

Caracterização da Unidade Curricular / Characterization of the Curricular Unit

Designação da Unidade Curricular (UC) / Title of Curricular Unit (CU): Projeto de Sistemas Embebidos / Embedded Systems Design

Área científica da UC / CU Scientific Area: Eletrónica e Automação / Electronics and Automation

Semestre / Semester: 4º

Número de créditos ECTS / Number of ECTS credits: 6

Carga horária por tipologia de horas / Workload by type of hours: TP: 22,5; PL: 45; OT: 9; O: 13,5

Carga letiva semanal / Weekly letive charge: 4,5h

Objetivos de aprendizagem (conhecimentos, aptidões e competências a desenvolver pelos estudantes):

Conceção de sistemas embebidos de pouca complexidade baseados em microcontroladores. Na vertente hardware é estudada uma arquitetura de processador real e os periféricos que o circulam. Na vertente software é estudada a interação com o hardware e o desenvolvimento de aplicações reais.

Os estudantes que terminam com sucesso esta unidade curricular serão capazes de:

- Analisar estruturas de hardware baseadas em microcontroladores.
- Desenvolver software de sistema interface com o hardware.
- Desenvolver software de aplicação com a máquina.
- Testar e depurar o software realizado, usando as ferramentas adequadas, sobre hardware real.
- Instalar programas em memória ROM e FLASH.
- Definir e interpretar especificações de sistemas embebidos.
- Integrar periféricos e desenvolver os respetivos gestores.
- Integrar pacotes de software genéricos.
- Configurar o sistema operativo de acordo com o hardware as necessidades da aplicação.
- Configurar um sistema autónomo, que inicie a sua operação a partir da ligação da energia.

Intended learning outcomes (knowledge, skills and competences to be developed by the students):

Design of low-complexity embedded systems based on microcontrollers. In the hardware aspect, a real processor architecture and the peripherals that circulate it are studied. In the software aspect, we study the interaction with hardware and the development of real applications.

Students who successfully complete this curricular unit will be able to:

- Examine microcontroller-based hardware structures.
- Create a software interface between the system and the hardware.
- Use the machine to create application software.
- Using the appropriate tools, test and debug the software on real hardware.
- Load programs into the ROM and FLASH memory.

- Define and interpret embedded system specifications.
- Integrate peripherals and train managers.
- Integrate general-purpose software packages.
- Configure the operating system according to the hardware needs of the application.
- Set up a stand-alone system that starts operating from the power connection.

Conteúdos programáticos:

1. Microcontroladores

- 1.1 Arquiteturas
- 1.2 Dispositivos de memória (DRAM, SRAM, SDRAM, E2PROM E FLASH)
- 1.3 Interfaces externa de processador (memória externa, SPI, I2C)
- 1.4 Dispositivos de comunicação série
- 1.5 Interfaces de comunicação sem fios
- 1.6 Interação com dispositivos periféricos
- 1.7 Produção de programas
- 1.8 Desenvolvimento de aplicações segundo o modelo de máquinas de estados

2. Sistemas Embebidos

- 2.1 Especificação de sistemas embebidos
- 2.2 Sistemas operativos para sistemas embebidos
- 2.3 Desenho de software para dispositivos com poucos recursos
- 2.4 Estruturação do software segundo padrões
- 2.5 Interfaces com o utilizador (teclado, LCD gráfico, ecrã tátil)

3. *Desenvolvimento de trabalhos com arquitetura AVR (utilizando a plataforma Arduino com interfaces de I/O digital e analógica), explorando o projeto de software em assembly e C, com o apoio de ferramentas de simulação e de emulação. Implementação de pequenos projetos práticos que envolverão componentes de input e output em cenários de informática industrial, automação e telecomunicações.*

4. Desenvolvimento de dois mini-projetos.

Syllabus:

1. Microcontrollers

- 1.1 Architectures
- 1.2 DRAM, SRAM, SDRAM, E2PROM, and FLASH are examples of memory devices.
- 1.3 External processor interfaces (external memory, SPI, I2C)

- 1.4 Devices for serial communication
- 1.5 Interfaces for wireless communication
- 1.6 Interaction with peripheral devices
- 1.7 Production of programs
- 1.8 Application development according to the state machine model

2. Saturated Systems

- 2.1 Soaked system specifications
- 2.2 Operating systems for embedded systems
- 2.3 Software design for low-featured devices
- 2.4 Structuring the software according to standards
- 2.5 Human-machine interfaces (keyboard, graphic LCD, touch screen)

3. Work with the AVR architecture (using the Arduino platform with digital and analog I/O interfaces), experimenting with software design in assembly and C, with the assistance of simulation and emulation tools. of small practical projects that will involve input and output components in industrial computing, automation, and telecommunications scenarios.

4. Creation of two mini-projects.
