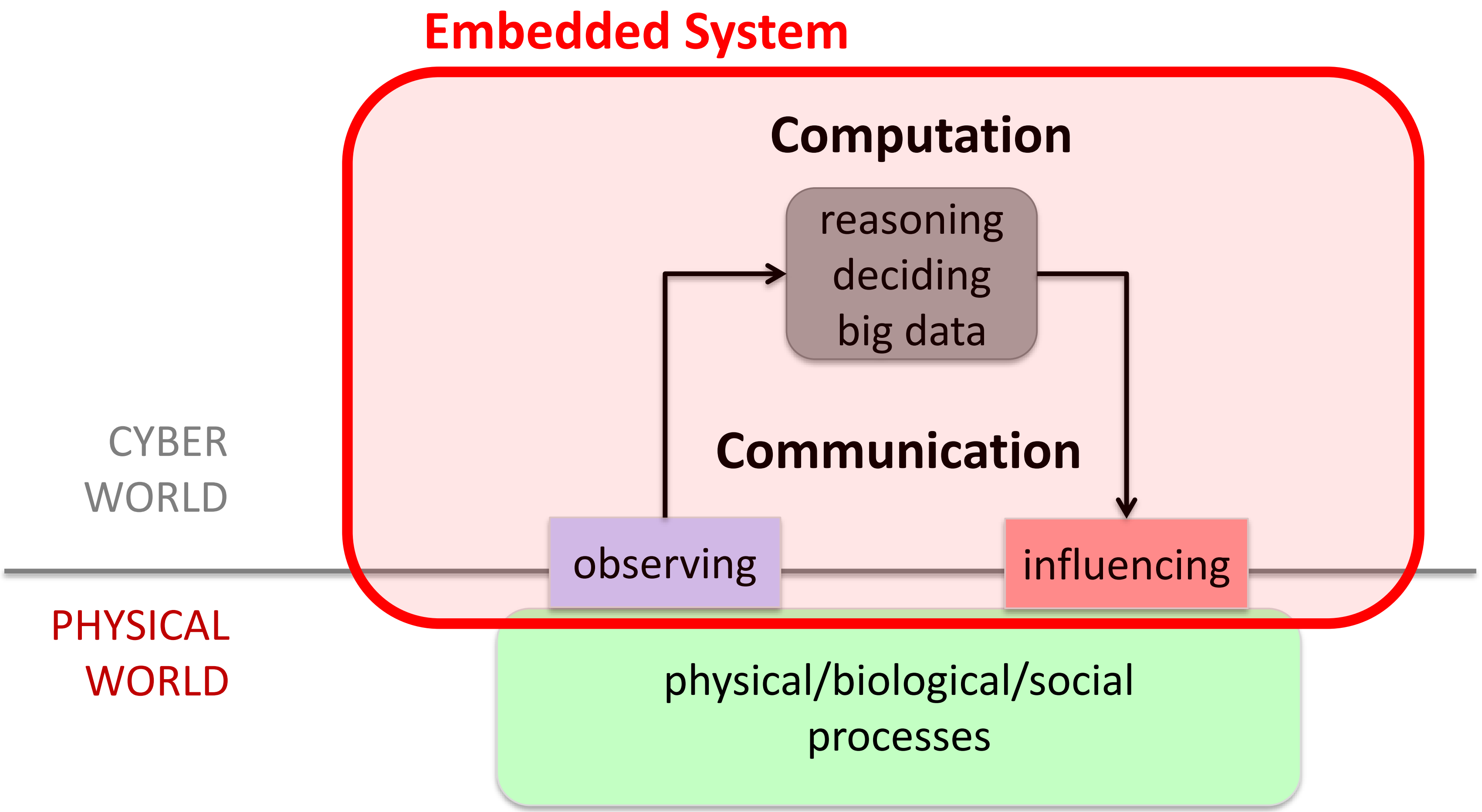
© www.braingrid.org
© www.openpr.com

Many Names – Similar Meanings



© Edward Lee

Embedded System



Use feedback to influence the dynamics of the physical world by taking smart decisions in the cyber world

Reactivity & Timing

Embedded systems are often reactive:

- Reactive systems must react to stimuli from the system environment :

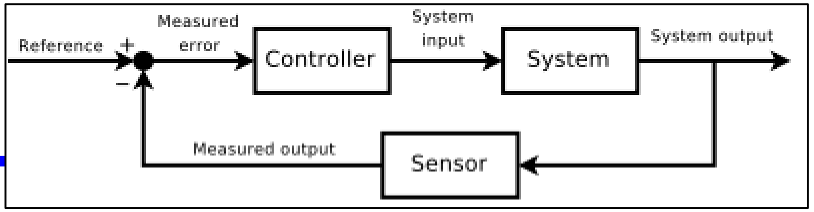
> *„A reactive system is one which is in continual interaction with is environment and executes at a pace determined by that environment“ [Bergé, 1995]*

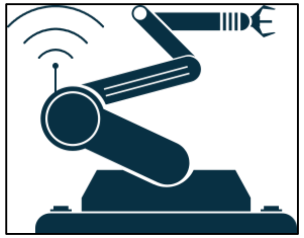
Embedded systems often must meet *real-time constraints:*

- For hard real-time systems, right answers arriving too late are wrong. All other time-constraints are called soft. A *guaranteed system response* has to be explained without statistical arguments.

> *„A real-time constraint is called hard, if not meeting that constraint could result in a catastrophe“ [Kopetz, 1997].*

Predictability & Dependability





CPS = cyber-physical system

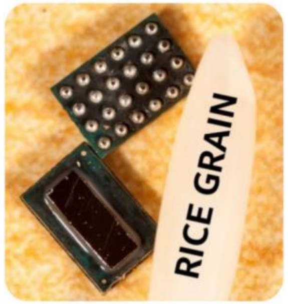
"It is essential to *predict* how a CPS is going to behave under any circumstances […] *before* it is deployed."[Maj14]

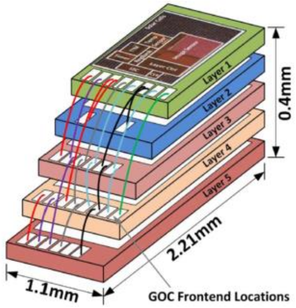
"CPS must *operate dependably*, safely, securely, efficiently and in real-time."[Raj10]

[Maj14] R. Majumdar & B. Brandenburg (2014). Foundations of Cyber-Physical Systems.
[Raj10] R. Rajkumar et al. (2010). Cyber-Physical Systems: The Next Computing Revolution.

Efficiency & Specialization

- Embedded systems must be *efficient:*
  - *Energy* efficient
  - *Code-size* and *data memory* efficient
  - *Run-time* efficient
  - *Weight* efficient
  - *Cost* efficient





Embedded Systems are often *specialized* towards a certain application or application domain:

- Knowledge about the expected behavior and the system environment at design time is exploited to *minimize resource usage* and to *maximize predictability and reliability*.

Comparison

Embedded Systems:

- Few applications that are known at design-time.
- Not programmable by end user.
- Fixed run-time requirements (additional computing power often not useful).
- Typical criteria:
  - cost
  - power consumption
  - size and weight
  - dependability
  - worst-case speed

General Purpose Computing

- Broad class of applications.
- Programmable by end user.
- Faster is better.
- Typical criteria:
  - cost
  - power consumption
  - average speed

Lecture Overview

Software

Hardware

1. Introduction to Embedded Systems
2. Software Development
3. Hardware-Software Interface
4. Programming Paradigms
5. Embedded Operating Systems
6. Real-time Scheduling
7. Shared Resources
8. Hardware Components
9. Power and Energy
10. Architecture Synthesis and System Integration

Hardware-Software

Agenda Embedded Systems

DATE	Topic
26.09 / Magno	Lecture 1: Chapters 1. Introduction
03.10/ Magno	Lecture 2: Chapters 2. Software Development and Chapter 3. Hardware-Software Interface (Processor VS MCU, Core ARM cortex-M 33,
10.10 / Magno	Lecture 3: Chapter 3 Hardware-Software Interface (Clocks, Memory, Interrupts of ARM Cortex M33)
17.10 / Magno	Lecture 4: Chapter 3 Hardware-Software Interface (Serial Interface)
24.10 / Magno	Lecture 5: Chapter 3 Hardware-Software Interface (ADC/ DAC/Timers/PWM)
31.10 / Dr. Heo	Summary, Labs, exercises
07.11 / Thiele	Lecture 6: Chapter 4 Programming Models and Chapter 5 Operating systems
14.11 / Thiele	Lecture 7: Operating Systems and. Chapter 6 Aperiodic and Periodic Scheduling
21.11 / Thiele	Lecture 8 L Chapter 6. Aperiodic and Periodic Scheduling
28.11 / Thiele	Lecture 9: Chapter 7 Shared Resources
05.12 / Magno	Lecture 11: Chapter 8 Power and Energy
12.12 / Michele	Lecture 12: Chapter 9 Sensing; energy considerations, data rates, buffering, spatial vs. temporal density.
19.12 / Michele	Lecture 13: Chapter 10 Architecture Synthesis and System Integration

Components and Requirements by Example

Components and Requirements by Example
- Hardware System Architecture -

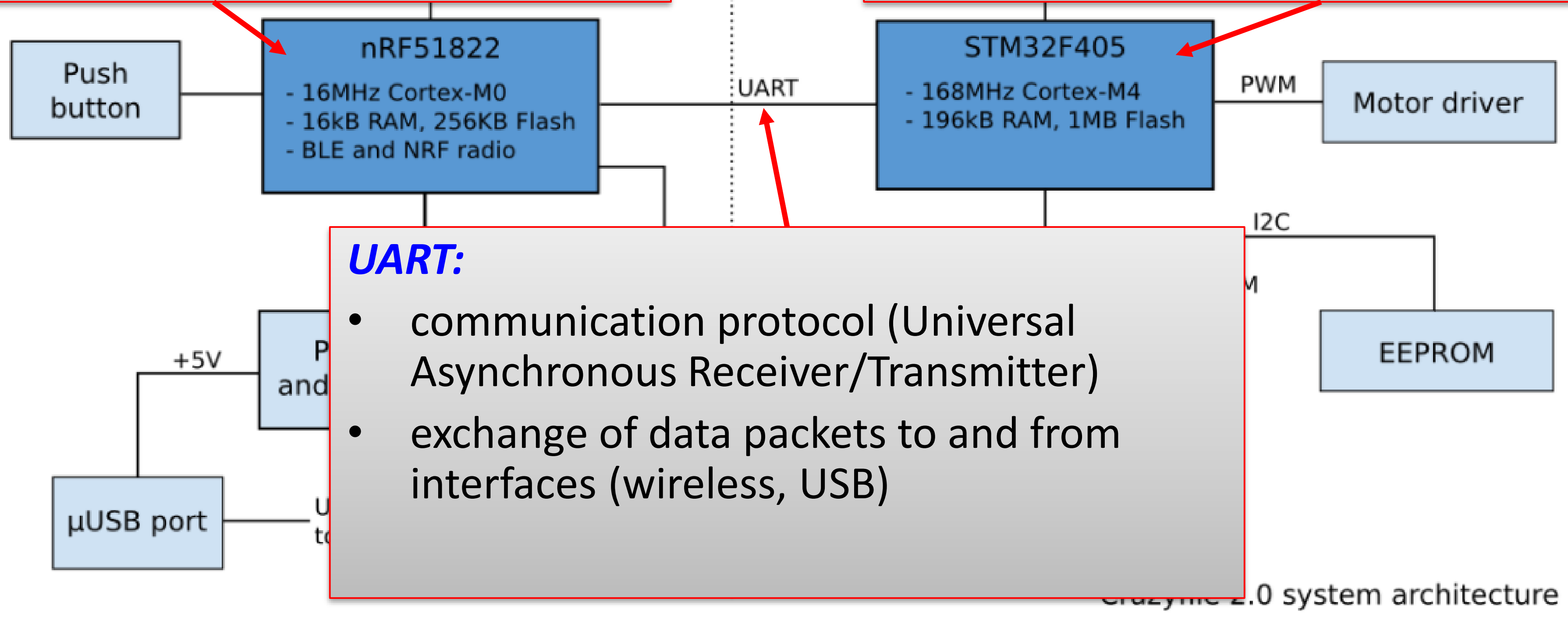
High-Level Block Diagram View

low power CPU

- enabling power to the rest of the system
- battery charging and voltage measurement
- wireless radio (boot and operate)
- detect and check expansion boards

higher performance CPU

- sensor reading and motor control
- flight control
- telemetry (including the battery voltage)
- additional user development
- USB connection



