Embedded Systems

Embedded systems refer all kinds of systems performing either signal processing or control tasks and that do not look like our PC or workstation, but are instead designed for a specific application or area of use and optimized in terms of size, cost, performance, speed, and interfaces.
In contrast to a saturated market of multi-purpose computers, the area of embedded systems will be a market billion-dollar worth and countless application possibilities in the coming years. The opportunities in the labor market are correspondingly very positive.
Examples of embedded systems are:

- Wireless and battery-powered systems: e.g., mobile phones, tablet PCs, Bluetooth communication devices, wearable computers (computers as part of clothing)
- Building automation: e.g., field bus systems, automotive systems, ambient intelligence, and industrial controls
- Sports and entertainment: e.g., electronic pulse measurement, navigation and monitoring, electronic games
- Dedicated computers and processors: e.g., digital signal processors, microcontrollers and reconfigurable computing systems, e.g., FPGAs

The focus of this major field of study is the design and design methodologies for embedded systems. In particular, the following topics are taught:

- How do I design an embedded system? Languages and models for the description, analysis, and simulation of functional and temporal behavior play an important role here.
- Which design problems need to be solved in product development? This includes the selection of suitable hardware and software modules, mapping functionality to these components and scheduling algorithms.
- Which constraints have to be met or considered in the design? Size, cost, weight, energy consumption, design time and performance are the most important quality criteria for the design of embedded systems. In this realm, it is essential how these qualities metrics can be determined, either analytically, by simulation, by synthesis or by other appropriate estimation techniques.
- How can I optimize my system concerning multiple objectives? A central question of the field of hardware/software co-design is whether a function should be better implemented in hardware or software for cost and efficiency reasons (so-called hardware/software partitioning).
- Finally, how do I show that my designed system works correctly? In this context, we will learn about methods for validation, e.g., test and simulation as well as formal verification.