Safety and Reliability of Embedded Systems - SRES (WS 19/20)
Problem Set 1

Problem 1: Software Intensive Systems

a) *Please define the general term "System" according to Birolini and explicitly name the parts a system can encompass. Explain your answer in the view of a technical field.*

-

System:

- Technical and organizational means for the autonomous fulfillment of a task.
- Generally, a system can consist of hardware, software, people and logistic assistance.
- Example: Aviation
  Hardware= Aircraft, tower, runway
  Software = Flight control software, software for in-flight entertainment system
  People= Pilot, co-pilot, stewardess, ground personnel
  Logistic assistance= Fueling, Baggage handling, ticketing.

b) *What is the difference to a "Technical System"?*

-

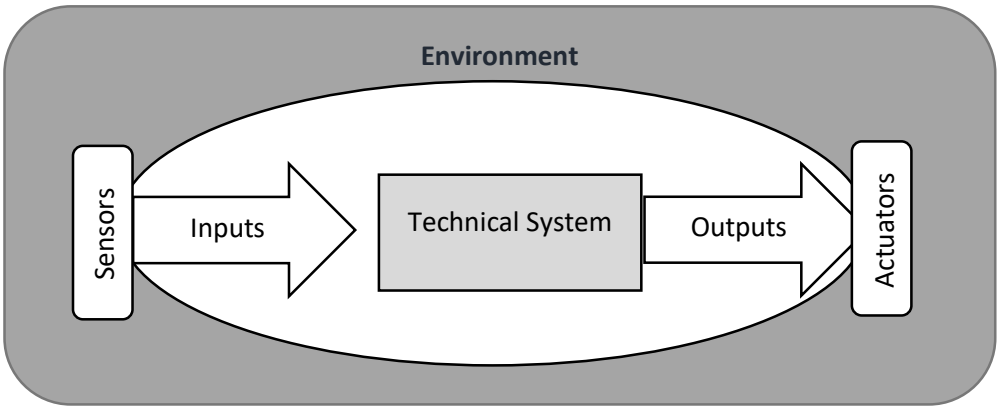
Technical system:

- system where influences by people and logistics are ignored.
- Example: Avionics
  Hardware = onboard flight control unit.
  Software = Flight control software.

c) *For the analysis of a technical (embedded) system it is crucial to extract it from its environment. How can this be achieved? Please sketch your ideas.*

-

The coupling between the technical system and its environment is modeled by the "inputs" and "outputs" coming from "sensors" and going to "actuators"



d) Please list important non-functional requirements for embedded systems. What category functional / non-functional) does Safety belong to? Why?

-.

- Real time behavior: Deadlines must be met as required by the environment.
- Minimal resource consumption: Memory, energy, dimensions, weight
- Cost
- Dependability
  - Availability – Readiness for correct service
  - Reliability – Continuity of correct service
  - Safety – Absence of catastrophic consequences on users and environment
  - Confidentiality – Absence of unauthorized disclosure of information
  - Integrity – Absence of improper system state alterations
  - Maintainability – Ability to undergo repairs and modifications
- Robustness: to deliver an acceptable behavior, also in exceptional situations.

Problem 2: Reliability vs. Availability

Please explain the difference between "Reliability" and "Availability".

-

Reliability: Measure for the ability of an item to remain functional. Behavior of an entity during or after time period, it depends on situations.

expressed by the probability that the required function is executed failure-free under given working conditions. during a given time period: MTBF

Availability: Measure for the ability of an item to be functional at a given time, expressed by the ration:

$$\frac{MTBF}{(MTBF+MTTR)}$$

; MTBF = Mean time between failures

; MTBR = Mean time between repair

Problem 3: Safety vs. Security

Please explain the terms "Safety" and "Security".

What is meant by "Technical Safety" in comparison to "Safety"?

-

Safety: state where the danger of personal or property damage is reduced to an acceptable value

Security: Security is the concurrent users only

a. availability for the authorized users only
b. confidentiality
c. integrity

Technical safety: Measure for the ability of a failure afflicted item to endanger neither persons, property nor the environment.

Problem 4: Failure, Fault, Error

What is meant by the terms "Failure", "Fault", and "Error"? Please illustrate your answer by means of the "Ariane 5" disaster (see lecture).

Does an error always result into a failure?

-

Failure: Inconsistent behavior w.r.t. specified behavior while running a system (happens dynamically during the execution) => Each failure has a time-stamp

Fault, defect: Statically existent cause of a failure, i.e. a bug (usually the consequence of an error made by the programmer)

Error: Basic cause for the fault (e.g., misunderstanding of a particular statement of the programming language)

Example: Maiden flight of Ariane 5

- Failure: Break-down of flight controller resulting in mechanical destruction of rocket.
- Fault: Conversion of a 64-bit floating point variable into a signed integer with range -32768…32767, leading to a data overflow within that variable.
  horizontal_veloc_bias := integer(horizontal_veloc_sensor);
- Error: Blind reuse of software components of Ariane 4, Which have not been tested sufficiently within the new environment.

Problem 5: Hardware Failures vs. Software Failures

Please explain the differences between hardware failures and software failures.

-

Hardware failures

- Commonly caused by manufacturing errors or wear (physical degradation)
- Traditionally, potential design faults within hardware are disregarded… but the assumption that today's complex hardware is free from design faults cannot be hold anymore! (use of SBLs VHDL, Verilog).
- In the ideal case, the system contains no hardware defects in the beginning and therefore shoes no hardware failures.
- The reliability of the system does not exceed its initial value through the substitution of components with new components.

Software failures

- It is commonly assumed that the software contains defects, which cannot be detected immediately
- Software failures are a result of design errors that are contained in the product from the start and appear accidentally
- After error correction the system reliability exceeds its initial value (under the assumption that no additional faults are introduced)
- Faults that are introduced during debugging decrease reliability

Problem 6: Correctness and Robustness

Please give your opinion on the following statements:

	True	False
Correctness has a binary character	☒	☐
An artifact is not consistent to its specification, if it is not correct	☒	☐
Robustness has a binary character	☐	☒
Robustness is a property only of the implementation	☐	☒
A safe system can suffer from security breach	☒	☐
Environment can influence system's safety	☒	☐