Microinformatic systems programming - opis przedmiotu

Informacje ogólne

Nazwa przedmiotu	Microinformatic systems programming
Kod przedmiotu	11.3-WE-INFD-MicroinSP-Er
Wydział	Wydział Nauk Inżynieryjno-Technicznych
Kierunek	Informatyka
Profil	ogólnoakademicki
Rodzaj studiów	Program Erasmus drugiego stopnia
Semestr rozpoczęcia	semestr zimowy 2022/2023

Informacje o przedmiocie

Semestr	3
Liczba punktów ECTS do zdobycia	6
Typ przedmiotu	obowiązkowy
Język nauczania	angielski
Sylabus opracował	• dr inż. Mirosław Kozioł

Formy zajęć

Forma zajęć	Liczba godzin w semestrze (stacjonarne)	Liczba godzin w tygodniu (stacjonarne)	Liczba godzin w semestrze (niestacjonarne)	Liczba godzin w tygodniu (niestacjonarne)	Forma zaliczenia
Wykład	30	2	-	-	Zaliczenie na ocenę
Laboratorium	30	2	-	-	Zaliczenie na ocenę

Cel przedmiotu

- 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.

Wymagania wstępne

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

- fundamental konowledge about analog and digital circuits,
- basic electronic graphical symbols used within circuit diagrams and schematics,
- fundamentals of programming in the C language,.

Zakres tematyczny

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.

Metody kształcenia

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

Efekty uczenia się i metody weryfikacji osiągania efektów uczenia się

Opis efektu	Symbole efektów	Metody weryfikacji	Forma zajęć
Student can write a program for a microcontroller to create an advanced microinformatic system designed for the implementation of selected services and applications.		• obserwacje i ocena umiejętności praktycznych studenta	• Laboratorium
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.		• obserwacje i ocena umiejętności praktycznych studenta • test z pytaniami zamkniętymi i otwartymi	• Wykład • Laboratorium
Student is able to use advanced techniques, methods and tools to implement software for microinformatic systems.		• obserwacje i ocena umiejętności praktycznych studenta	• Laboratorium

Warunki zaliczenia

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%

Literatura podstawowa

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.
3. Yifeng Z.: *Embedded Systems with Arm Cortex-M Microcontrollers in Assembly Language and C*. E-man Press LCC, Third edition, 2017.

Literatura uzupełniająca

Uwagi

Zmodyfikowane przez dr inż. Mirosław Kozioł (ostatnia modyfikacja: 12-04-2022 14:05)