CE-ESY. Embedded Systems
CE-ESY0. Introduction to embedded systems [core]

CE-ESY1. Embedded microcontrollers [core]

CE-ESY2. Embedded programs [core]

CE-ESY3. Real-time operating systems [core]

CE-ESY4. Low-power computing [core]

CE-ESY5. Reliable system design [core]

CE-ESY6. Design methodologies [core]

CE-ESY7. Tool support [elective]

CE-ESY8. Embedded multiprocessors [elective]

CE-ESY9. Networked embedded systems [elective]

CE-ESY10. Interfacing and mixed-signal systems [elective]

CE-ESY11. Embedded computing system examples [elective]

CE-ESY0. History and overview of embedded systems [core]

*Suggested time: 1 hour*

**Topics:**
- Indicate some reasons for studying embedded systems.
- Highlight some people that influenced or contributed to the area of embedded systems.
- Indicate some important topic areas such as mapping between language and hardware, classifications, influence of software engineering, applications and techniques, and tool support.
- Contrast between an embedded system and other computer systems.
- Mention the role of programming and its associated languages as applied to embedded systems.
- Explore some additional resources associated with embedded systems.
- Explain the purpose and role of embedded systems in computer engineering.

**Learning objectives:**
- Identify some contributors to embedded systems and relate their achievements to the knowledge area.
- Describe the meaning of an embedded system.
- Explain the reasons for the importance of embedded systems.
- Describe the relationship between programming languages and embedded systems.
- Describe how computer engineering uses or benefits from embedded systems.

CE-ESY1. Embedded microcontrollers [core]

*Suggested time: 6 hours*

**Topics:**
- Structure of a basic computer system: CPU, memory, I/O devices on a bus.
- CPU families used in microcontrollers: 4-bit, 8-bit, 16-32-bit.
- Basic I/O devices: timers/counters, GPIO, A/D, D/A.
- Polled I/O vs. interrupt-driven I/O.
- Interrupt structures: vectored and prioritized interrupts.
- DMA transfers.
• Memory management units.
• Memory hierarchies and caches.

Learning objectives:
• Understand the CPU in the context of a complete system with I/O and memory.
• Understand how the CPU talks to the outside world through devices.
• Understand how memory system design (caches, memory management) affect program design and performance.

CE-ESY2. Embedded programs [core]
Suggested time: 3 hours
Topics:
• The program translation process: compilation, assembly, linking.
• Representations of programs: data flow and control flow.
• Fundamental concepts of assembly language and linking: labels, address management.
• Compilation tasks: mapping variables to memory, managing data structures, translating control structures, translating expressions, etc.
• What can and cannot be controlled through the compiler; when writing assembly language makes sense.

Learning objectives:
• Understand how high-level language programs are translated into executable code.
• Know what compilers are good at and what they are not good at.
• Comprehend basic representations of programs that can be used to manipulate programs either in a compiler or by hand.

CE-ESY3. Real-time operating systems [core]
Suggested time: 3 hours
Topics:
• Context switching mechanisms.
• Scheduling policies.
• Rate-monotonic scheduling: theory and practice.
• Priority inversion.
• Other scheduling policies: EDF, etc.
• Message-passing vs. shared memory communication.
• Interprocess communication styles: mailbox, RPC, etc.

Learning objectives:
• Distinguish RTOSs from workstation/server OS.
• Distinguish real-time scheduling from traditional OS scheduling.
• Understand major real-time scheduling policies.
• Understand interprocess communication mechanisms.

CE-ESY4. Low-power computing [core]
Suggested time: 3 hours
Topics:
Sources of energy consumption: toggling, leakage.
Instruction-level strategies for power management: function unit management.
Memory system power consumption: caches, off-chip memory.
Power consumption with multiple processes.
System-level power management: deterministic, probabilistic methods.

Learning objectives:
- Understand why low-power computing is important.
- Identify sources of energy consumption.
- Identify possible remedies for energy consumption at various levels of design abstraction.

CE-ESY5. Reliable system design [core]
Suggested time: 3 hours
Topics:
- Transient vs. permanent failures in hardware.
- Sources of errors from software.
- The role of design verification in reliable system design.
- Fault-tolerance techniques.
- Famous failures of embedded computers.

Learning objectives:
- Understand the variety of sources of faults in embedded computing systems
- Be able to identify strategies to find problems.
- Be able to identify strategies to minimize the effects of problems.

CE-ESY6. Design methodologies [core]
Suggested time: 3 hours
Topics:
- Multi-person design projects.
- Designing on-time and on-budget.
- Design reviews.
- Tracking error rates and sources.
- Change management.

Learning objectives:
- Understand why real-world projects are not the same as class projects.
- Identify important goals of the methodology.
- Understand the importance of design tracking and documentation.

CE-ESY7. Tool support [elective]
Suggested time: 3 hours
Topics:
- Compilers and programming environments.
- Logic analyzers.
- RTOS tools.
- Power analysis.
- Software management tools.
• Project management tools.

Learning objectives:
• Understand role of hardware and software tools in system development.
• Understand how tools are used to support the methodology.

CE-ESY8. Embedded multiprocessors [elective]

*Suggested time: 3 hours*

Topics:
• Why multiprocessors are important: performance, power, cost.
• Hardware/software partitioning for single-bus systems.
• More general architectures.
• Platform FPGAs as multiprocessors.

Learning objectives:
• Understand why multiple processors are used in embedded systems.
• Identify trade-offs between CPUs and hardwired logic in multiprocessors.
• Understand basic design techniques.

CE-ESY9. Networked embedded systems [elective]

*Suggested time: 3 hours*

Topics:
• Why networked embedded systems.
• Example networked embedded systems: automobiles, factory automation systems.
• The OSI reference model.
• Types of network fabrics.
• Network performance analysis.
• Basic principles of the Internet protocol.
• Internet-enabled embedded systems.

Learning objectives:
• Understand why networks are components of embedded systems.
• Identify roles of hardware and software in networked embedded systems.
• Compare networks designed for embedded computing with Internet networking.
CE-ESY10. Interfacing and mixed-signal systems [elective]

*Suggested time: 3 hours*

**Topics:**
- Digital-to-analog conversion.
- Analog-to-digital conversion.
- How to partition analog/digital processing in interfaces.
- Digital processing and real-time considerations.

**Learning objectives:**
- Understand pros and cons of digital and analog processing in interfaces.
- Understand fundamentals of A/D and D/A conversion.

CE-ESY11. Embedded computing system examples [elective]

*Suggested time: 3 hours*

**Topics:**
- Compact disk and DVD.
- Laser and ink jet printers.
- Automotive electronics.
- Cellular telephones: handsets and base stations.
- Video games.

**Learning objectives:**
- Apply embedded system design concepts to real-world systems.