Microinformatic systems programming - course description

General information

Course name	Microinformatic systems programming
Course ID	11.3-WE-INFD-MicroinSP-Er
Faculty	Faculty of Computer Science, Electrical Engineering and Automatics
Field of study	Computer Science
Education profile	academic
Level of studies	Second-cycle Erasmus programme
Beginning semester	winter term 2021/2022

Course information

Semester	3
ECTS credits to win	6
Course type	obligatory
Teaching language	english
Author of syllabus	• dr inż. Mirosław Kozioł

Classes forms

The class form	Hours per semester (full-time)	Hours per week (full-time)	Hours per semester (part-time)	Hours per week (part-time)	Form of assignment
Lecture	30	2	-	-	Credit with grade
Laboratory	30	2	-	-	Credit with grade

Aim of the course

- To provide knowledge about the basic peripherals that occur in microinformatic systems and methods of its handling.
- To develop and shape the skills in the software design for microinformatic systems.

Prerequisites

By entering this course, student should know the following isssues:

- fundamentals of programming in the C language,
- fundamental konowledge about analog and digital circuits,
- fundamental knowledge about analog-to-digital and digital-to-analog converters.

Scope

Microinformatic system. Basic components of a microprocessor system. Microinformatic system and microprocessor system. Microcontroller as an example of a microinformatic system.

Cortex-M7 processor. The family of Cortex processors. Architecture of the Cortex-M7 processor: basic functional blocks, buses, programer's model.

STM32F7 microcontrollers as an example of advanced microinformatic system with Cortex-M7 processor. Evolution of STM32 microcontrollers. Architecture of STM32F7 microcontrollers. Processor access paths to the data and program memory. Memory map. Available peripherals. Hardware components supporting debugging and tracing of the program code.

Software development for microinformatic systems. Software development flow. Creating the object code. Tools with a graphical user interface supporting the creation of a program code template for STM32F7 microcontrollers. HAL Libraries. Basic files of the project generated by the STM32CubeMX configurator.

Block of clock signal generation in STM32F7 microcontrollers. Available sources of clock signals. Basic configuration registers. Configuration of the clock signals using the STM32CubeMX program.

General purpose inputs-outputs. I/O ports in STM32F7 microcontrollers. Basic modes of operation of the port lines. Configuration of the port lines with the direct use of registers and through the STM32CubeMX.

Exceptions. The concept of an exception in STM32F7 microcontrollers. NVIC interrupt controller. Interrupt priority. The group priority and subpriority of exceptions. Exception handling. Interrupts and events generated on the port lines.

I2C interface as an example of a local serial interface. Basic interface characteristics. Interface configuration in the STM32F7 microcontrollers. The basic functions of the HAL library to support the interface.

Timers. Main blocks of timers in STM32F7 microcontrollers. Configuration of the timers using the STM32CubeMX. Examples of the practical use of timers.

Cooperation of the microcontroller with analog signals. Configuration and operation of the analog-to-digital (ADC) and digital-to-analog (DAC) converters in STM32F7 microcontrollers. Examples of the practical use of ADC and DAC.

Teaching methods

- Lecture: conventional/traditional lecture.
- Laboratory: laboratory exercises.

Learning outcomes and methods of theirs verification

Outcome description	Outcome symbols	Methods of verification	The class form
Student can write a program for a microcontroller to create an advanced microinformatic system designed for the implementation of selected services and applications.		• an observation and evaluation of the student's practical skills	• Laboratory
Student is able to select and write a program for the proper microcontroller peripherals in order to implement the appropriate functionality of the microinformatic system.		• a multiple choice and open questions test • an observation and evaluation of the student's practical skills	• Lecture • Laboratory
Student is able to use advanced techniques, methods and tools to implement software for microinformatic systems.		• an observation and evaluation of the student's practical skills	• Laboratory

Assignment conditions

Lecture: to receive a final passing grade student has to receive positive grade from final test.

Laboratory: to receive a final passing grade student has to obtain positive grades for all laboratory exercises provided in the laboratory syllabus.

Calculation of the final grade = lecture 45% + laboratory 55%

Recommended reading

1. Yiu J.: The Definitive Guide to ARM Cortex-M3 and Cortex-M4 Processors, Elsevier Science & Technology, 2011.
2. Martin T.: The Designer's Guide to the Cortex-M Processor Family, Elsevier Science & Technology, 2016.

Further reading

Notes

Modified by dr inż. Mirosław Kozioł (last modification: 15-07-2021 18:10)