POZNAN UNIVERSITY OF TECHNOLOGY

EUROPEAN CREDIT TRANSFER AND ACCUMULATION SYSTEM (ECTS)

COURSE DESCRIPTION CARD - SYLLABUS

Course name
Internet of Things applications [N1AiR2>PO8-AIR]

Course

Field of study	Year/Semester
Automatic Control and Robotics	4/7
Area of study (specialization)	**Profile of study**
–	general academic
Level of study	**Course offered in**
first-cycle	Polish
Form of study	**Requirements**
part-time	elective

Number of hours

Lecture	Laboratory classes	Other
10	20	0
Tutorials	**Projects/seminars**	
0	0	

Number of credit points

3,00

Coordinators

dr inż. Dominik Łuczak
dominik.luczak@put.poznan.pl

Lecturers

Prerequisites

Knowledge: A student starting this subject should have basic knowledge of electronics and basic programming. Skills: The student should have the ability to solve basic problems in the field of digital signal processing and the ability to obtain information from specified sources. He should also understand the need to expand his competences and be ready to cooperate in a team. Social competences: In addition, in the area of social competences, the student must exhibit such qualities as honesty, responsibility, perseverance, cognitive curiosity, creativity, personal culture, respect for other people.

Course objective

The aim of the course is to learn the theoretical foundations, principles of operation and possible applications of devices communicating with other objects of a distributed control system without human intervention (Machine to Machine, M2M), both wired and wirelessly via the global Internet network or local network. After completing the education, the student should be able to: 1) develop a simple application for an edge device allowing remote management of a selected item connected to the global network, 2) develop a device enabling communication with other objects on the Internet or local network, 3) prepare a user interface for remote management of one or several objects.

Course-related learning outcomes

Knowledge:
1. The student has advanced knowledge of selected algorithms and data structures as well as procedural and object-oriented programming methodology and techniques; - [K1_W8],
2. has structured knowledge of computer architectures, computer systems and networks as well as operating systems, including real-time operating systems; - [K1_W9],
3. knows and understands typical engineering technologies, principles and techniques for constructing simple automation and robotics systems; knows and understands the principles of selection of executive systems, computational units as well as measuring and control elements and devices - [K1_W20],
4. is aware of the current state and the latest development trends in the field of automation and robotics - [K1_W21],
5. knows and understands the fundamental dilemmas of modern civilization associated with the development of automation and robotics - [K1_W28].

Skills:
1. The student is able to design and practically use simple diagnostic and decision systems dedicated to automation and robotics systems; - [K1_U21],
2. is able to select the type and parameters of the measurement system, control unit and peripheral and communication modules for the selected application and integrate them in the form of the resulting measurement and control system - [K1_U22],
3. is able to identify and formulate specifications for simple engineering tasks in the field of automation and robotics; - [K1_U23],
4. is able to develop a solution to a simple engineering task and implement, test and run it in a selected programming environment on a PC for selected operating systems - [K1_U26].

Social competences:
1. The student is aware of the importance and understands the non-technical aspects and effects of engineering activities, including its impact on the environment and the associated responsibility for decisions; is ready to care for the achievements and traditions of the profession; - [K1_K2]
2. is aware of the need for a professional approach to technical issues, meticulous familiarization with the documentation and environmental conditions in which the devices and their elements can function - [K1_K5]

Methods for verifying learning outcomes and assessment criteria

Learning outcomes presented above are verified as follows:

Formative assessment:
a) in terms of lectures:
based on answers to questions about the material covered in previous lectures,
b) within the scope of the laboratory:
based on the assessment of knowledge and understanding of current issues presented in the course.
c) In both forms of classes, it is possible to use Problem-Based Learning (PBL) tasks that support the current research and technical needs of the course coordinator and are supervised by the instructor, taking into account the iterative and cyclical nature of task implementation, provided that they are consistent with the course content.

Summary rating:
a) in the field of lectures, verification of the assumed learning outcomes is carried out by:
i. assessment of knowledge and skills demonstrated on a multiple-choice test.
ii. discussion of the results,
b) in the laboratory, verification of the assumed learning outcomes is carried out by:
i. development of an IoT control and measurement system in its basic version.
ii. development of an IoT control and measurement system in an extended version.
c) The summative assessment may include the results of Problem-Based Learning (PBL) assignments developed for the research and technical needs of the course coordinator and supervised by the instructor, provided they are consistent with the course curriculum.
d) Obtaining additional points for activity during classes, in particular for:
i. independent construction of the electronic module of the control and measurement system and preparation of documentation
ii. effectiveness of applying acquired knowledge when solving a given problem
iii. comments related to the improvement of teaching materials.

Programme content

The subject covers a wide range of issues related to the Internet of Things (IoT), from basic elements and application tasks, to system construction, development modules, programming environments and data exchange formats. Students also learn web applications with a RESTful interface, configuration of Linux operating systems with selected hardware elements, e.g. GPIO, SPI, UART, I2C, creating a server application for devices with limited resources, structure and interaction of web interfaces (e.g. HTML, jQuery, CSS, Bootstrap), selected network protocols (e.g. Ethernet, IP, TCP, UDP, DHCP, ARP, HTTP, MQTT) and security issues.

