

Software Development for Embedded and Realtime Systems

Introduction to Embedded and Realtime Systems

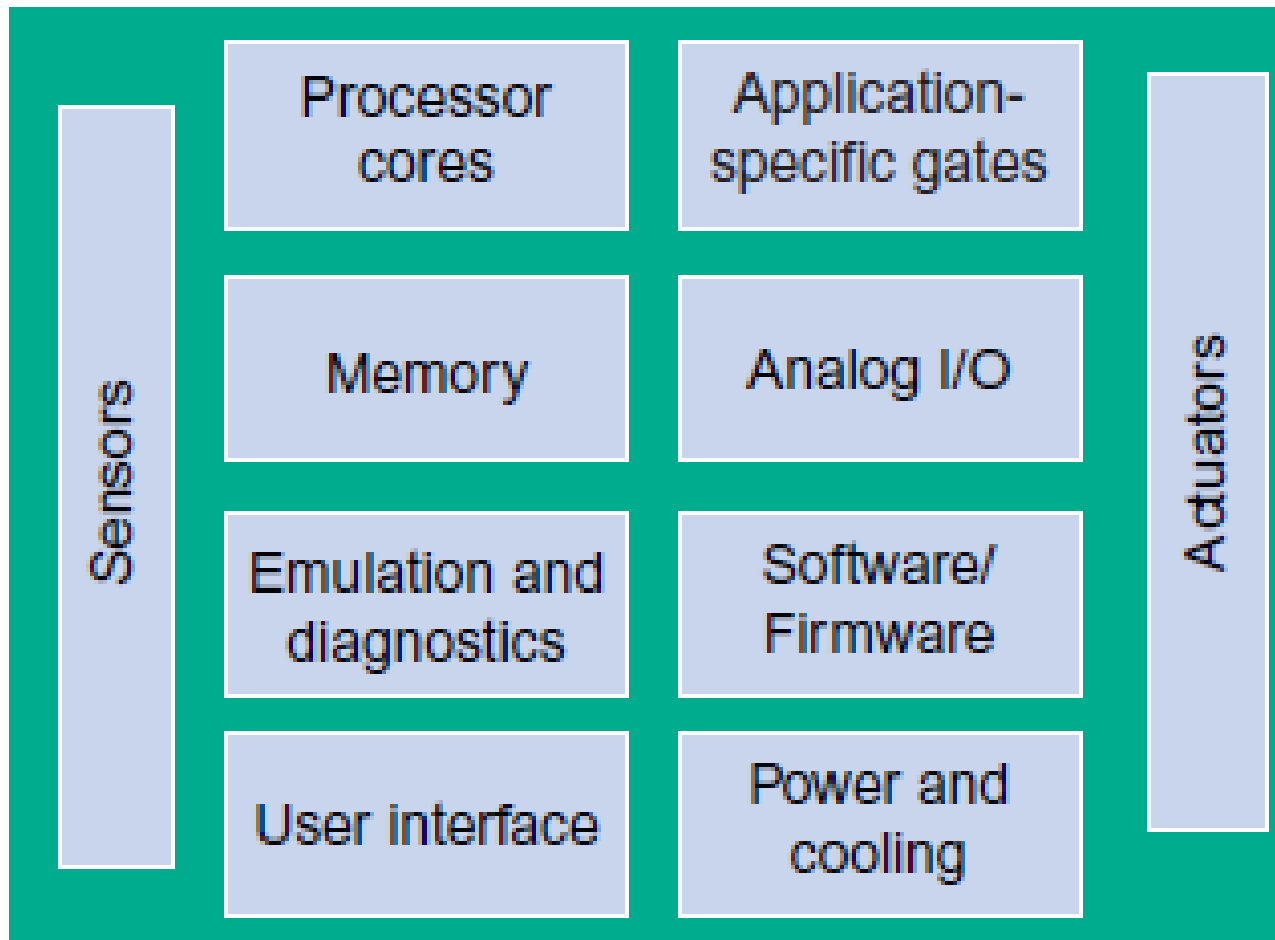
Course Objectives

- Key software development phases for embedded systems
 - Problem definition
 - Develop an architecture
 - Develop a design
 - Implementation (Component-based)
 - Verification and Validation