UART:

- communication protocol (Universal Asynchronous Receiver/Transmitter)
- exchange of data packets to and from interfaces (wireless, USB)

High-Level Block Diagram View

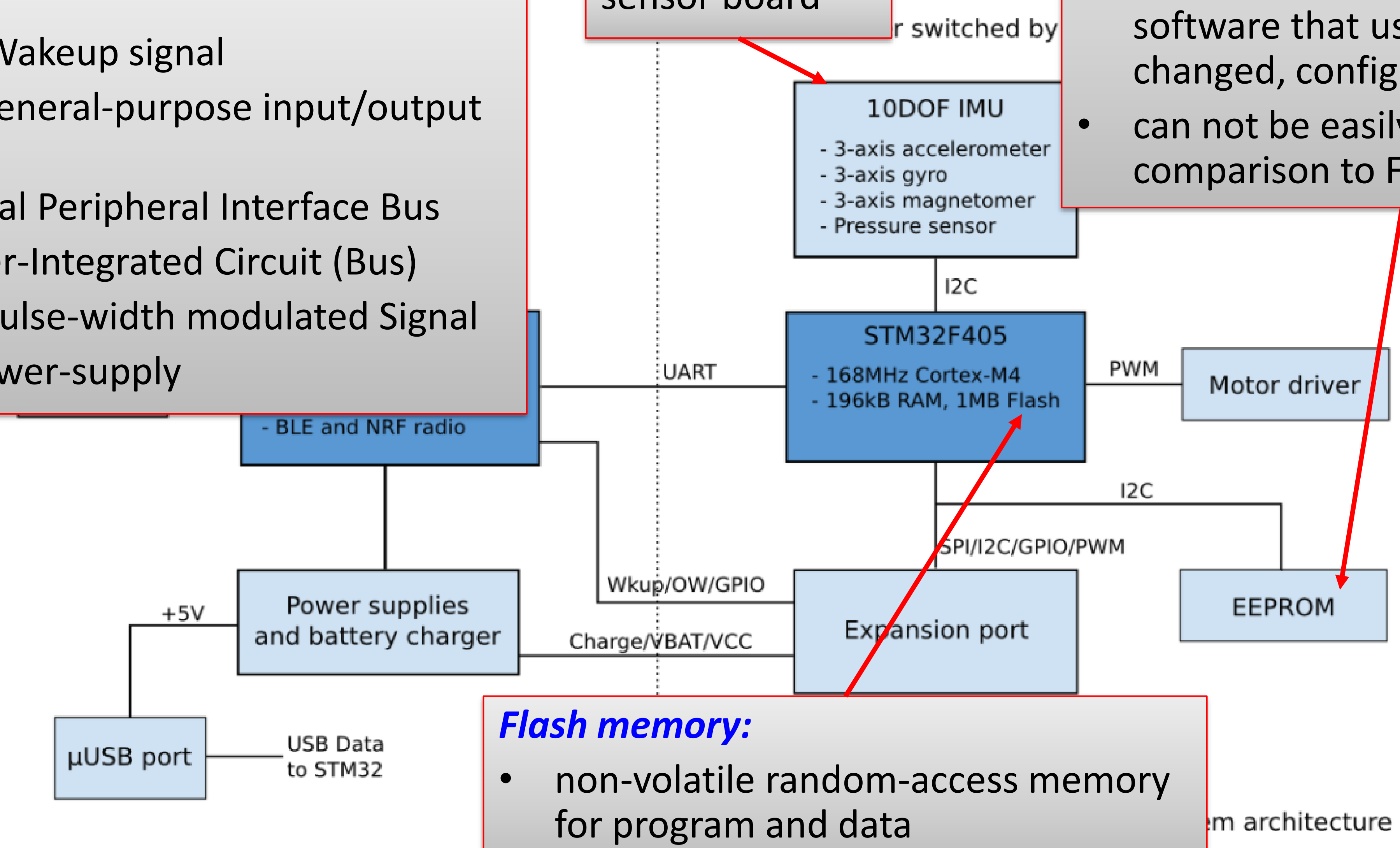
Acronyms:

- Wkup: Wakeup signal
- GPIO: General-purpose input/output signal
- SPI: Serial Peripheral Interface Bus
- I2C: Inter-Integrated Circuit (Bus)
- PWM: Pulse-width modulated Signal
- VCC: power-supply

EEPROM:

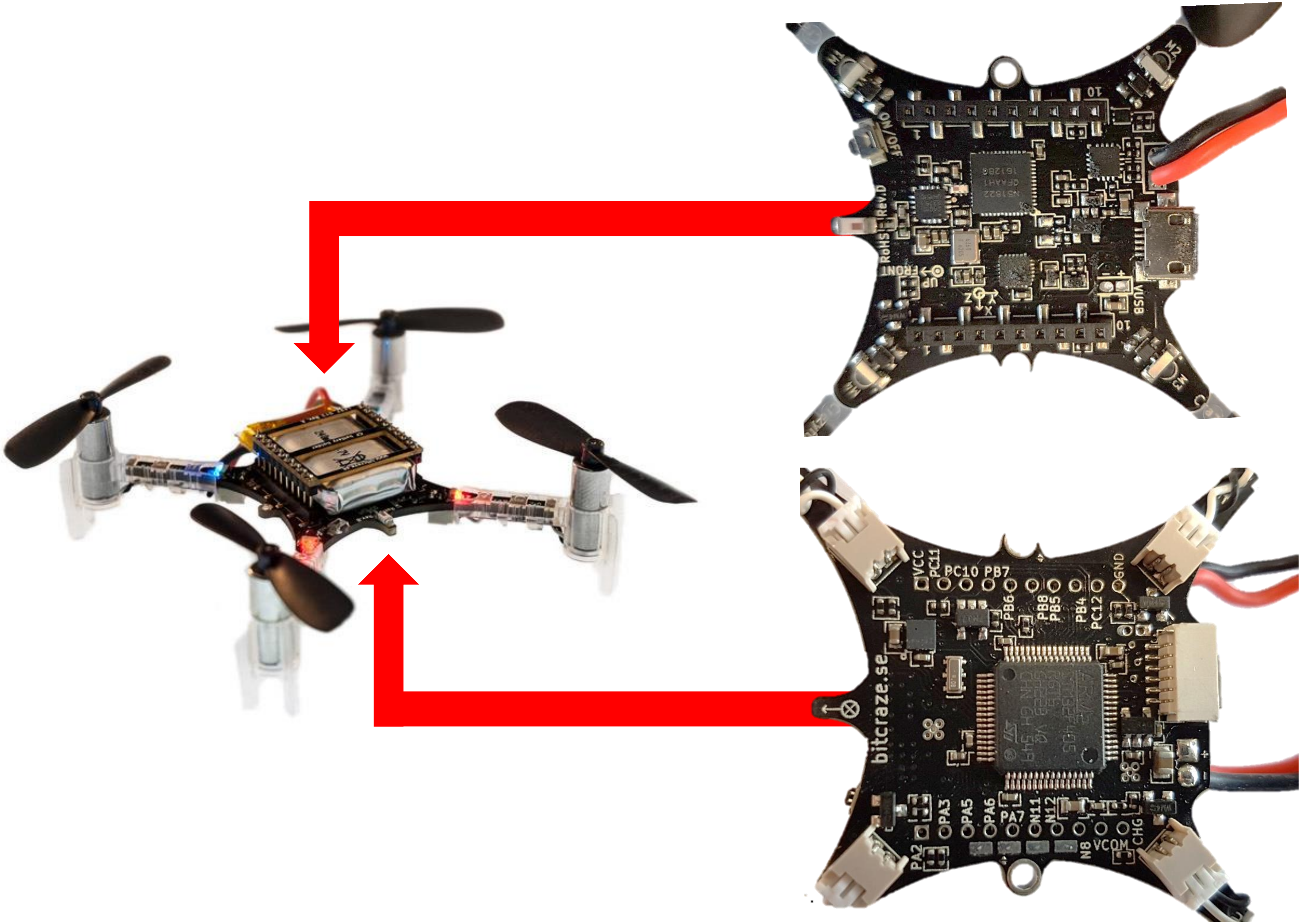
- electrically erasable programmable read-only memory
- used for firmware (part of data and software that usually is not changed, configuration data)
- can not be easily overwritten in comparison to Flash