Laboratory classes give students the opportunity to practically apply the acquired knowledge. They include familiarization with the equipment, occupational health and safety rules, design environment, JSON format, development of a RESTful interface, configuration of the Linux system, development of applications for Linux to support selected GPIO, SPI, UART, I2C interfaces, user interface design, development of web interfaces (e.g. .HTML, jQuery, Bootstrap) with interactive elements, with the possibility of using signal processing algorithms, as well as development and testing of the control and measurement system.

The course program provides comprehensive preparation for working with Internet of Things applications. Students will gain theoretical knowledge and practical skills necessary to design, create and implement IoT edge devices in various fields.

Course topics

The lecture program covers the following topics:
1. Elements and tasks of Internet of Things applications. Data exchange format between systems - JSON, REST interface (representational state transfer), HTTP client.
2. REST application in PHP. HTTP server on IoT and REST device.
3. Linux operating system configuration - GPIO, SPI, UART, I2C configuration.
4. Linux system - access rights, shell scripts, creating users, SSH remote access.
5. Structure of the web interface – HTML, jQuery. Web interface interaction - JS, jQuery
6. Presentation of the web interface - CSS, Bootsrap.
7. Selected communication protocols (e.g. Ethernet, IP, TCP, UDP, TLS, DHCP, ARP, HTTP, MQTT).

The laboratory curriculum covers the following topics:
1. Organizational classes - familiarization with health and safety equipment and footnotes, introduction to the design environment. JSON data exchange format.
2. Parsing JSON data.
3. REST application in PHP. HTTP server on the IoT edge device.
4. HTTP client.
5. CLI (Command Line Interface) for LED arrays. REST interface for LED arrays.
6. Command line interface for sensors. REST interface for sensors.
7. I2C interface. REST interface for a sensor with an I2C interface.
8. Web user interface - dynamic generation of the interface structure for sensors (jQuery and HTML).
9. Web user interface - dynamic generation of the interface structure for LED arrays (jQuery and HTML).
10. Web user interface - presentation layer (CSS and Bootstrap).
11. Final application including a basic application with the possibility of extension with additional modules, e.g. MQTT, WS, HTTPS, CRON, a single-board computer via SPI or UART manages the NUCLEO-ST32F7 board implementing automatic regulation applications.

Teaching methods

1. Lecture: multimedia presentation illustrated with computer simulations and a real system
2. Laboratory classes: the use of single-board development modules with a set of sensors, programming environments for Internet applications
3. Both forms of instruction offer the opportunity to incorporate elements of Problem-Based Learning (PBL), in which students work on problems and projects defined for the research and technical needs of the course coordinator and supervised by the instructor. This approach places particular emphasis on the iterative nature of work, encompassing problem analysis, solution design, practical verification, and systematic refinement.

Bibliography

Basic
1. Ilya Grigorik, Wydajne aplikacje internetowe. Przewodnik, Helion, 2014 / Ilya Grigorik, High Performance Browser Networking, 2013
2. Justin Hutchens, Skanowanie sieci z Kali Linux : receptury : bezpieczeństwo sieci w Twoich rękach!, Helion, 2015 / Justin Hutchens, Kali Linux Network Scanning Cookbook, 2014
3. Adam Gerber, Clifton Craig, Android Studio : wygodne i efektywne tworzenie aplikacji, Helion, 2016 / Adam Gerber, Clifton Craig, Learn Android Studio Build Android Apps Quickly and Effectively, 2015
4. Adrian McEwen, Hakim Cassimally, Designing the Internet of Things, Wiley, 2013.
Additional
1. Łuczak, D. Machine Fault Diagnosis through Vibration Analysis: Continuous Wavelet Transform with Complex Morlet Wavelet and Time–Frequency RGB Image Recognition via Convolutional Neural Network. Electronics 2024, 13, 452, doi:10.3390/electronics13020452.
2. Łuczak, D.; Brock, S.; Siembab, K. Cloud Based Fault Diagnosis by Convolutional Neural Network as Time–Frequency RGB Image Recognition of Industrial Machine Vibration with Internet of Things Connectivity. Sensors 2023, 23, 3755, doi:10.3390/s23073755.
3. Łuczak D., Remote laboratory with WEB interface, Computer Applications in Electrical Engineering, Vol. 9, str. 257-268, Poznań, 2011, ISSN 1508-4248.
4. Karol Rogowski, Świat poza jQuery : biblioteki : AngularJS, KnockoutJS i BackbonesJS, Wydawnictwo Naukowe PWN, 2014.

Breakdown of average student's workload

	Hours	ECTS
Total workload	75	3,00
Classes requiring direct contact with the teacher	30	1,00
Student's own work (literature studies, preparation for laboratory classes/ tutorials, preparation for tests/exam, project preparation)	45	2,00