1. INFORMATION ABOUT THE COURSE
    A. Basic information

| | |
|---|---|
| Name of course | Industrial Internet of Things |
| Study level | First degree |
| Unit running the study programme | Faculty of Telecommunication, Computer Science and Electrical Engineering |
| Study programme | |
| Speciality | |
| Name of teacher (s) and his academic degree | Gracjan Kątek MSc Eng. |
| Introductory courses | none |
| Prerequisites | Basis of programming (C/C++ or Python), basic knowledge of network protocols |

    B. Semester/week schedule of classes

| Semester | Lectures | Classes | Laboratories | Project | Seminars | Field exercises | ECTS |
|---|---|---|---|---|---|---|---|
| winter or summer | | | 30 | | | | 2 |

2. EFFECTS OF EDUCATION (acc. to National Qualifications Framework)

| | |
|---|---|
| Knowledge | Student has deep knowledge about the architecture of Industrial IoT systems (Edge-Fog-Cloud), industrial communication interfaces, time-series data storage, and hardware-level cybersecurity challenges in Industry 4.0. |
| Skills | Student can independently design, implement, and secure an end-to-end IIoT system. Student can configure edge nodes (ESP32, Pico), industrial gateways (Raspberry Pi 5), program utilizing industrial protocols (MQTT, Modbus, OPC UA), and create operational dashboards. |
| Competences | During the implementation of the project, the student is aware of technical standards, documentation reliability, data security regulations, linguistic correctness, and timely delivery of subsequent project milestones. |

3. TEACHING METHODS

| |
|---|
| multimedia lecture, laboratories and project |

4. METHODS OF EXAMINATION

| |
|---|
| written and/or oral exam |

5. SCOPE

| | |
|---|---|
| Lectures | • Introduction to IIoT and Industry 4.0: Differences between IoT and IIoT. Reference architectures (Edge, Fog, Cloud).<br>• Edge Layer Hardware & Security: Microcontrollers vs. SBCs. Hardware security analysis based on ESP32 and RP2350 (Raspberry Pi Pico 2). Secure boot concepts.<br>• IoT Communication Protocols: In-depth analysis of MQTT (QoS, Retain, Last Will) and HTTP REST APIs.<br>• Industrial Standards & Integration: Modbus RTU/TCP protocols and OPC UA architecture for modern factories.<br>• Fog/Edge Computing with Raspberry Pi 5: Role of a local gateway, Linux environment configuration, and containerization concepts (Docker) in industrial networks.<br>• Time-Series Databases & Data Visualization: Relational vs. Time-Series Databases (InfluxDB) for continuous sensory data streams.<br>• Cybersecurity in IIoT |
| Laboratories | Practical implementation of the tasks discussed in the lecture using a rich hardware ecosystem.<br>• Lab 1: Hardware Environment Setup: Visual Studio Code + PlatformIO / MicroPython configuration. Architecture comparison of Pico (RP2040) vs Pico 2 (RP2350) vs ESP32.<br>• Lab 2: Industrial Sensors & Reliability: Multi-sensor data acquisition (I2C, SPI, 1-Wire, 4-20mA via ADC). Implementation of software filtering and Watchdog Timers (WDT) for fault tolerance.<br>• Lab 3: Wireless M2M Communication: Wi-Fi deployment on ESP32, MQTT/REST client implementation, deep sleep modes, and local flash memory data buffering during connection drops.<br>• Lab 4: Modbus Protocol Integration: Using Raspberry Pi Pico to simulate a Modbus RTU/TCP industrial actuator/slave device.<br>• Lab 5: Gateways & Message Routing: Setup of Raspberry Pi 5 as an industrial gateway, installing Mosquitto MQTT broker, and utilizing Node-RED for local data orchestration.<br>• Lab 6: Containerization & Fog Analytics: Deploying Docker containers on Raspberry Pi 5. Setting up |

| | |
|---|---|
| | InfluxDB (Time-Series) and connecting it to the gateway stream.<br>• Lab 7: Secure Visualization & TLS: Enforcing MQTTS (TLS encryption) between ESP32/Pico nodes and the Raspberry Pi 5 gateway. Configuring Grafana dashboards with critical alerting rules. |
| Project | Individual or small-team engineering project involving the design and implementation of a complete End-to-End Industrial Monitoring and Control System.<br>Students must utilize the provided hardware to build a resilient chain: Sensory Edge Nodes (Pico/ESP32 with Watchdog and local caching) → Industrial Protocol Transmission (MQTT/Modbus) → Secured Industrial Gateway (Raspberry Pi 5 with Dockerized Mosquitto & InfluxDB) → Enterprise Visualization Dashboard (Grafana with alert thresholds). The project requires delivering a functional prototype and a short technical engineering report |

6. LITERATURE

| | |
|---|---|
| Basic literature | Programming with MicroPython. Embedded Programming with Microcontrollers and Python, Nicholas H. Tollervey, O'Reilly Media Inc. 2017.<br>Or<br>Programming with STM32: Getting Started with the Nucleo Board and C/C++, Donald Noris, March 2018, McGraw-Hill Education TAB ISBN: 9781260031324 |
| Supplementary literature | Get started with MicroPython on Raspberry Pi Pico G.Halfacree, B. Everard, Raspberry Pi Trading Ltd. 2021. |