UNIVERSITÀ
DEGLI STUDI DI MILANO-BICOCCA

COURSE SYLLABUS

Embedded Systems Architectures and Design

2223-1-F9102Q012

Aims

This course aims at providing the students an advanced understanding of the main aspects of Embedded and Cyber-physical Systems (ECPS) with AI capabilities. At the end of this course the students will know the main concepts of ECPS, their most relevant applicative fields, the essential requirements for ECPS in the different applicative fields, the current technologies available to build ECPS with AI capabilities. The students will be able to design embedded applications with AI capabilities ensuring suitable real-time, memory and energy consumption requirements. The acquired knowledge will be applied by realizing some simple example ECPS during the laboratory sessions.

Contents

Embedded and cyber-physical systems (ECPS) in their context. The structure of ECPS: processing units, memories, communication systems, sensors, actuators. AI computation in ECPS. Introduction to real-time scheduling. Designing ECPS with real-time, memory and energy constraints.

Detailed program

Embedded and cyber-physical systems (ECPS) in their context:

- Types of ECPS and their features; Internet of Things (IoT), Wireless Sensor Networks (WSN); application domains; market value and diffusion.
- Basic requirements: timing, reliability, efficiency; requirements in the different application domains; standards.

The structure of ECPS:

- Processing units: CPU and microcontrollers, DSP and GPU, ASIC, programmable logics (FPGA); AI accelerators for embedded systems.
- Memories; Communication systems: GPIO, synchronous and asynchronous serial buses.
- Models of sensors and actuators: Affine models, saturation, harmonic distortion, dynamic range. Analog-to-digital and digital-to-analog converters.

AI computation in ECPS:

- Introduction to Python and Tensor Flow.
- Cloud, edge and endpoint AI. Introduction to Tiny Machine Learning (TinyML).

Introduction to real-time scheduling:

- Definition of real-time system and basic concepts in real-time scheduling: tasks and jobs, periodic, aperiodic e sporadic tasks, metrics, optimality.
- Scheduling for aperiodic and periodic tasks. Scheduling for heterogeneous sets of tasks with sporadic tasks.
- Coordination of tasks sharing resources: critical sections and locking; priority inversion; resource access protocols.

Designing ECPS with real-time, memory and energy constraints:

- Polling and interrupt-driven interaction with peripherals.
- Software architectures: non preemptive (“superloop”) and preemptive (real-time operating system).
- Designing real-time systems with hierarchical state machines.
- Memory-aware and energy-aware software design.

Prerequisites

Basic knowledge of programming in procedural and object-oriented languages (a basic introduction to the C and Python programming languages will be given during the course); basic knowledge of operating system principles (processes, threads, scheduling) and of computer architecture; basic knowledge of principles of software requirements analysis and software design with UML; basic knowledge of the principles of machine learning as taught in the course Advanced Foundations of Artificial Intelligence.

Teaching form

In-presence classes and laboratory practice consisting in the development of project works in small groups. Attendance to classes and laboratories is highly recommended.

Textbook and teaching resource

The topics that are taught in the course are not comprehensively discussed by any single book: Correspondingly, the course has no textbook. The learning material will be provided during the course as handouts, slides, research articles and a selection of chapters from books including:

E. A. Lee, S.A. Seshia. Introduction to Embedded Systems: A Cyber-Physical Approach. Second Edition, MIT Press, 2017.
G. C. Buttazzo. Hard Real-Time Computing Systems, Predictable Scheduling Algorithms and Applications. Third Edition. Springer, 2011.
P. Warden, D. Situnayake. TinyML: Machine Learning with TensorFlow Lite on Arduino and Ultra-Low-Power Microcontrollers. O'Reilly, 2019.

Semester

Second.

Assessment method

Written examination with open questions, problems and exercises; written report of the laboratory activity.

Office hours

By appointment (send an email to the course instructor to agree a date and time for a meeting).

Sustainable Development Goals