

Robotic Programming Environments (winter 2024/2025)

MSc Łukasz Janiec

Wrocław University of Science and Technology
Faculty of Electronics, Photonics and Microsystems
Department of Cybernetics and Robotics

October 7, 2024



Wrocław University
of Science and Technology



HR EXCELLENCE IN RESEARCH

unite!



University Network for Innovation,
Technology and Engineering

Outline of the course plan

- 1 Information
- 2 Special needs
- 3 Course credit rules
- 4 Health and safety in the laboratory
- 5 What is this course about?

MSc Łukasz Janiec

- room 310/C-3
- e-mail: lukasz.janiec@pwr.edu.pl
- <https://kcir.pwr.edu.pl/~ljaniec/>

Consultation hours, winter 2024/2025

- Wednesday 18:00-19:00
- Thursday 10:00-11:00

Additional consultation hours are arranged on an individual basis. The need for a consultation should be notified in advance by e-mail with a **full description of the problems** encountered, a **list of questions** and **attempts made to resolve them independently**.

<https://dontasktoask.com/>

Special needs

If you have special needs due to your health, disability, or other reasons that affect how you participate in class or access materials, please let me know in a private chat or email. I'll make sure everyone has an equal opportunity to learn, without lowering standards.

Important

Accommodating special needs doesn't mean making things easier. It's more about ensuring fair educational opportunities for everyone.

First, a few comments:

- Lecturer website with laboratory exercises:
<https://www.mdrwiega.com/edu/rpe>
- More material will be added later to my website.
- Access to the free Datacamp Classroom is provided to you -important for learning git and Python used in the course.
- Presentation of your own ROS-based (or similar) robotic project can get you extra credit (e.g. points for 5.5 if sufficiently advanced). Similar with any meaningful open-source contributions.
- Up to one properly excused absence from classes is allowed.

Requirements

- student attendance (!) AND preparation work before the laboratory according to the course materials (external resources, lecture examples, instructions, etc.).
- deadline is **at the end of the laboratory**, you have to put your work on `git.kcir.pwr.edu.pl` (step-by-step instructions *how-to* later in the presentation)

Evaluation Score	Total point percentage
3.0	50+%
3.5	60+%
4.0	70+%
4.5	80+%
5.0	90+%

Usually one task per 1–2 classes, 10 - 20 pts each. For a task after the deadline, you lose $n \times 30\%$ total points, where n - the number of weeks of delay.

If you do not publish the code and work in the Gitlab repository, you get 0 pts.

Health and safety in the laboratory

- Use common sense.
- Communicate any hardware or software errors to me or technicians.
- Inform me if you are not feeling well.
- Consuming food and beverages near computer equipment is prohibited. Spills can cause damage and create electrical hazards.
- Try to work on the university PC, do not bring your own laptop. Use your LDAP account and our GitLab.
- Unauthorized modifications to electrical equipment are strictly prohibited. Don't pull the PC plugs from the sockets!

Please note that a local **student** account with a **student** password (for emergency login for students without a personal LDAP account) is automatically cleared of all changes made when you logout.

(Gather WSH signatures from students)

What is this course about?

We will learn about the fundamentals needed in the robotic projects.

Some key abilities

- component-based software engineering - built from loosely-coupled & reusable modular components that communicate via interfaces
- communication between modules - like MQTT protocol or ZeroMQ, asynchronous messaging library (" *The Bulk of Software Engineering is Just Plumbing*"), with good coding practices (e.g. `pydantic`, Google protocol buffers and so on)
- ROS 2 proficiency - concepts, architecture, popular packages, simulations
- OROCOS*, ROS 1 etc. - some legacy frameworks for real-time* control of the robot

Commands - GitLab (git.kcir.pwr.edu.pl)

- On the website - create a repository named `rpe_album_number`, e.g., `rpe_209223`
- Add me (`ljaniec`) as a Maintainer to the repository members
- Clone the repository locally (`git clone your-git-https`) on your PC
- Inside the repository `rpe_student_number`, create a folder `EX0` (`mkdir EX0`)
- Save the signed statement file (`https://kcir.pwr.edu.pl/~ljaniec/statement.txt`) in `EX0`
- Use `git status` to check for local changes to be added
- Add the files for tracking (from the main folder `git add .`)
- Use `git status` to check the tracked changes
- Add a commit with a message (`git commit -m ''EX0 Work in class''`) – if you haven't done this before, configure your local account according to the displayed instructions
- Use `git status` and `git log` to check the added commit
- Push the changes from the local repository to the remote repository via `git push`
- Use `git status` to check again
- Check if you can see the changes in the remote repository

Questions?



Feel free to ask questions.