sensor board



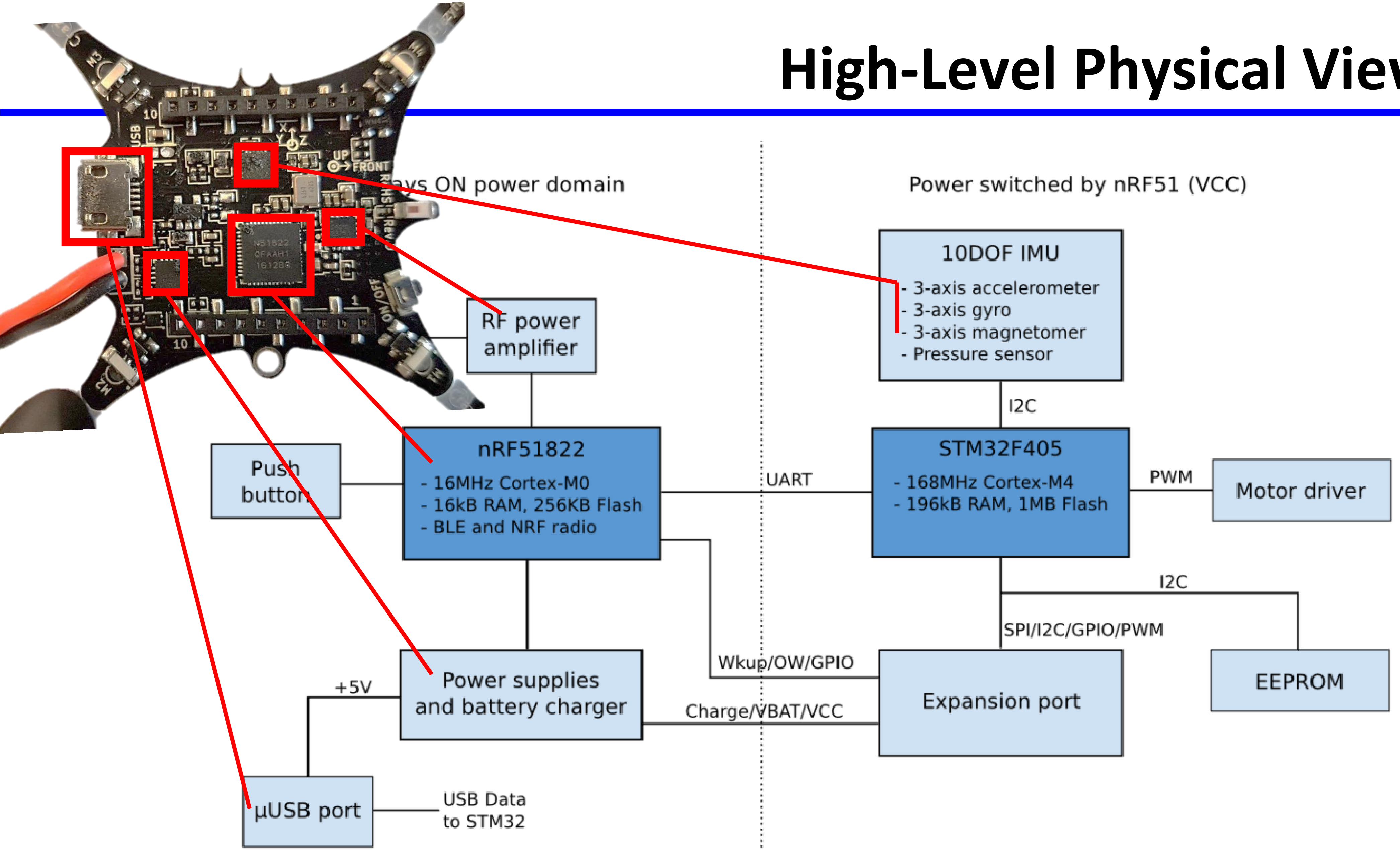
Flash memory:

- non-volatile random-access memory for program and data



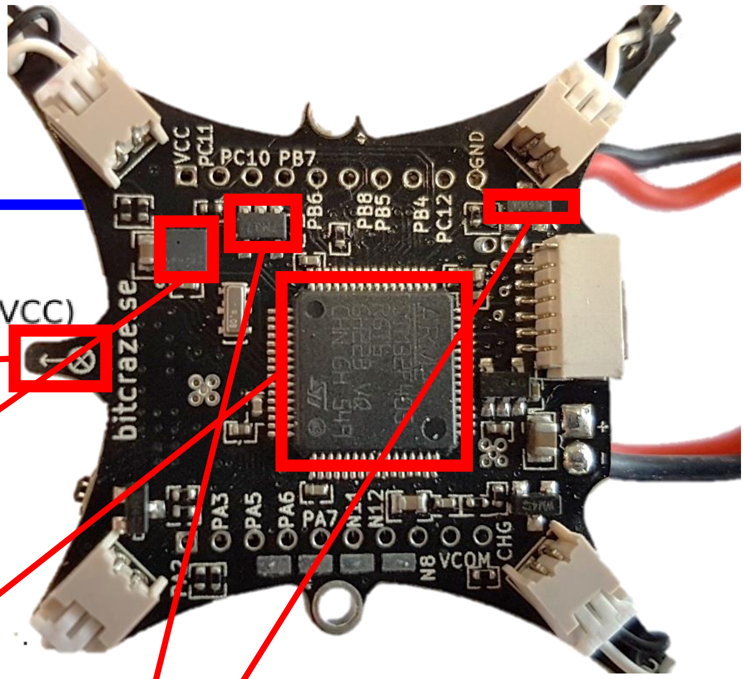
bitcraze.se
VCC
PC11
PC10 PB7
PB6
PB8
PB5
PB4
PC12
GND
PA2
PA3
PA5
PA6
PA7
VCOM
CHG
VUSB
FRONT
UP

