AEROSPACE ENGINEERING (LM52)

(Brindisi - Università degli Studi)

Teaching EMBEDDED AND CERTIFIED SOFTWARE MOD.2

GenCod A005159

Owner professor MICHELE RUTA

Teaching in italian EMBEDDED AND CERTIFIED SOFTWARE MOD.2

Teaching EMBEDDED AND CERTIFIED SOFTWARE MOD.2

SSD code ING-INF/05

Reference course AEROSPACE ENGINEERING

Course type Laurea Magistrale

Credits 6.0

Teaching hours Front activity hours: 54.0

For enrolled in 2018/2019

Taught in 2018/2019

Course year 1

Language ENGLISH

Curriculum AEROSPACE ENGINEERING SYSTEMS

Location Brindisi

Semester First Semester

Exam type Oral

Assessment

Course timetable https://easyroom.unisalento.it/Orario

BRIEF COURSE DESCRIPTION

- Software development for embedded systems: general overview. Case study: the GCC compiler and the GDB debugger.
- Software for mobile devices. Case study: iOS and Android. iOS and Android OS architecture. Introduction to mobile Applications development; certificato requirements for the applications distribution.
- Software for robotics. Case study: ROS (Robot Operating System). General overview. Development of a ROS node. Introduction to typical issues in autonomous robots.
- Software for real-time embedded Operating Systems. Case study: OSEK-OS. AUTOSAR (quick overview).
- Model-based embedded software design: general overview. Model checking and statistical model checking. Case study: Uppaal SMC.

REQUIREMENTS

Fundamentals of Computer Science. Fundamentals of Digital Electronics. Knowledge of at least a programming language.

COURSE AIMS

- **Knowledge and understanding**

Main concepts of design, development, test and certification of embedded software for a specific application in mobile, robotic and control systems.

- **Ability to apply knowledge and understanding**

Capability of designino, developing, testing and validating an embedded software according to external requirements (user and systems requirements) and internal requirements (current regulations and certification processes).

- **Ability of evaluation**

Capability of knowledge of problems and ability to identify a proper solution.

- **Ability of speaking**

Capability of communicating with a proper technical language.

- **Learning ability**

Capability of autonomously improving abilities and knowledge.

TEACHING METHODOLOGY

Lectures for presenting general teorethical concepts and models corroborated by selected case studies, examples and exercises.

ASSESSMENT TYPE

End-course written test containing open answer questions, multiple choice questions and practical exercises in order to evaluate learning and speaking abilities as well as the capability of problem understanding and solving.
Optional ora examination on the course contents, only after a positive output of the written test (mark equal or above 18/30).

ASSESSMENT SESSIONS

15/02/2019 ore 11:00 Aula B
24/05/2019 ore 11:00
28/06/2019 ore 11:00
18/07/2019 ore 11:00
12/09/2019 ore 11:00
26/09/2019 ore 11:00
15/11/2019 ore 11:00

OTHER USEFUL INFORMATION

Further e-mail for communications: michele.ruta@poliba.it

FULL SYLLABUS

- Development of software for embedded systems: general overview, software compiling, cross-compiling, development environments, build system. Case study: GCC compiler and debugger GDB.
- Software for mobile devices: general overview. Case studies: iOS and Android. Architecture of the Operating Systems (O.S.) iOS and Android: kernel, layers, runtime environment. Security features and energy management. Introduction to mobile applications development: lifecycle of an application, architectural patterns and basic APIs. Certification requirements for distributing applications on the App Store.
- Software for robotic devices. Case study: ROS (Robot Operating System). General requirements, architecture, publish/subscribe framework, services, package. Development of a ROS node. Introduction to typical issues of autonomous robots: mapping, path planning, path following, motion control. Gazebo simulator and RViz viewer.
- Software for real-time embedded O.S.. Case study OSEK-OS: task development model, OIL language for system configuration specification, task lyfecycle, O.S. features. AUTOSAR platforms Classic and Adaptive (quick overview).
- Model-based embedded software design: general overview, verification and validation, V model. Generali concepts of model checking: automatas and Kripke structures, propositional and temporali logics LTL e CTL, verifiable property types. Statistical model checking. Case study: Uppaal SMC.

REFERENCE TEXT BOOKS

- Teaching resources at the Web page: http://sisinflab.poliba.it/ruta/(link 'Embedded and Certified Software')
- Manuals and tutorials of software tools presented as case studies: GCC and GDB, official documentation for iOS and Android developers, ROS, Catkin, Gazebo, OSEK-OS, RT-Druid, Uppaal SMC.
- A. Silberschatz, P.B. Galvin, G. Gagne, *Operating System Concepts*, Wiley