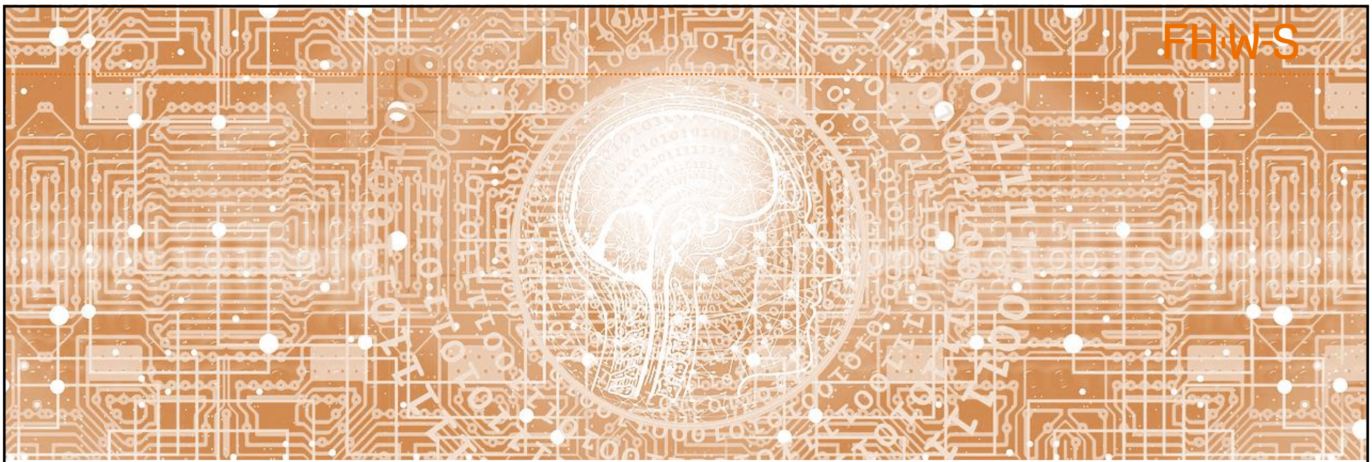


Embedded Systems and Field Buses

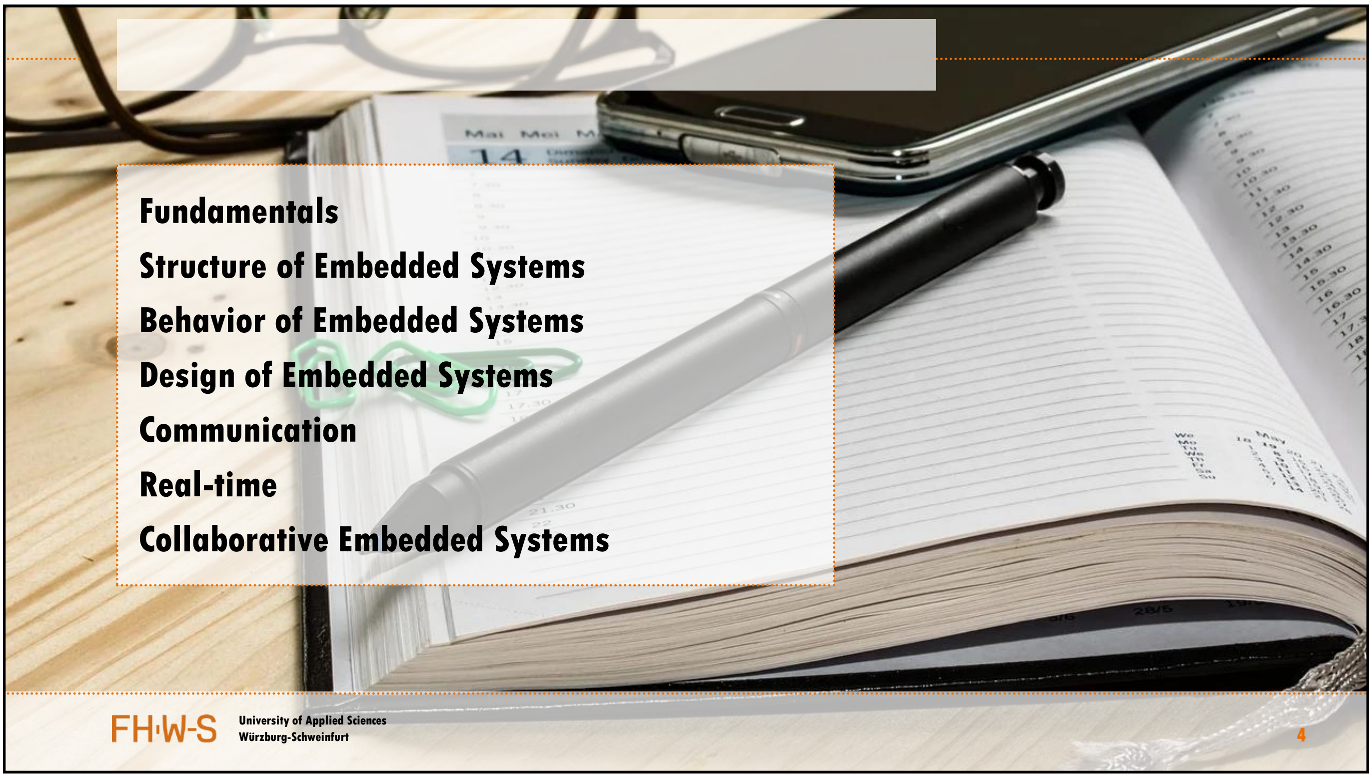
Prof. Dr. Marian Daun

FH·W-S University of Applied Sciences Würzburg-Schweinfurt

Group Discussion

Expectations

What this course is about and what it is not



- Fundamentals
- Structure of Embedded Systems
- Behavior of Embedded Systems
- Design of Embedded Systems
- Communication
- Real-time
- Collaborative Embedded Systems

FH·W-S

Embedded Systems Fundamentals

Terminology

Group Discussion

What is an Embedded System

Exercise

Define and differentiate from Embedded Systems:

- Information Systems
- Cyber-physical Systems
- IOT-Systems
- Software-intensive Systems
- Safety-critical Systems
- Distributed Systems
- Collaborative Systems

Characteristics

Characteristics of Embedded Systems include:

- **Real-time ability**
- **Safety**

- **Dependability**
- **Reactivity**
- **Efficiency**

- **New: Security**

Safety vs Security

In English often used synonymous but when used in German strictly distinguished!
Also officially distinguished according to ISO/IEC/IEEE 24765:

Safety
- expectation that a system does not, under defined conditions, lead to a state in which human life, health, property, or the environment is endangered

Safety vs Security

Security (according to ISO/IEC/IEEE 24765)

- protection against intentional subversion or forced failure
- defining, achieving, and maintaining confidentiality, integrity, availability, non-repudiation, accountability, authenticity, and reliability of a system
- degree to which a product or system protects information and data so that persons or other products or systems have the degree of data access appropriate to their types and levels of authorization
- protection of computer hardware or software from accidental or malicious access, use, modification, destruction, or disclosure