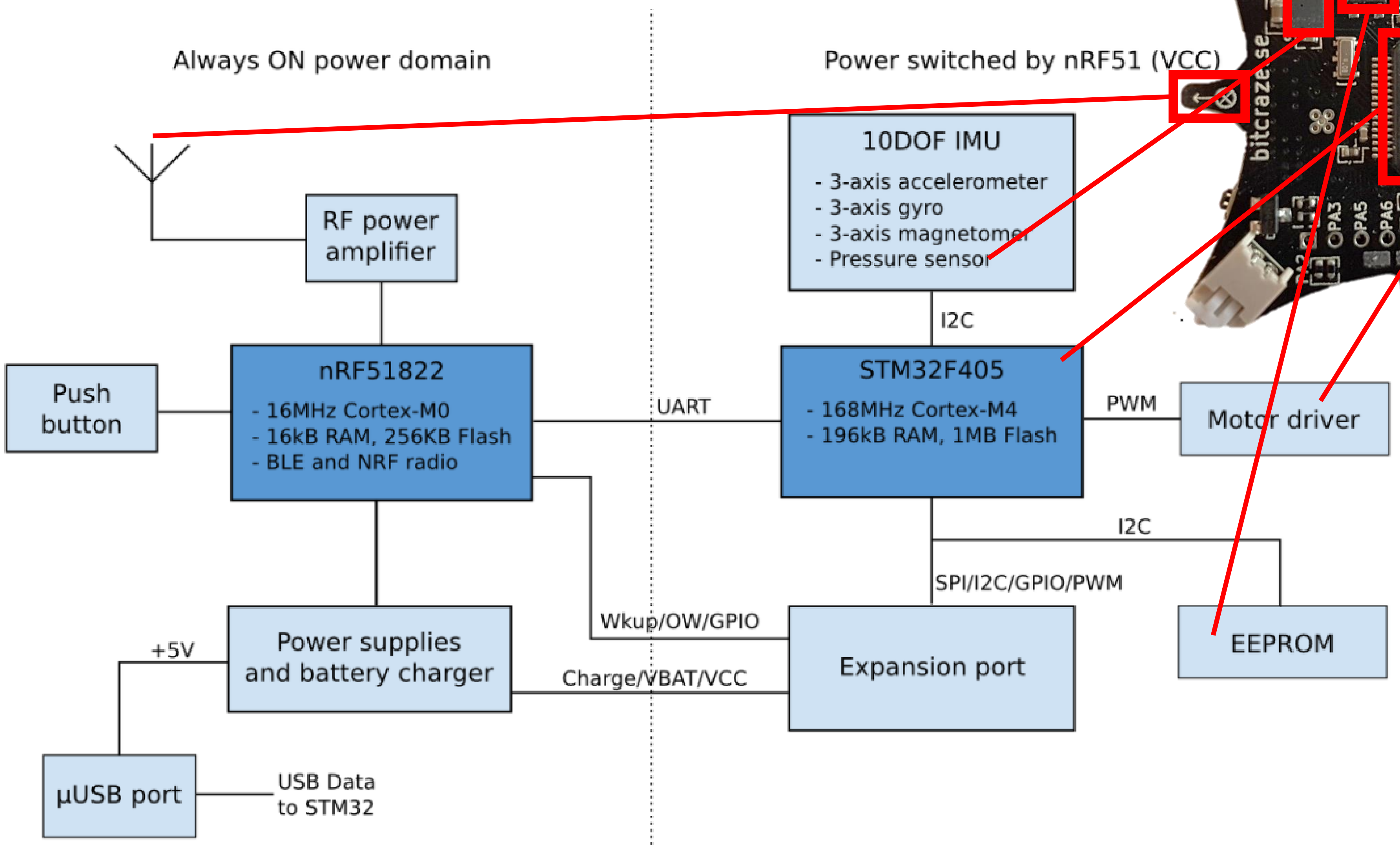
High-Level Physical View



Crazyflie 2.0 system architecture

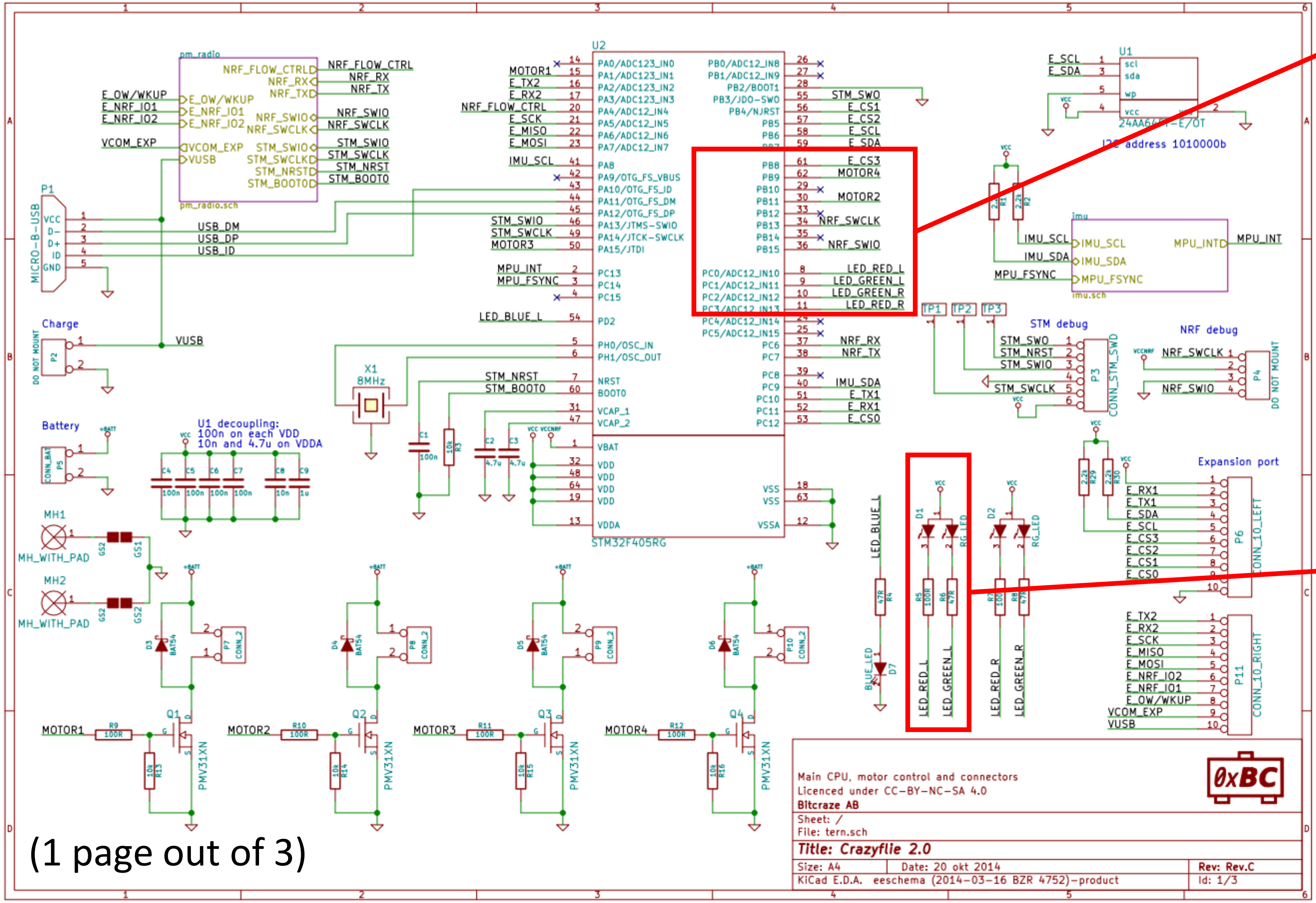
High-Level Physical View

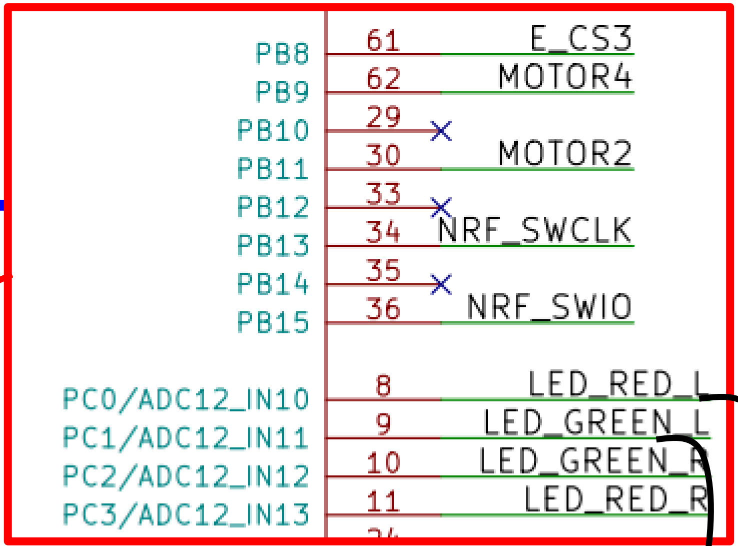


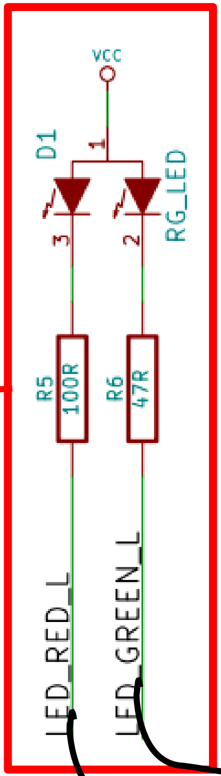


Crazyflie 2.0 system architecture

Low-Level Schematic Diagram View



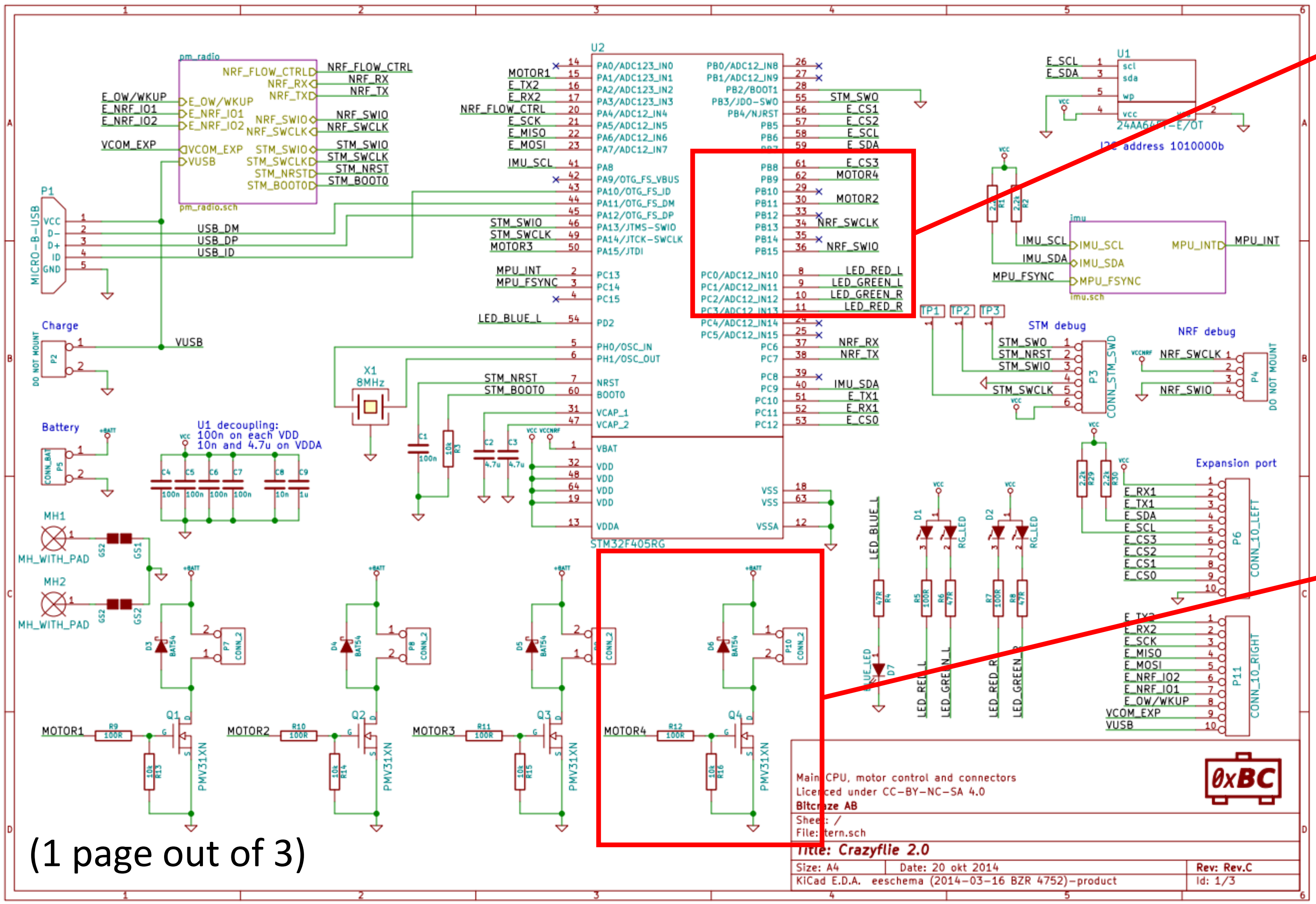


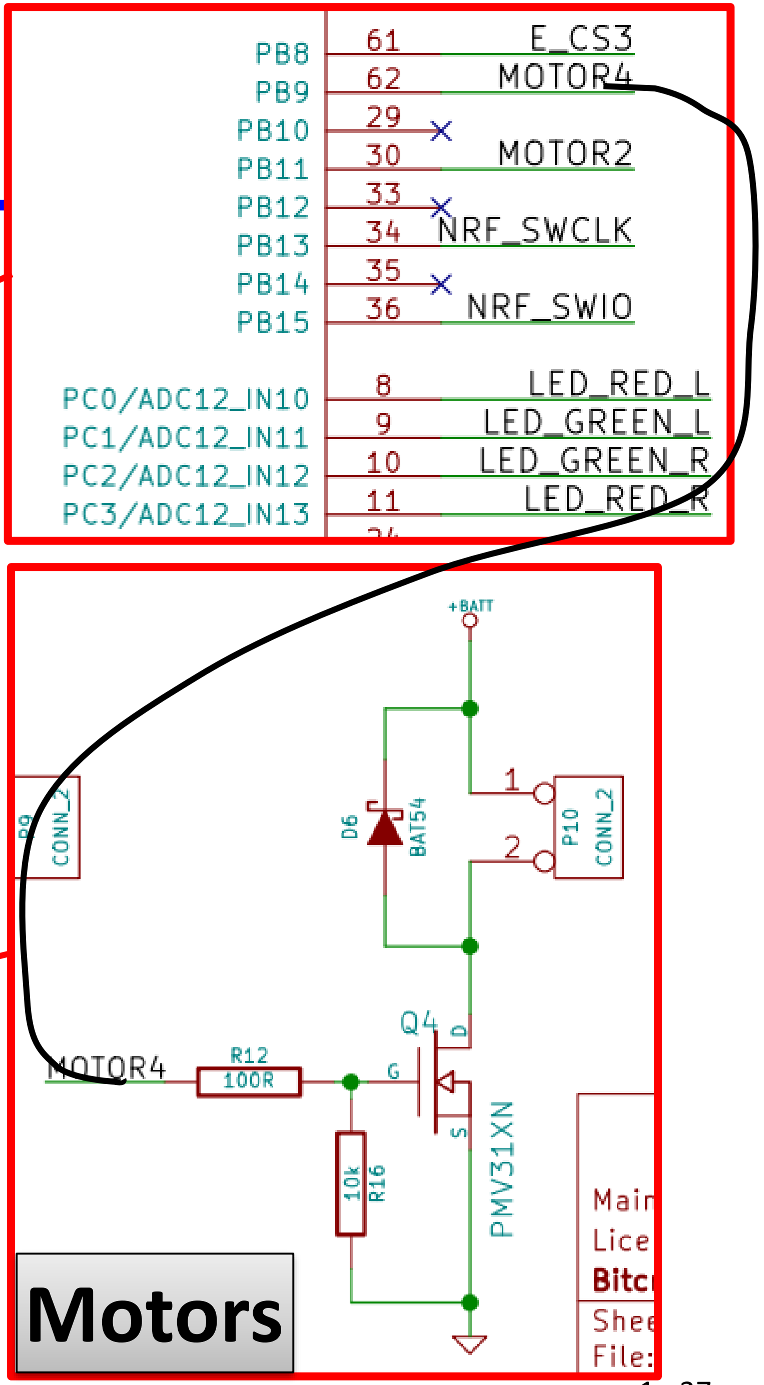


LEDs

(1 page out of 3)