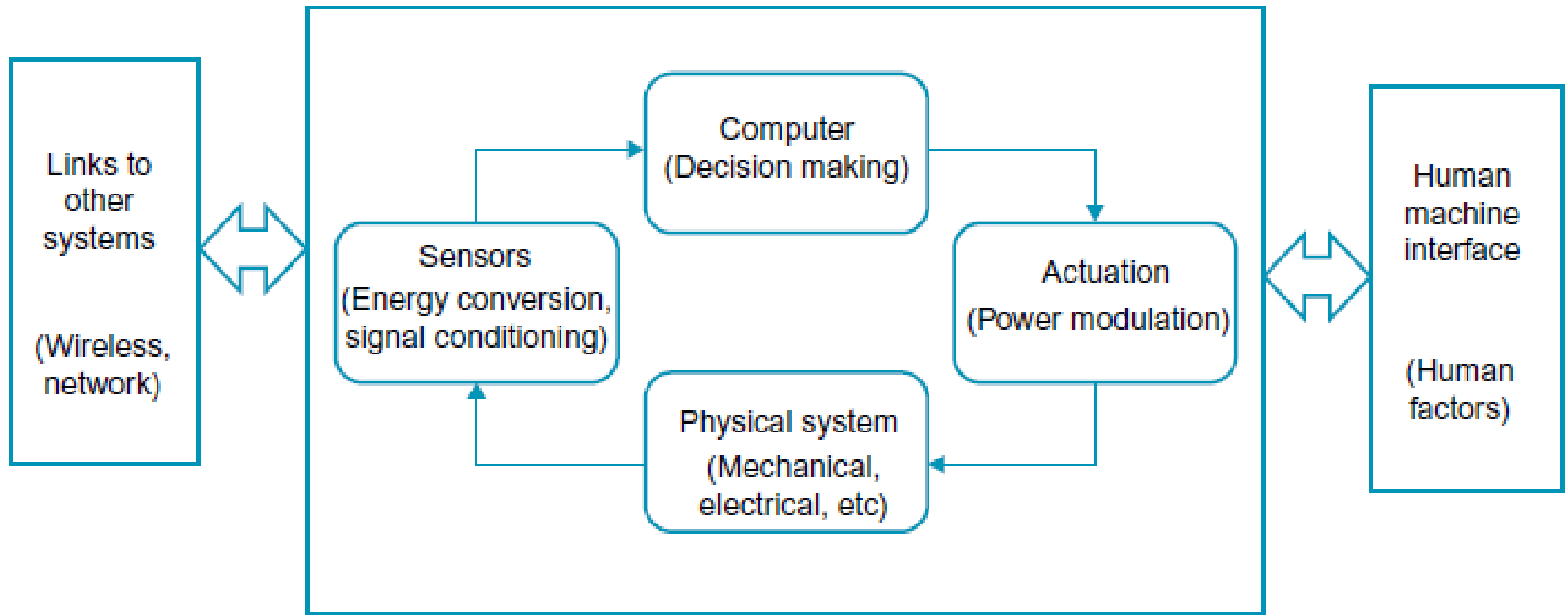
What is an Embedded System?

- A combination of hardware and software components to form a computational engine that will perform a **specific function**.
- Perform in reactive and time-constrained environments
- Responds to the environment via **sensors** and controls the environment using **actuators**
- Realtime – **hard real time** vs **soft real time**
- Multi-rate - can handle multiple processing rates