- protection of information and data so that unauthorized persons or systems cannot read or modify them and authorized persons or systems are not denied access to them
- protection against intentional subversion or forced failure, containing a composite of four attributes: confidentiality, integrity, availability and accountability, plus aspects of a fifth, usability, all of which have the related issue of their assurance

Group Discussion

How do Safety and Security influence each other?

Dependability

Typically consists (at least) of:

- **Availability**
  - degree to which a system or component is operational and accessible when required for use
- **Reliability**
  - ability of a system or component to perform its required functions under stated conditions for a specified period of time
- **Maintainability**
  - ease with which a software system or component can be modified to change or add capabilities, correct faults or defects, improve performance or other attributes, or adapt to a changed environment
  - ease with which a hardware system or component can be retained in, or restored to, a state in which it can perform its required functions
  - …

[ISO/IEC/IEEE 24765]

Real-Time

1. problem, system, or application that is concurrent and has timing constraints whereby incoming events must be processed within a given timeframe
2. pertaining to a system or mode of operation in which computation is performed during the actual time that an external process occurs, in order that the computation results can be used to control, monitor, or respond in a timely manner to the external process

[ISO/IEC/IEEE 24765]

Reactivity and Efficiency

Reactivity

- Stimulus and Response
- Input-driven systems

Efficiency

- degree to which a system or component performs its designated functions with minimum consumption of resources [ISO/IEC/IEEE 24765]
- Hardware is expensive
- Control units are expensive
- Communication is expensive