Low-Level Schematic Diagram View

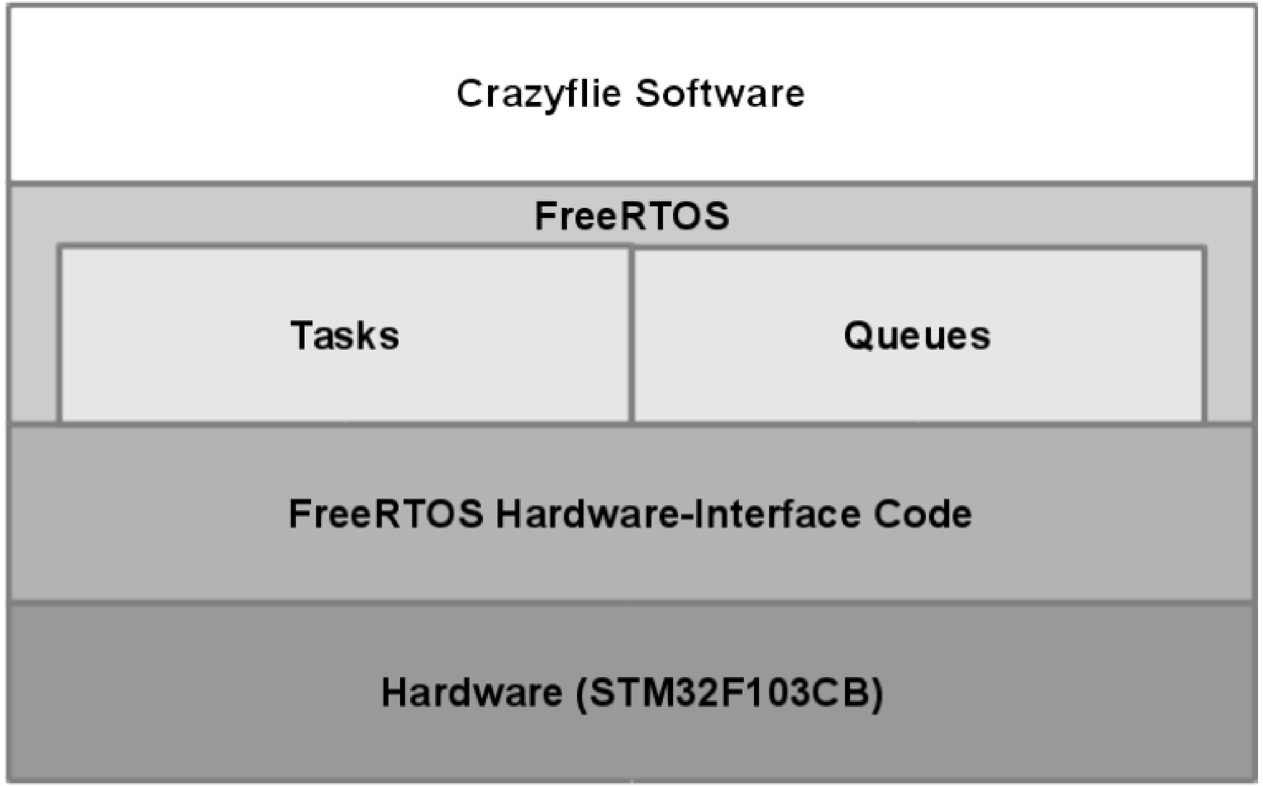




(1 page out of 3)

High-Level Software View

- The software is built on top of a *real-time operating system* “FreeRTOS”.
- We will use the same operating system in the ES-Lab … .



High-Level Software View

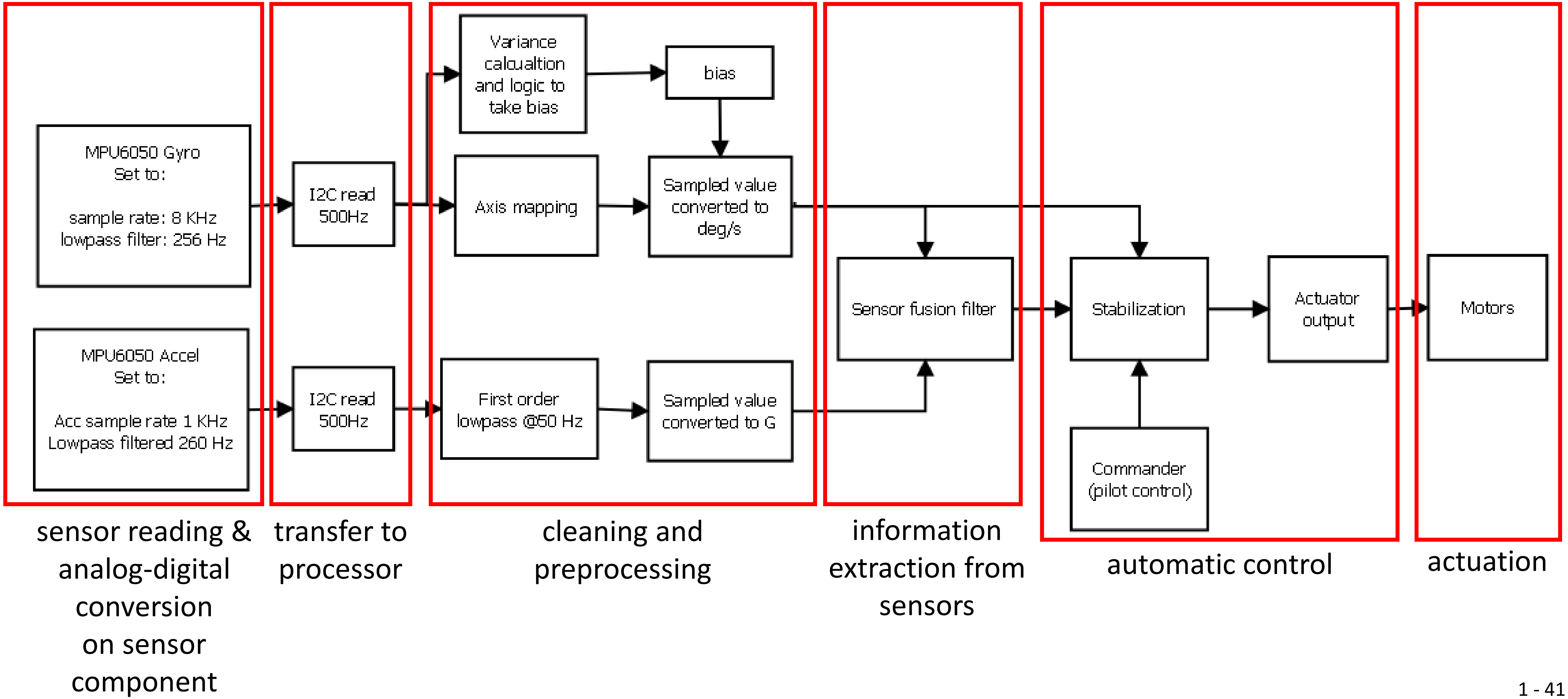
The *software architecture* supports

- *real-time tasks* for motor control (gathering sensor values and pilot commands, sensor fusion, automatic control, driving motors using PWM (pulse width modulation, …) but also

- *non-real-time tasks* (maintenance and test, handling external events, pilot commands, …).

High-Level Software View

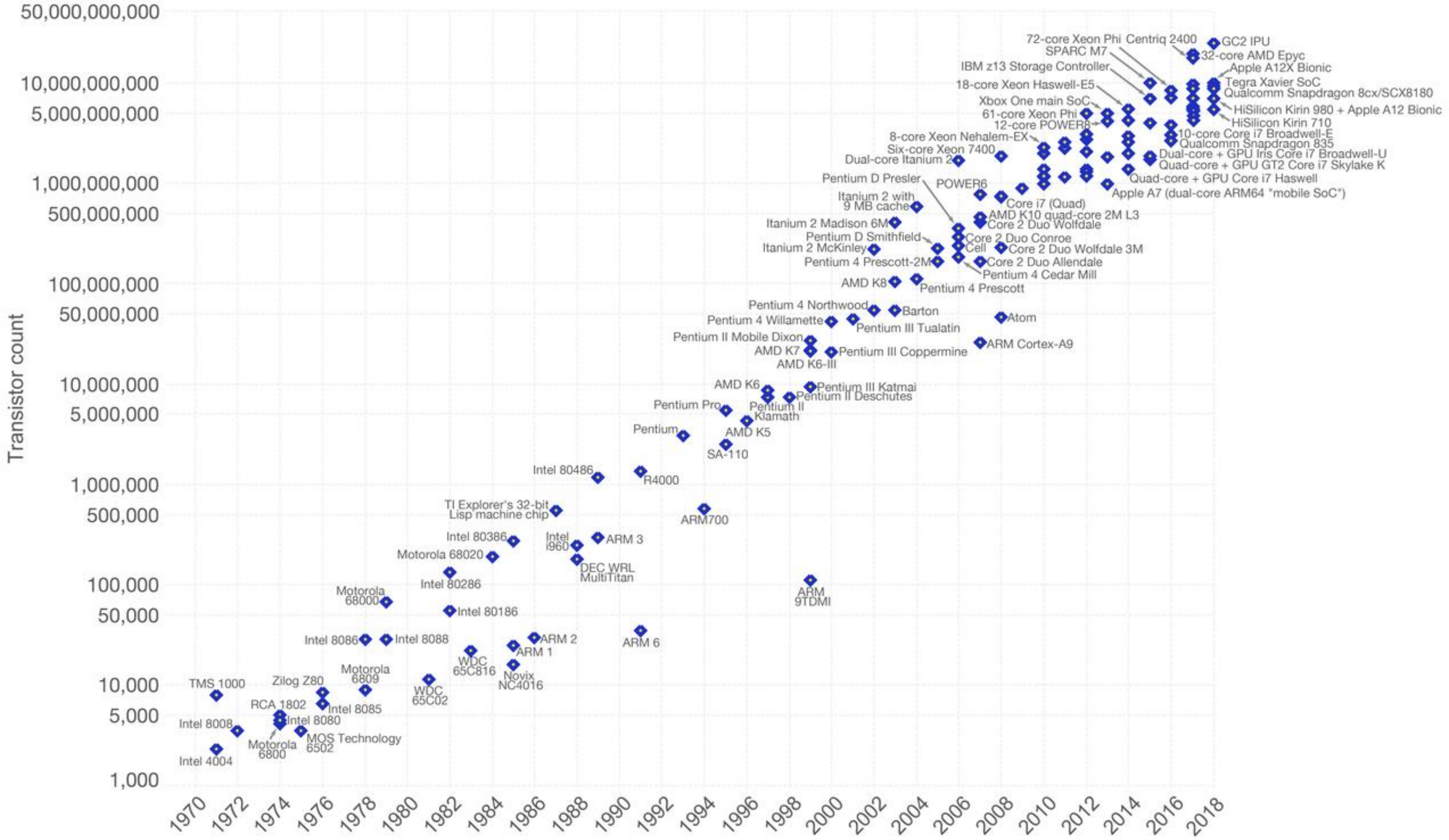
Block diagram of the stabilization system:



Components and Requirements by Example
- Processing Elements -

What can you do to increase performance?

