COURSE DESCRIPTION CARD - SYLLABUS

Course name

Embedded System Architecture

Course

Field of study

Computer Science

Area of study (specialization)

Computer microsystems

Level of study

Second-cycle studies

Form of study

full-time

Year/Semester

1/1

Profile of study

general academic

Course offered in

polish

Requirements

compulsory

Number of hours

Lecture

30

Tutorials

Laboratory classes

30

Projects/seminars

Other (e.g. online)

Number of credit points

5

Lecturers

Responsible for the course/lecturer:

dr inż. Adam Turkot

email: adam.turkot@put.poznan.pl

tel. 61 6652284

ul. Piotrowo 3a, 60-965 Poznań

Responsible for the course/lecturer:

Prerequisites

Students commencing this course should have basic knowledge of UDP, SSH, ICMP, C and C++ programming protocols, Bash command interpreter, Python and JavaScript. He/she should also understand the need to broaden his/her competences / be ready to start cooperation within the team. Moreover, in terms of social competences, the student must present such attitudes as honesty, responsibility, perseverance, cognitive curiosity, creativity, personal culture, respect for other people.

Course objective

Familiarizing students with the methodology of embedded system architecture design. To provide students with extended knowledge of single-processor, multi-processor, multi-computer embedded system architectures. Developing skills in programming technology ensuring: efficient use of hardware resources of embedded systems; optimal, for a given task, implementation of applications using a microcontroller, with support for dedicated peripheral modules and taking into account requirements related to energy saving and computing efficiency. Mastering communication techniques between microcontrollers and digital and analogue components of embedded systems. To familiarize students with the possibilities and limitations of building embedded systems based on signal processors and personal computers. Developing students' teamwork skills by implementing project elements and combining them into a whole.

Course-related learning outcomes

Knowledge

1. Has advanced and in-depth knowledge of widely understood IT systems and methods and tools used for their implementation.
2. Has advanced detailed knowledge of selected issues in the field of IT.
3. Has an advanced and detailed knowledge of the processes taking place in the life cycle of IT systems.

Skills

1. Is able, in formulating and solving engineering tasks, to integrate knowledge from different fields of computer science (and, if necessary, knowledge from other scientific disciplines) and apply a system approach that also takes account of non-technical aspects.
2. Can evaluate the usefulness and applicability of new developments (methods and tools) and new IT products.
3. Is able to critically analyse existing technical solutions and propose improvements (improvements) to them.
4. Can assess the suitability of methods and tools for solving an engineering task involving the construction or evaluation of an IT system or its components, including the limitations of those methods and tools.
5. Can, according to a specification, design a complex device, IT system or process and carry out the project using appropriate methods, techniques and tools, including adapting existing ones or developing new ones for that purpose.

Social competences

1. Understands that in computer science knowledge and skills become obsolete very quickly.
2. Understands the importance of using the latest knowledge in computer science to solve research and practical problems.

Methods for verifying learning outcomes and assessment criteria

Learning outcomes presented above are verified as follows:

Formal evaluation:

(a) for lectures: on the basis of answers to questions concerning the material discussed in previous

lectures
(b) for laboratorys/exercises: based on an assessment of the current progress of the tasks,
Summary evaluation:
a) in the scope of lectures, the verification of the assumed educational results is carried out by: an oral examination combined with the defence of the project, in case of doubt a written part (an electronic test on the Moodle platform);
b) as far as laboratories are concerned, the verification of the assumed educational results is carried out by: oral examination combined with the project defence: evaluation of the student's preparation for individual sessions of laboratory classes by checking the preparation of given projects/exercises and evaluation of skills related to the implementation of laboratory classes, continuous evaluation, during each class (oral answers), rewarding the increase in the ability to use the learned principles and methods, evaluation of documentation created systematically along with the progress of project work; documentation prepared partly during the classes and partly after their completion; this evaluation also includes the ability to work in a team, evaluation and defence by the student of the report on project implementation,

Programme content

Basics of Embedded System Architecture . Techniques for the effective use of hardware resources . Assessment of hardware capabilities . Programming environment . Techniques of programming . Code optimization methods . User Interface . Hardware and programming solutions to manage power consumption. Software security techniques (program integrity, resistance to unauthorized copying). Computer architectures. Multi-processor and multi-computer architectures. Increased performance of multi-processor and multi-computer architecture compared to monoprocessor systems. Master's of multiprocessor systems. Local and shared resources, consequences of resource sharing. Typical STE, MULTIBUS, VME, PCI, COMPACT PCI bus solutions. Management of access to shared resources. Masters in distributed systems. Layers 4 and 7 of ISO communication model. Hardware and software techniques for increased reliability of the communication link. Communication protocols used in the cloud. Increasing the reliability of unmanned systems, techniques for energy management of autonomous systems.
The laboratory classes are conducted in the form of fifteen 2-hour exercises, held in the laboratory, preceded by a 2-hour instructional session at the beginning of the semester. The exercises are carried out by 2-person teams of students.
Laboratories include Introduction to Intel Galileo programming. Installation of the system and drivers on the Intel Galileo platform. Intel IoT Analytics in embedded systems. The use of a web server as a communication interface with the embedded system. Use of logical state analyzer in the construction and testing of embedded systems. Cloud communication (oAuth2 authorization, JSON, Google Drive). Raspberry Pi programming in C and Python. Programming of SOM architecture on the example of Colibri iMX7.

Teaching methods

Teaching methods:
lecture: multimedia presentation, presentation illustrated with examples given on the board, presentation of selected student solutions.
laboratory exercises: practical exercises, performing experiments, discussion, teamwork.

Bibliography

Basic
Linux w systemach embedded Bis, Marcin. Wydawnictwo btc, 2011

Building embedded Linux systems Yaghmour, Karim. O'Reilly, cop. 2003.

Additional
Wbudowane systemy mikroprocesorowe Timofiejew, Aleksander., Akademia Podlaska (Siedlce).

Wydawnictwo.Wydawnictwo Akademii Podlaskiej, 2010.

Breakdown of average student's workload

	Hours	ECTS
Total workload	120	5
Classes requiring direct contact with the teacher	60	2,5
Student's own work (literature studies, preparation for laboratory classes/tutorials, preparation for tests/exam, project preparation) [1]	60	2,5

[1] delete or add other activities as appropriate