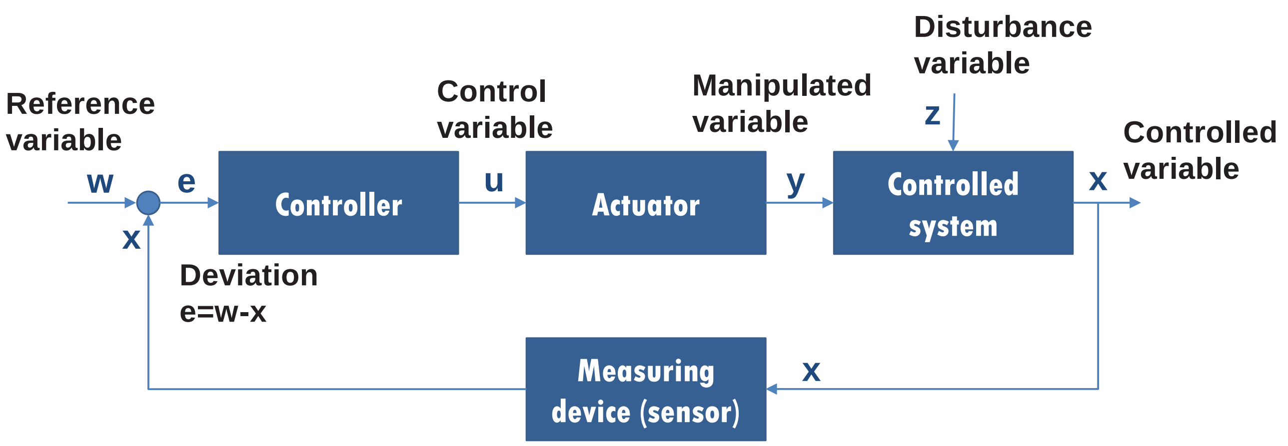
Exercise

Where can these characteristics be found in:

- Aircrafts?
- Cars?
- Smart Factories?
- Autonomous Driving?

Function of Embedded Systems

Control Loop (Steuer- und Regelkreis)

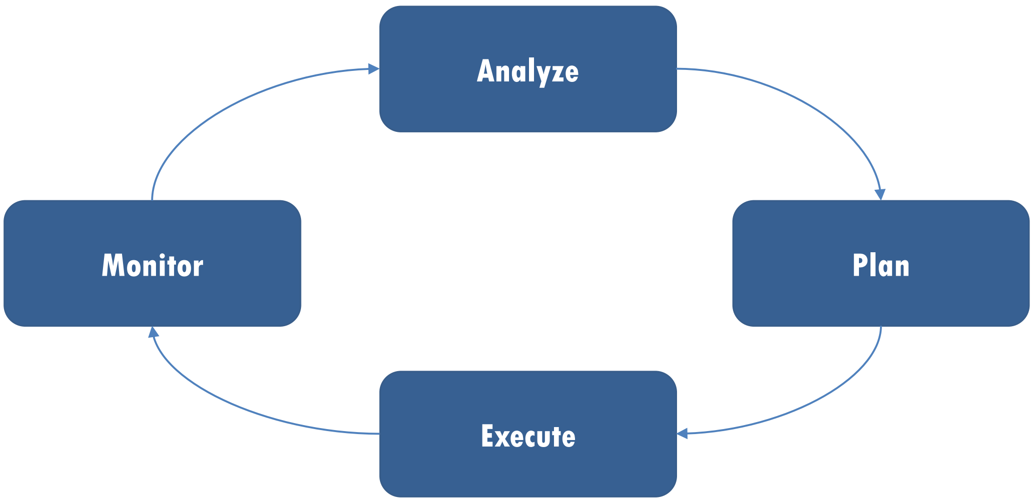


Exercise

For each system, identify variables to be controlled:

- Aircraft
- Car
- Smart Factory
- Autonomous Driving

MAPE-Loop



Monitor

What values do we need to monitor?
What inputs can we actually monitor?
How can we calculate/estimate the desired values from the monitored inputs?
How often to we need to monitor a value?

Examples:

Monitor current vehicle's speed via sensor at brakes

Monitor engine torque to estimate acceleration

Be Careful!
Values are influenced by multiple MAPE loops!

For instance, engine torque is not only increasing when accelerating but, e.g., also when climate control cools down the car.

FH·W-S University of Applied Sciences Würzburg-Schweinfurt

Analyze

Be Careful!
When is a deviation a deviation!

Desired speed and current speed will almost never be equal. Define acceptable ranges.

Is there a deviation?

Examples:

Vehicle's speed is lower than the desired speed

Engine torque is higher than is safe for the engineh

FH·W-S University of Applied Sciences Würzburg-Schweinfurt

Plan

Analyze

How do we get to where we want?

Examples:

Vehicle’s speed is lower than the desired speed
=> Accelerate (But how much?)