From Computer Engineering



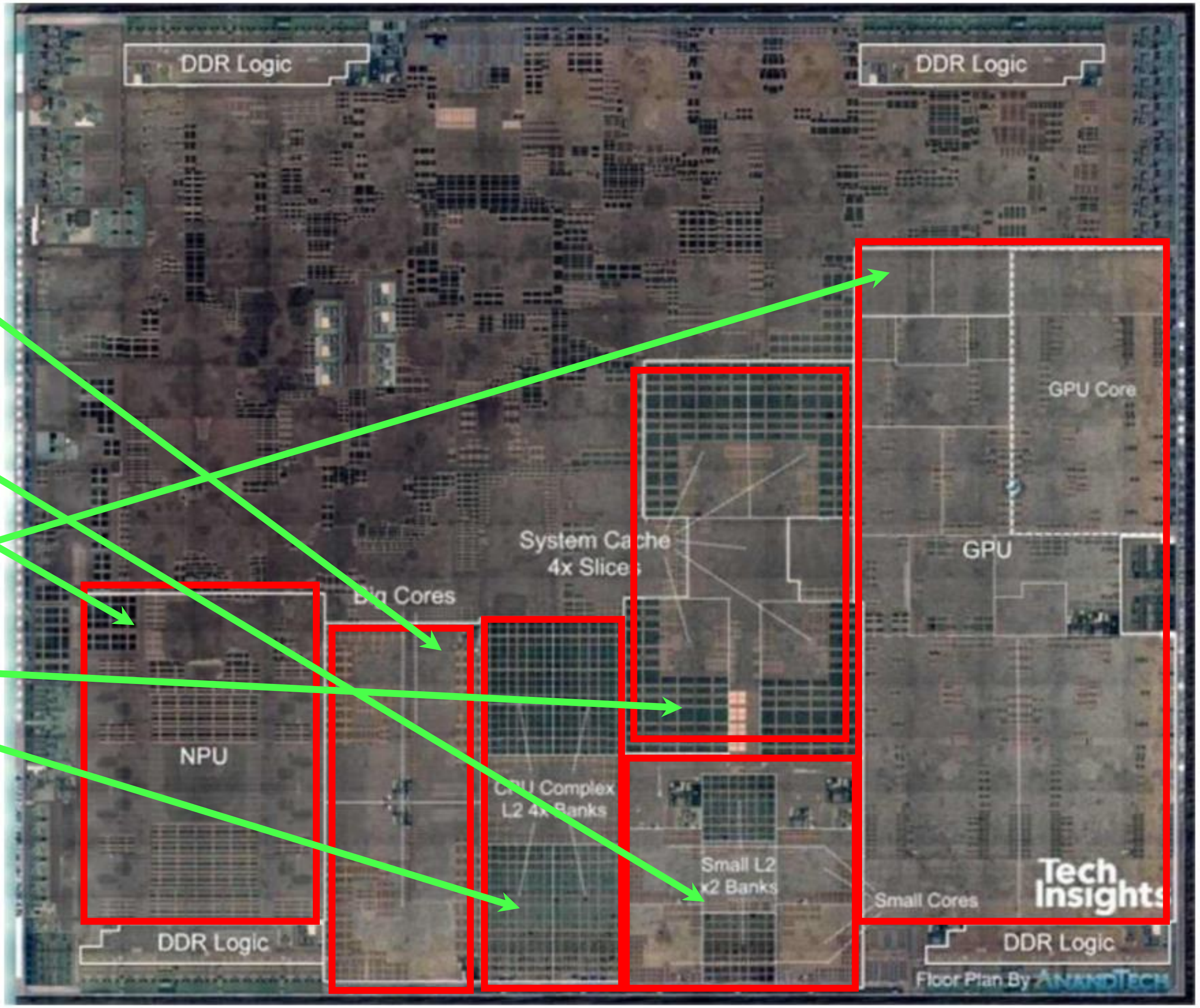
Data source: Wikipedia (https://en.wikipedia.org/wiki/Transistor_count)
The data visualization is available at OurWorldinData.org. There you find more visualizations and research on this topic.

Licensed under CC-BY-SA by the author Max Roser.

From Computer Engineering

iPhone Prozessor A12

- 2 processor cores
  - high performance
- 4 processor cores - less performant
- Acceleration for Neural Networks
- Graphics processor
- Caches



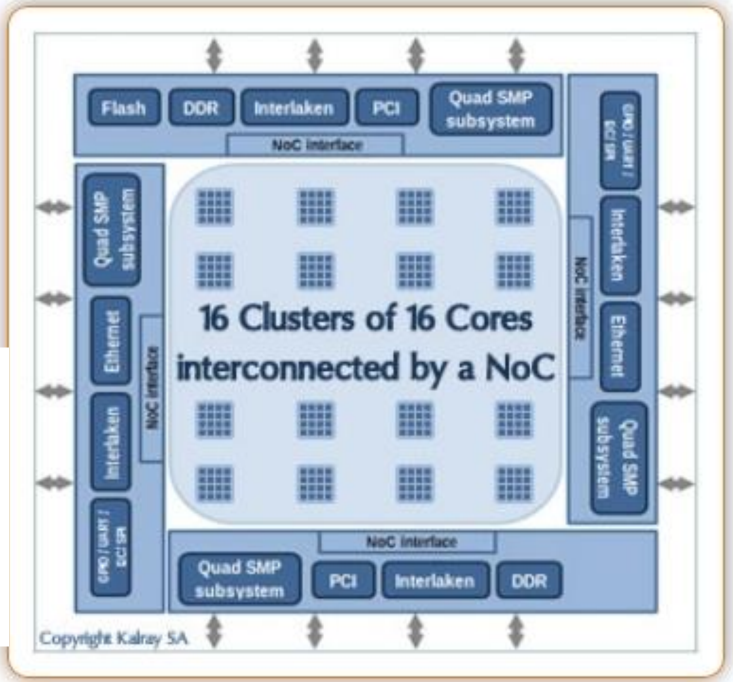
What can you do to decrease power consumption?

Embedded Multicore Example

Trends:

- Specialize multicore processors towards real-time processing and low power consumption (parallelism can decrease energy consumption)
- Target domains:

Core Generation	Number of Processing Cores	GFLOPS/W	GOPS/W
Andey	256	25	75
Bostan (2014)	256	50	80
Coolidge (2015)	64/256/1024	75	115

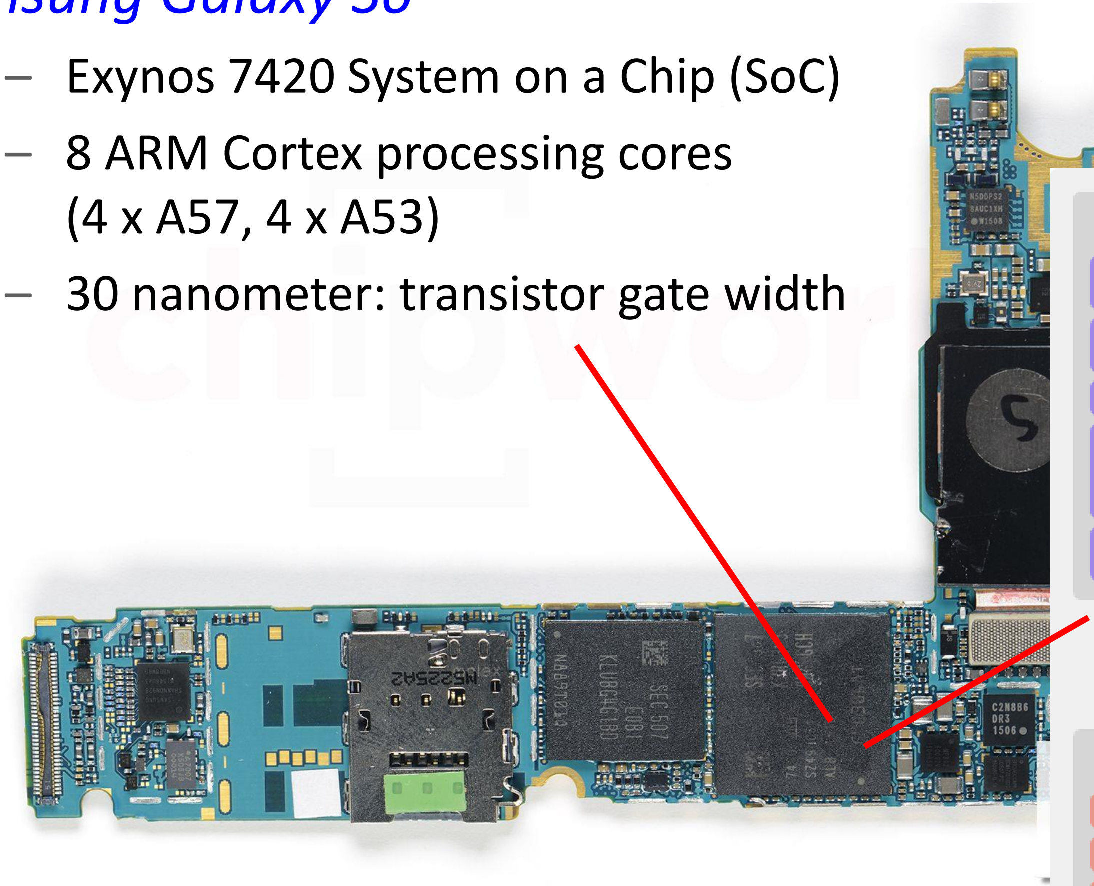


Why does higher parallelism help in reducing power?

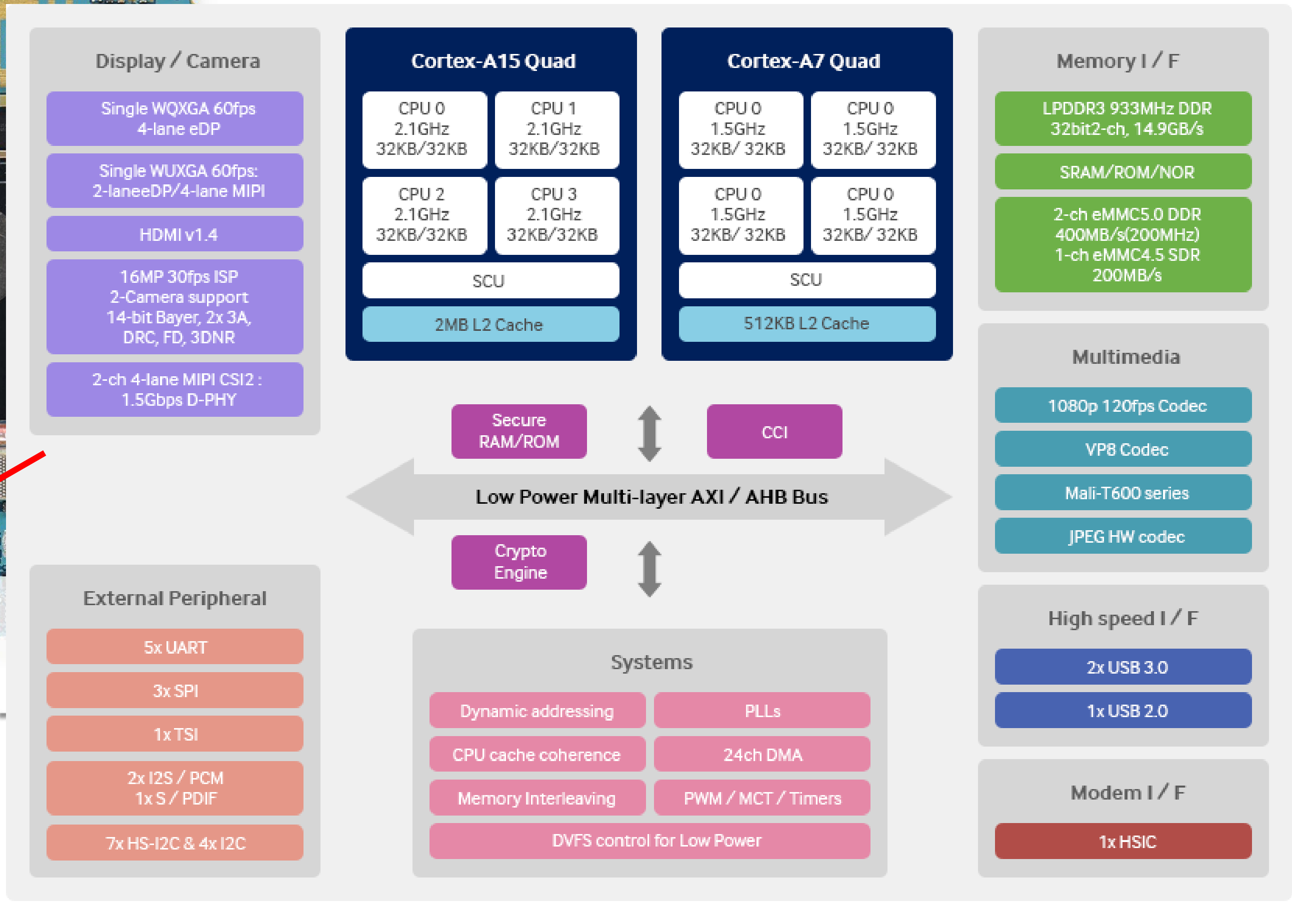
System-on-Chip

Samsung Galaxy S6

- Exynos 7420 System on a Chip (SoC)
- 8 ARM Cortex processing cores (4 x A57, 4 x A53)
- 30 nanometer: transistor gate width



