# GDAŃSK UNIVERSITY OF TECHNOLOGY

## Subject card

| | |
|---|---|
| Subject name and code | Physical basis of microcontroller measurement systems, PG_00051072 |
| Field of study | Technical Physics |

| | | | |
|---|---|---|---|
| Date of commencement of studies | October 2022 | Academic year of realisation of subject | 2024/2025 |
| Education level | first-cycle studies | Subject group | Optional subject group |
| Mode of study | Full-time studies | Mode of delivery | at the university |
| Year of study | 3 | Language of instruction | Polish |
| Semester of study | 6 | ECTS credits | 3.0 |
| Learning profile | general academic profile | Assessment form | assessment |

| | |
|---|---|
| Conducting unit | Division of Theoretical Physics and Quantum Informaton -> Institute of Physics and Applied Computer Science -> Faculty of Applied Physics and Mathematics |

| Name and surname of lecturer (lecturers) | Subject supervisor | dr inż. Michał Piłat |
|---|---|---|
| | Teachers | dr inż. Michał Piłat |
| | | dr inż. Bartosz Reichel |

| Lesson types and methods of instruction | Lesson type | Lecture | Tutorial | Laboratory | Project | Seminar | SUM |
|---|---|---|---|---|---|---|---|
| | Number of study hours | 15.0 | 0.0 | 0.0 | 30.0 | 0.0 | 45 |
| | E-learning hours included: 0.0 | | | | | | |

| Learning activity and number of study hours | Learning activity | Participation in didactic classes included in study plan | Participation in consultation hours | Self-study | SUM |
|---|---|---|---|---|---|
| | Number of study hours | 45 | 5.0 | 25.0 | 75 |

| | |
|---|---|
| Subject objectives | The aim of the course is to familiarize students with the structure and basic methods of designing and programming simple embedded systems based on microcontrollers, i.e. specialized IT systems responsible for performing strictly defined tasks - mainly related to monitoring and control. The physical basis of the functioning of sensors (receptors) and actuators (effectors / actuators) as the basic components of this type of systems will be discussed. |

| Learning outcomes | Course outcome | Subject outcome | Method of verification |
|---|---|---|---|
| | K6_W06 | The student is able to describe the functioning of an embedded system on the basis of electronics. | [SW1] Assessment of factual knowledge [SW3] Assessment of knowledge contained in written work and projects |
| | K6_U05 | The student is able to design, build and program simple, specialized embedded systems based on selected microcontrollers. | [SU1] Assessment of task fulfilment [SU2] Assessment of ability to analyse information [SU4] Assessment of ability to use methods and tools [SU5] Assessment of ability to present the results of task |
| | K6_W02 | The student is able to explain the physics of the functioning of selected electronic components (receptors and effectors) used in embedded systems. | [SW1] Assessment of factual knowledge [SW3] Assessment of knowledge contained in written work and projects |
| | K6_U06 | The student is able to estimate the technical and economic feasibility of building an embedded system. | [SU2] Assessment of ability to analyse information [SU3] Assessment of ability to use knowledge gained from the subject |