Engine torque is higher than is safe for the engine
=> Go to Fail/Safe-Mode

Plan

Execute

Execute

Execute the plan!
Inform other Embedded Systems and mostly ask other embedded systems to do what you want them to do

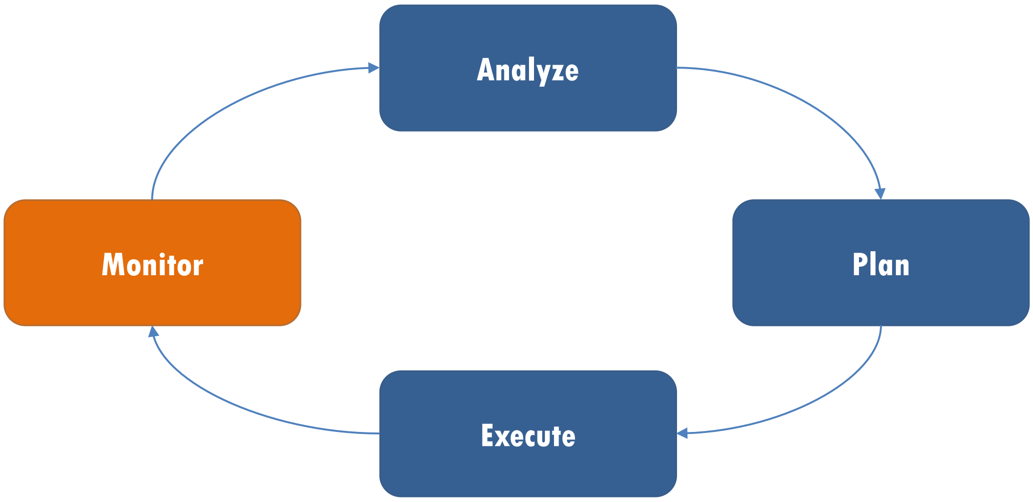
Example:

Vehicle’s speed is lower than the desired speed
=> Ask the engine control unit to accelerate

Values should not be changed by multiple embedded systems at once

One system needs to be responsible

And Again: Monitor



Exercise

For each system, define one relevant MAPE-Loop:

- Aircraft
- Car
- Smart Factory
- Autonomous Driving

HW/SW Co-design

Engineers like top-down approaches.

However, this does not work with software.

The development of Software and Hardware components of Embedded Systems needs to be intertwined.

There is a multitude of solutions. We can have hardware solutions, software solutions, or mixtures. Thus, the hardware defines what software we need and the software defines what hardware we need.

Continuous vs Discrete Values

Hardware is continuous

Software is discrete

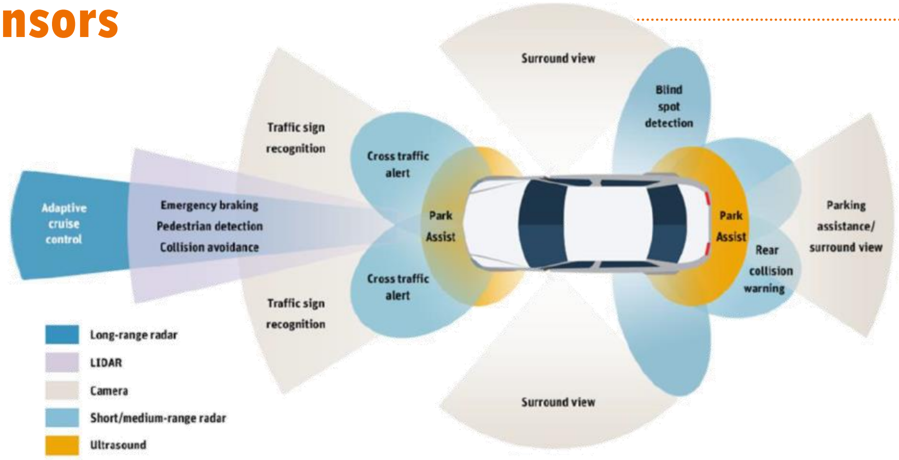
What does this mean?

Sensors

Group Discussion

What sensors do we know?

Exemplary Sensors



https://www.ansys.com/about-ansys/advantage-magazine/volume-xii-issue-1-2018/autonomous-vehicle-radar

Slide taken from Lecture Materials provided by Prof. Dr. Marco Schmidt

Actuators

Group Discussion

What actuators do we know?

Exercise

For the MAPE-Loops identified before, define the necessary Sensors and Actuators

Literature

[ISO/IEC/IEEE 24765] Systems and software engineering – Vocabulary. International Standard, ISO/IEC/IEEE, 2017.

Questions for Self-Assessment

What is an embedded system?
How do embedded systems differ from other systems?
What are the main characteristics of embedded systems?
Where is the difference between safety and security?
What is the difference between a control loop and a MAPE loop?
What problems can arise when defining a MAPE loop?
Why is the development of hardware and software intertwined?