Exynos 5422

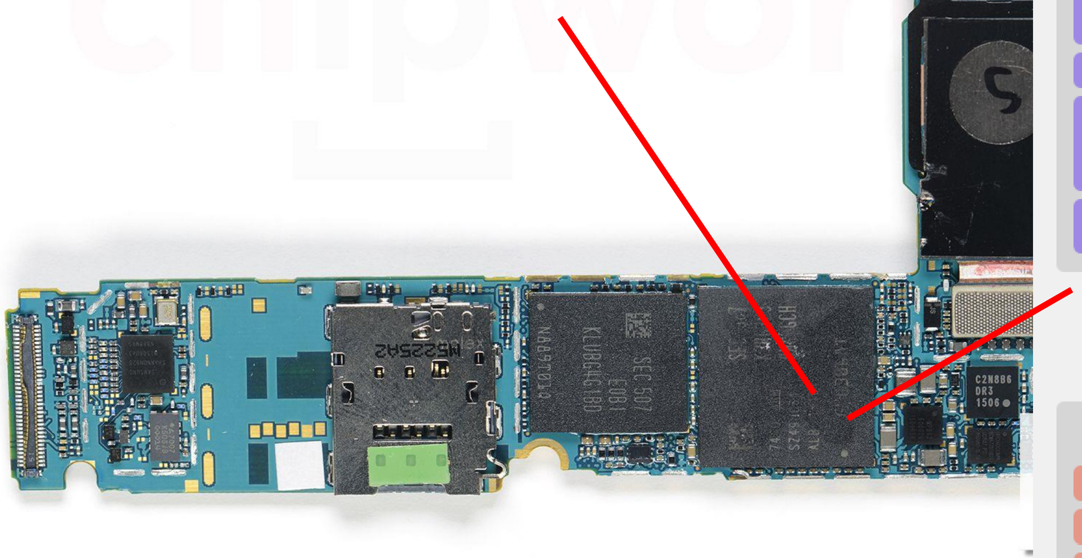


How to manage extreme workload variability?

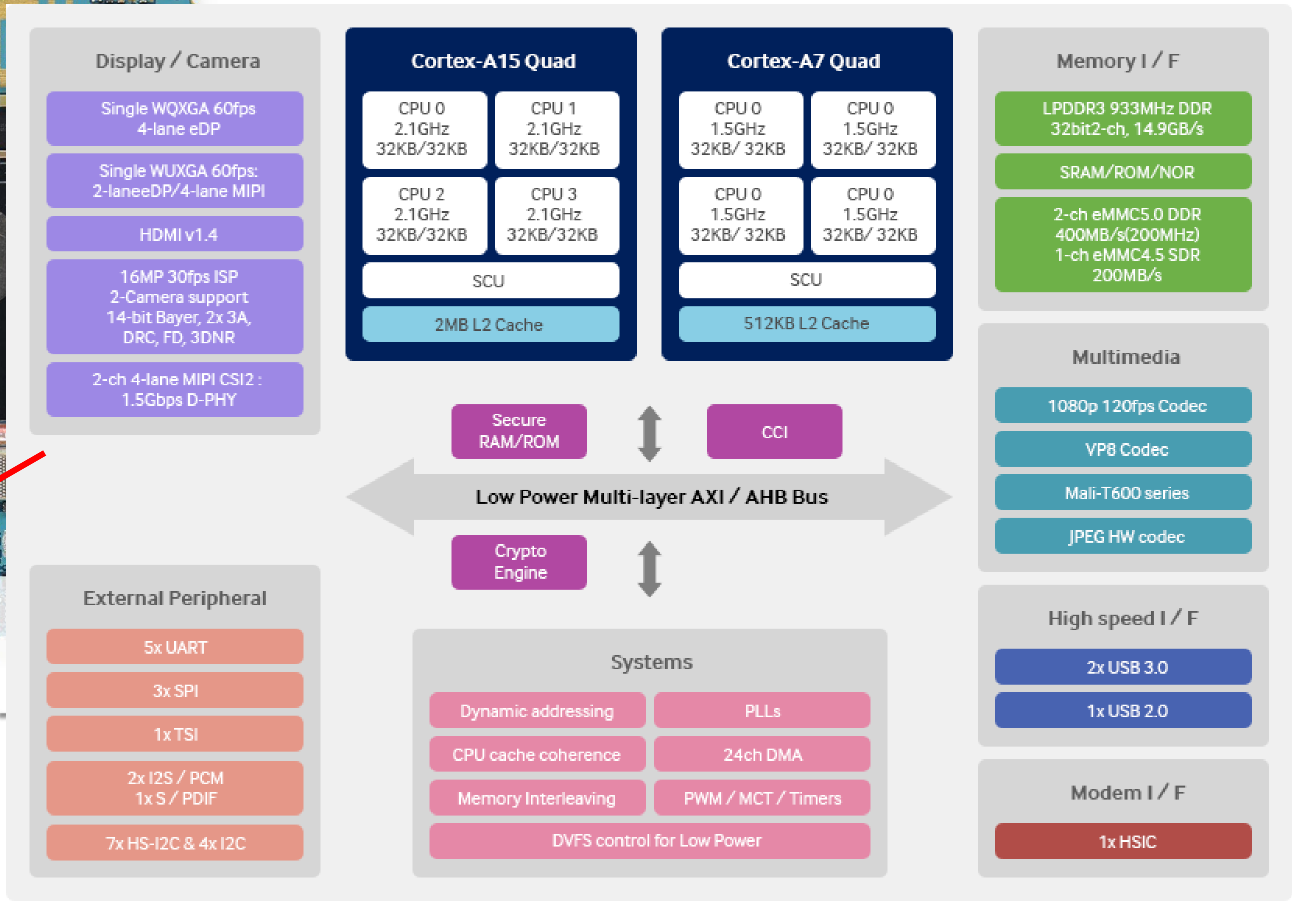
System-on-Chip

Samsung Galaxy S6

- Exynos 7420 System on a Chip (SoC)
- 8 ARM Cortex processing cores
  (4 x A57, 4 x A53)
- 30 nanometer: transistor gate width