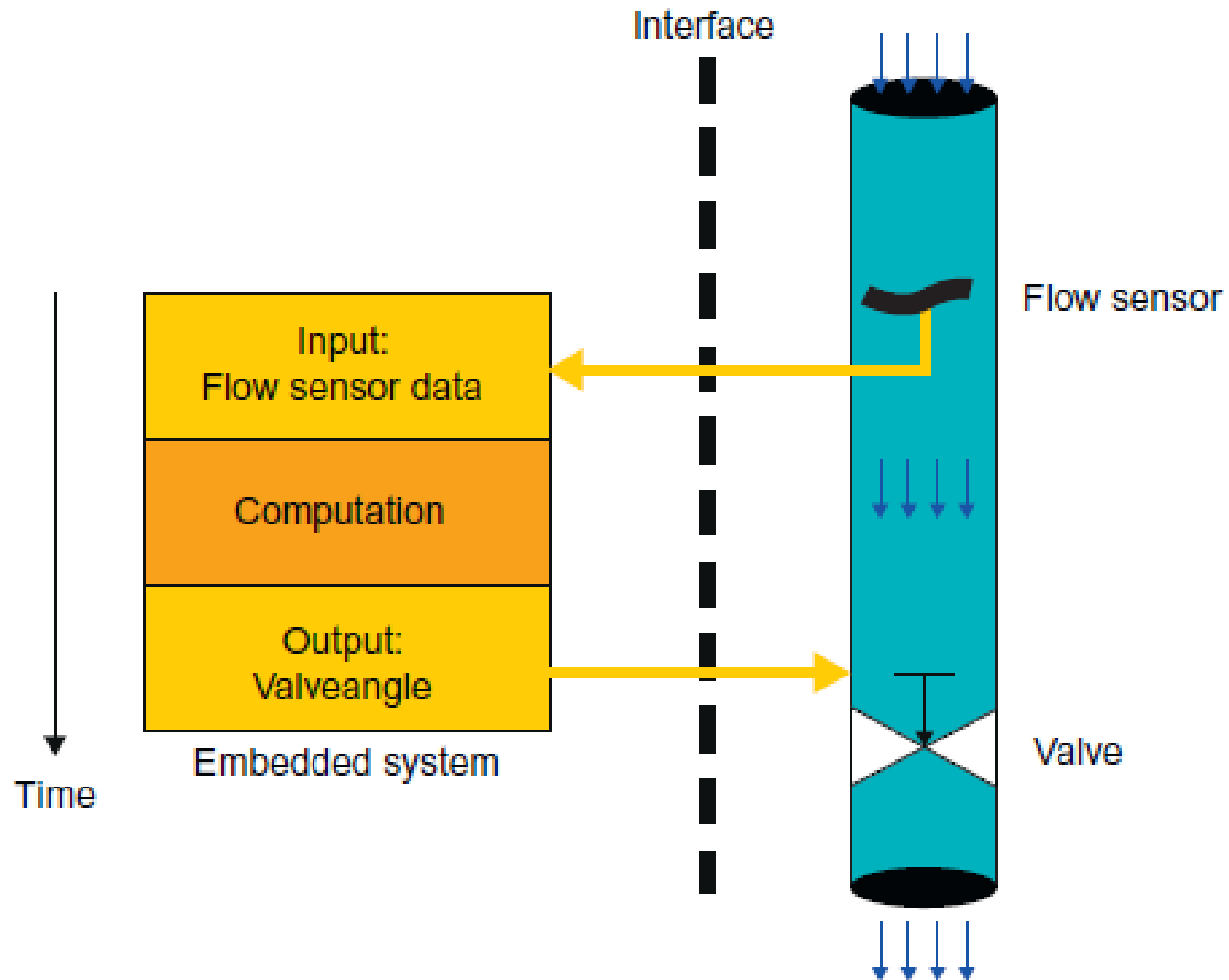
Typical Embedded System



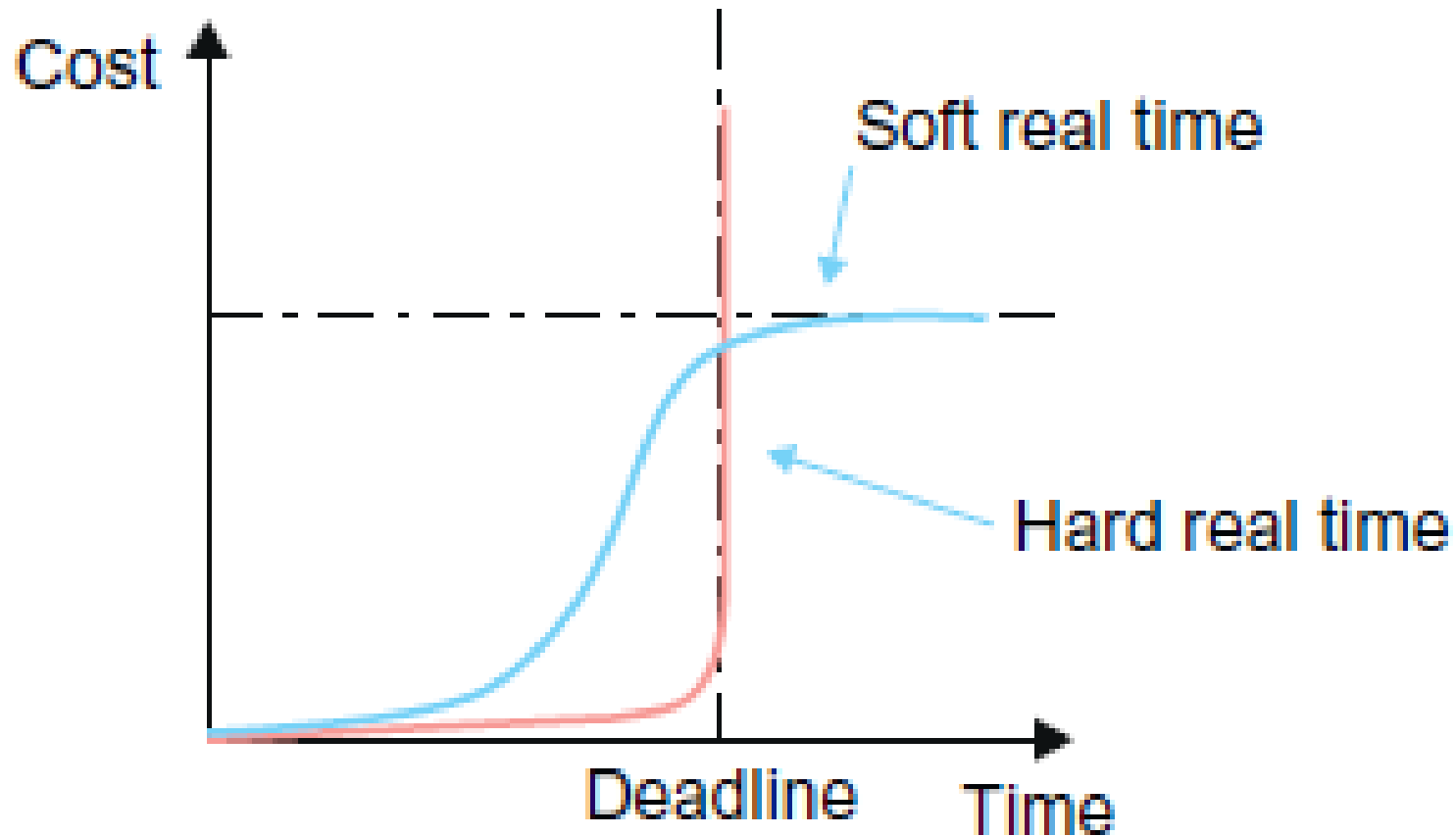
Abstract Model of an Embedded System



Example Embedded System



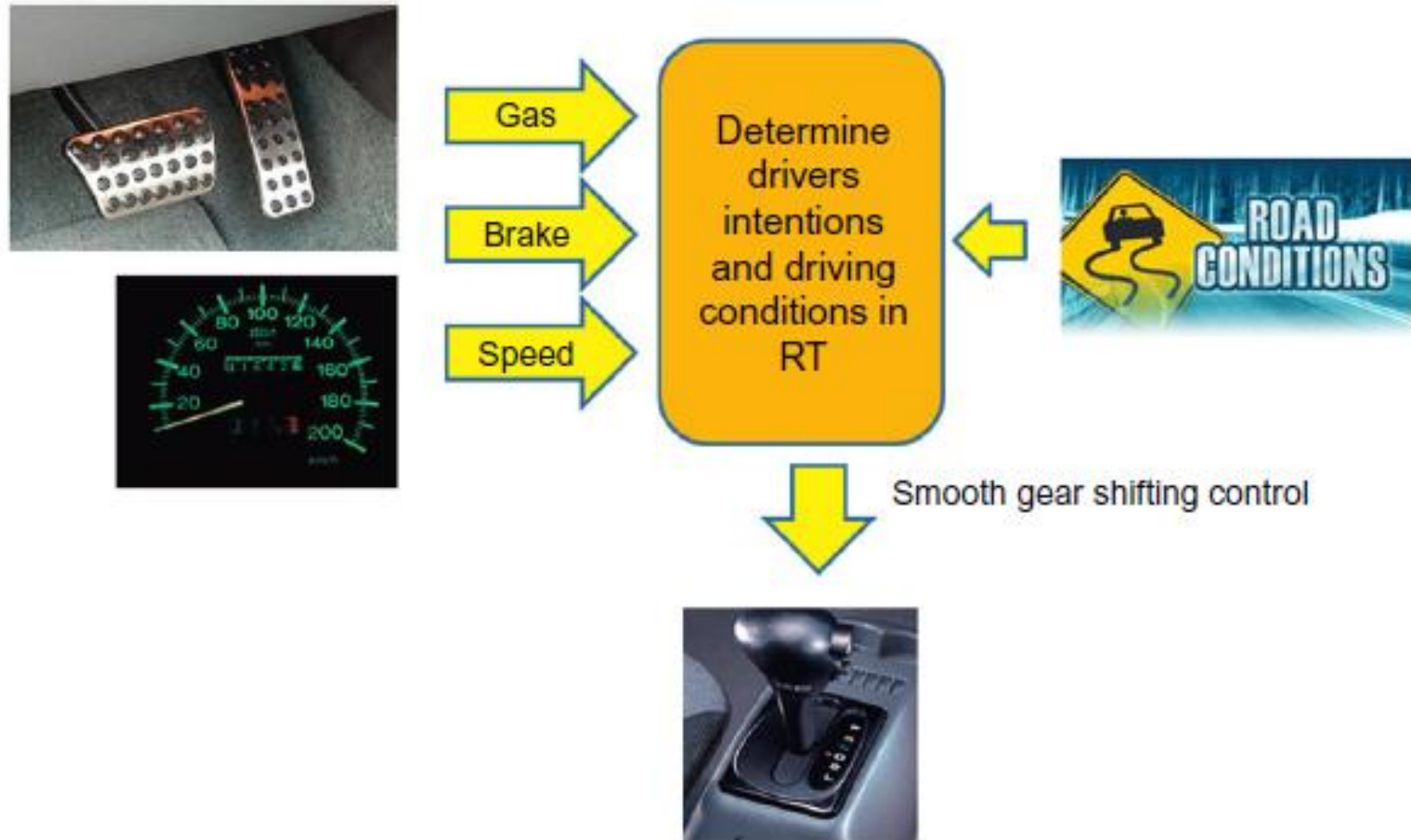
Hard vs Soft Real time



Examples of hard and soft real time systems

System type	Hard or soft real time?
Traffic light control	Hard RT – Critical
Automated teller machine	Soft RT – Non-Critical
Controller for radiation therapy machine	Hard RT – Critical
Car simulator for driver training	Hard RT – Non Critical
Highway car counter	Soft RT – Non-Critical
Missile control	Hard RT – Critical
Video games	Hard RT – Non Critical
Network chat	Soft RT – Non-Critical

Hard real time safety critical system



Real-time vs Time-shared

Characteristic	Time-Shared Systems	Real-Time Systems
System capacity	High throughput	Schedulability and the ability of system tasks to meet all deadlines
Responsiveness	Fast average response time	Ensured worst case latency which is the worst-case response time to events
Overload	Fairness to all	Stability; when the system is overloaded important tasks must meet deadlines while others may be starved

Real time system constraints

- A computing system being hard real-time says nothing about the magnitudes of the deadlines.
 - may be microseconds or weeks.
- A real-time system task and execution environment must be adjusted to enable a **schedule** and resource allocation which **meets all deadlines**.

Real time event categories

- **Asynchronous events**
 - entirely unpredictable, e.g. cell phone call arriving at a cellular base station
- **Synchronous events**
 - predictable events and occur with precise regularity, e.g. video streaming
- **Isochronous events**
 - occur with regularity within a given time window,
 - e.g. audio data in a networked multimedia application must appear within a window of time when the corresponding video stream arrives

References

- **Chapter 1:** Oshana, Robert, ed. **Software Engineering for Embedded Systems: Methods, Practical Techniques, and Applications.** Newnes, 2013.

Thank you for
your attention.