| Subject contents | Lecture: |
|---|---|
| | Discussion of the physical basis of operation of basic measurement systems based on selected microcontrollers.<br>Detailed programme<br>A. Hardware platforms:<br>a. Arduino (Uno, Mega, Nano, Pro Mini, etc.) - a platform based on 8-bit microcontrollers from the Atmel AVR family<br>b. Raspberry Pi - a family of 32/64-bit single board minicomputers based on microprocessors from the ARM family (11, Cortex)<br>c. ESP8266 / ESP32 - 32-bit RISC microcontroller with built-in WiFi / Wifi+Bluetooth transmission, making it convenient to connect the system to the Internet<br>d. Raspberry Pi Pico - a microcontroller board based on the Raspberry Pi RP2040 chip<br>e. STM32 - a family of 32-bit microcontrollers from the ARM Cortex family<br><br>B. Programming languages:<br>a. C/C++ - in the context of programming for Arduino / ESP8266 / STM32<br>b. Python - a general-purpose scripting language, particularly convenient for programming for the Raspberry Pi<br>c. MicroPython - a version of Python for microcontrollers<br><br>C. Programming tools and support hardware:<br>a. Arduino IDE - a development environment for Arduino and ESP8266, including the necessary libraries<br>b. NodeMCU - a framework/circuit software to facilitate ESP8266 programming in the context of the Internet of Things<br>c. Sensors (e.g. temperature, pressure, distance, rain, motion, colour, sound), motors, LCD/LED displays, GPS, GSM, RFID devices, camera support, etc.<br>d. Web services for the collection and presentation of measurement data<br><br>D. Description and mechanisms of sensors<br><br>• Inductive and Hall sensor<br>• Potentiometer sensor and thermistors<br>• Tensometric sensors<br>• Capacitance and piezoelectric sensors<br>• Ultrasound, radar and lidar sensors<br>• Photoelectric sensors<br>• Electrolytic-resistic sensor<br><br>Most topics will be illustrated with a demonstration of the operation of a particular device / hardware / programming language.<br><br>Project:<br>During the first weeks, students will learn, under the supervision of the instructor, how to work with the hardware and will build and program basic systems (e.g. control of LEDs, buzzer, motor, relay, measurement of temperature, atmospheric pressure, distance, recognition of movement, rain, colour, use of GPS, GSM and RFID devices, etc.).<br><br>Then, students will present self-prepared prototypes of embedded systems projects<br>for the chosen platform (including implementation), e.g.<br>- internet radio with remote control (Raspberry Pi + WiFi + remote control / infrared sensor / Python)<br>- room access control system (Arduino + RFID / C++) weather station with WWW reporting (Arduino + selected sensors + ESP8266 / C++)<br>- room monitoring with WWW reporting (Arduino + selected sensors + ESP8266 / C++)<br>- wheeled self-driving robot (Arduino + distance and obstacle sensors + robot platform with motors and wheels / C++)<br>- home automation (Arduino or Raspberry Pi + sensors + effectors / C++ or Python)<br>- webcam with object recognition (Raspberry Pi + camera / Python + OpenCV library)<br><br>Hardware (microcontroller boards, sensors, effectors, etc.) will be made available to students for the duration of their projects. It is also possible to design the system in TinkerCad or an equivalent service. |
| Prerequisites<br>and co-requisites | Basic knowledge of programming<br>Basic knowledge of electronics |

| Assessment methods and criteria | Subject passing criteria | Passing threshold | Percentage of the final grade |
|---|---|---|---|
| | Completion of the project | 50.0% | 80.0% |
| | Passing the theoretical part | 50.0% | 20.0% |

| Recommended reading | Basic literature | Michael J. McGrathCliodhna Ní Scanaill, Sensor Technologies. Healthcare, Wellness, and Environmental Applications. Apress, 2013<br>Andy King, Programming the Internet of Things. O'Reilly Media, 2021<br>Vedat Ozan Oner, Developing IoT Projects with ESP32. Packt Publishing, 2021 |
|---|---|---|
| | Supplementary literature | Husan Mahey, Robotic Process Automation with Automation Anywhere. Packt Publishing, 2020 |

| | eResources addresses | Podstawowe |
|---|---|---|
| | | https://enauczanie.pg.edu.pl/moodle/course/view.php?id=44283 - Course on eNauczanie |
| | | Adresy na platformie eNauczanie: |
| Example issues/ example questions/ tasks being completed | - Internet radio with remote control (Raspberry Pi + WiFi + remote control / infrared sensor / Python) <br> - room access control system (Arduino + RFID / C ++) weather station with reporting on the website (Arduino + selected sensors + ESP8266 / C ++) <br> - room monitoring with reporting on the website (Arduino + selected sensors + ESP8266 / C ++) <br> - wheeled self-propelled robot (Arduino + distance and obstacle sensors + robot platform with motors and wheels / C ++) <br> - home automation (Arduino or Raspberry Pi + sensors + effectors / C ++ or Python) <br> - webcam with object recognition (Raspberry Pi + camera / Python + OpenCV library) | |
| Work placement | Not applicable | |

Document generated electronically. Does not require a seal or signature.