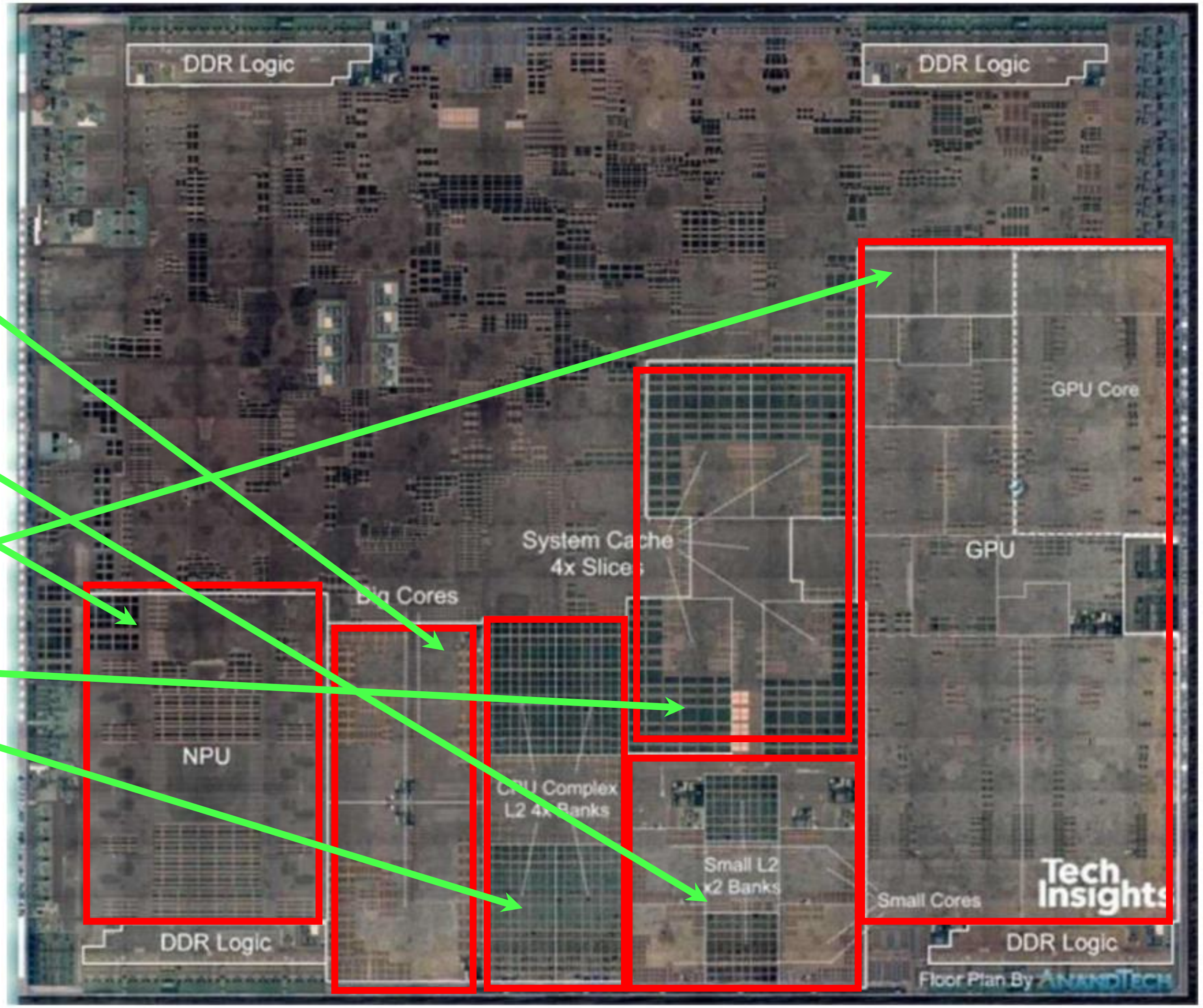
Exynos 5422



From Computer Engineering

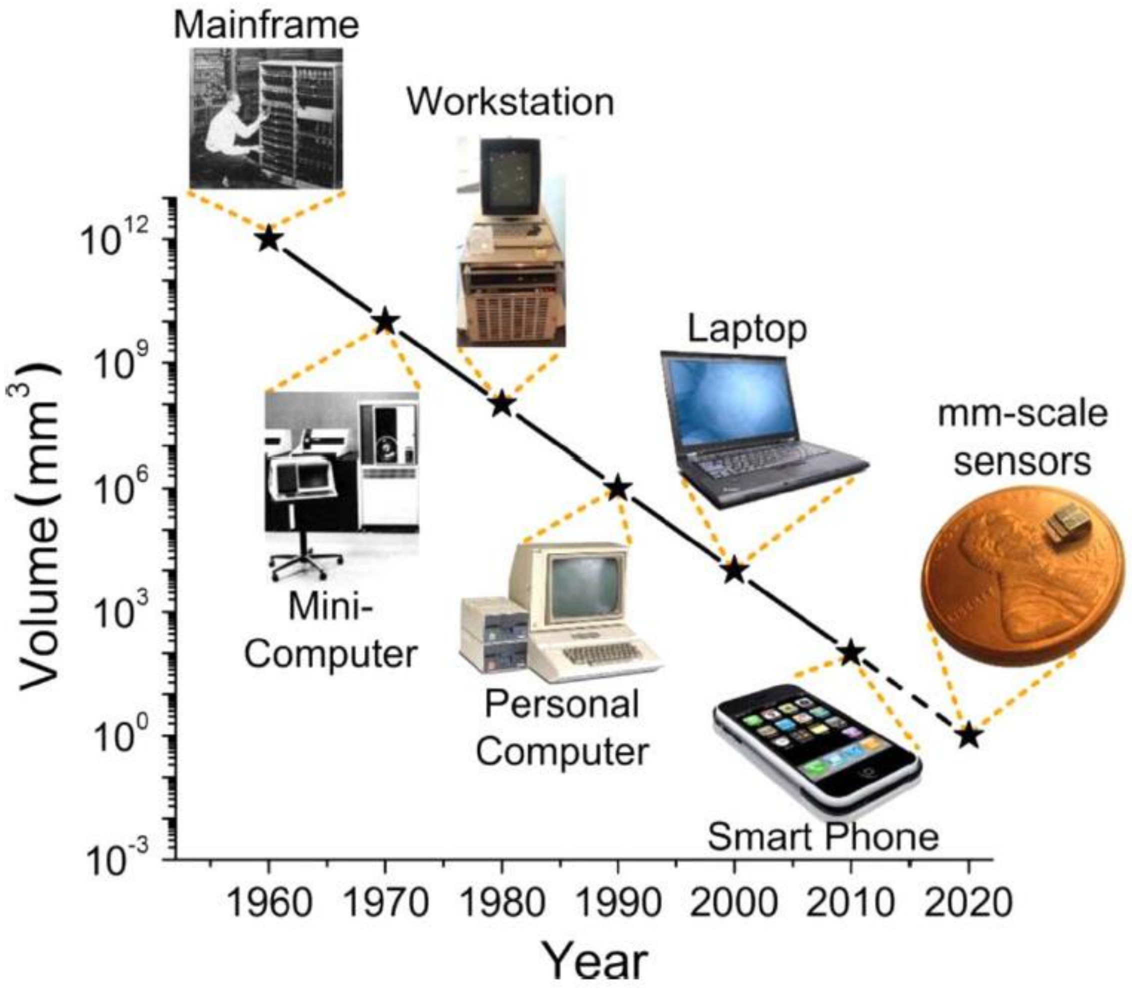
iPhone Prozessor A12

- 2 processor cores
  - high performance
- 4 processor cores - less performant
- Acceleration for Neural Networks
- Graphics processor
- Caches



Components and Requirements by Example
- Systems -

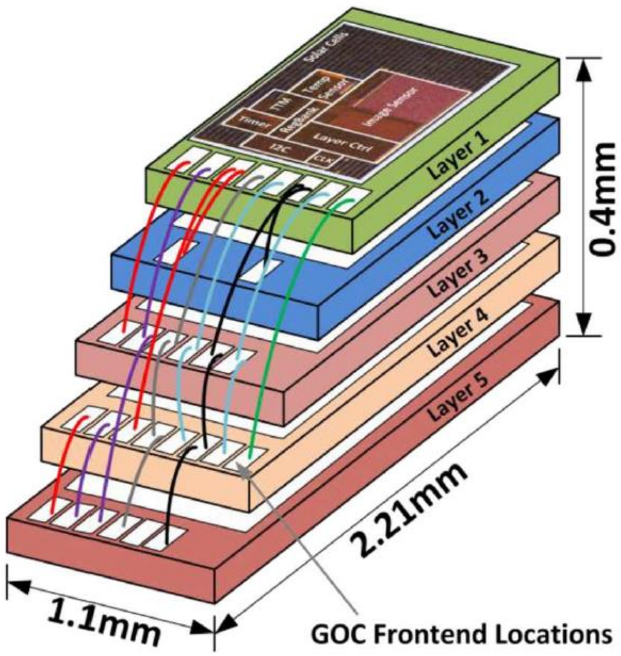
Zero Power Systems and Sensors

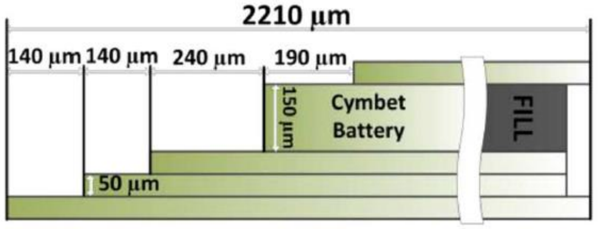


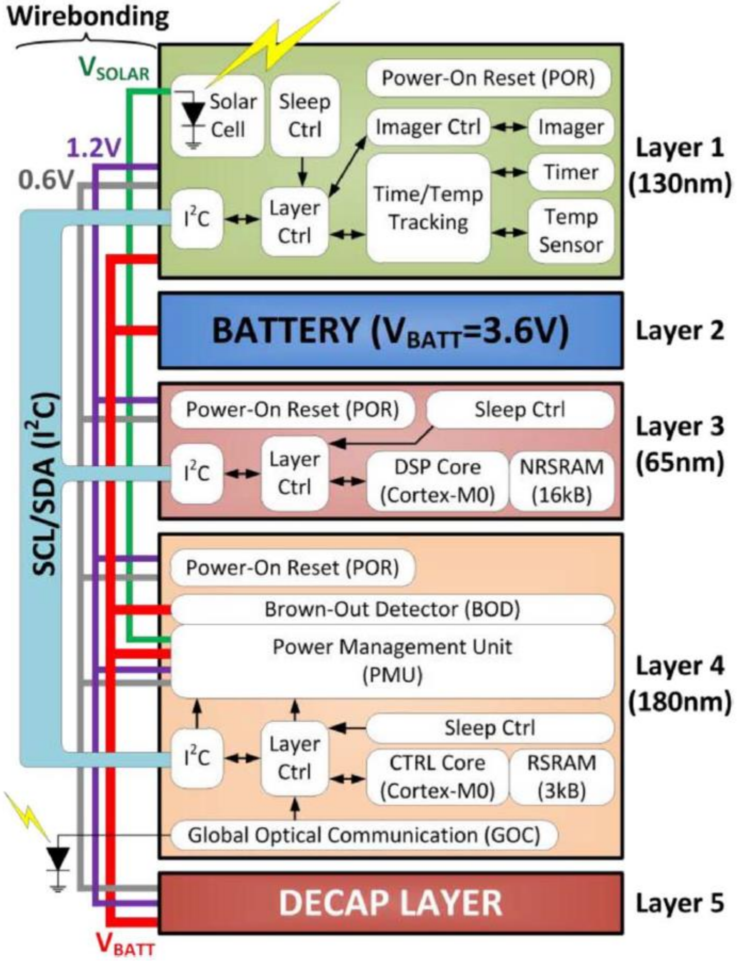
Streaming information to and from the physical world:

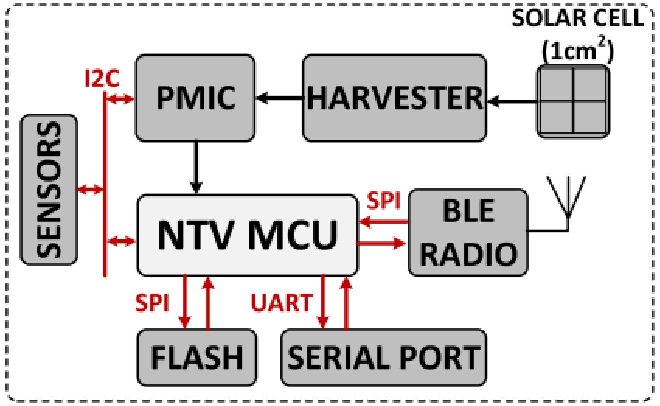
- "Smart Dust"
- Sensor Networks
- Cyber-Physical Systems
- Internet-of-Things (IoT)

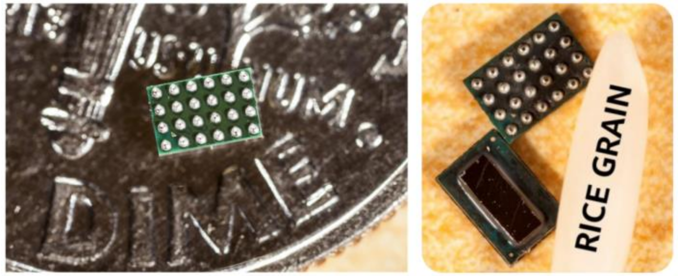
Zero Power Systems and Sensors











IEEE Journal of Solid-State Circuits, Jan 2013, 229-243.

IEEE Journal of Solid-State Circuits, April 2017, 961-971.

Trends …

- *Embedded systems are communicating with each other,* with servers or with the cloud. Communication is increasingly wireless.

- *Higher degree of integration* on a single chip or integrated components:
  - Memory + processor + I/O-units + (wireless) communication.
  - Use of networks-on-chip for communication between units.
  - Use of homogeneous or heterogeneous multiprocessor systems on a chip (MPSoC).
  - Use of integrated microsystems that contain energy harvesting, energy storage, sensing, processing and communication ("zero power systems").
  - The complexity and amount of software is increasing.

- *Low power and energy constraints* (portable or unattended devices) are increasingly important, as well as temperature constraints (overheating).
- There is increasing interest in *energy harvesting* to achieve long term autonomous operation.

Learning objective of Today Class

- Where we are what we will do.
- What are Embedded Systems and CPS
- Both hardware and software are important and you will learn
- Processors and evolution of them
- Trends and topics

How we do learning at PBL?

Research focus: Embedded Systems and Wireless Smart Sensors for a better and greener world

- An embedded system is some combination of computer hardware and software, either fixed in capability or programmable, that is **designed for a specific function** or for specific functions within a larger system.

Main Teaching at ETH

- Embedded systems – 300 students per year
- Machine Learning on Microcontrollers – 100 students per year
- Over 100 students projects per year.

Next Generation of IoT devices: Always-on Smart Sensors PBL research activities in all the 4 areas.

1.) Edge Signal Processing and AI

Smart devices

for perpetual operation

3.) Low power system design

2.) Energy harvesting

4.) Low Power and long-range communication

You are kindly invited for the PBL OPEN day… right now ☺

See you later or next week!