



COURSE DESCRIPTION CARD - SYLLABUS

Course name

Basic tools and software for autonomous robots [N2AiR1-RiSA>PNiMPRA]

Course

Field of study

Automatic Control and Robotics

Year/Semester

1/1

Area of study (specialization)

Autonomous Robots and Systems

Profile of study

general academic

Level of study

second-cycle

Course offered in

Polish

Form of study

part-time

Requirements

compulsory

Number of hours

Lecture

20

Laboratory classes

20

Other

0

Tutorials

0

Projects/seminars

0

Number of credit points

4,00

Coordinators

dr hab. inż. Dominik Belter prof. PP
dominik.belter@put.poznan.pl

Lecturers

Prerequisites

A student starting this course should have basic knowledge of robotics and programming. He or she should also have the ability to obtain information from various sources and be ready to cooperate within the team.

Course objective

To provide students with knowledge about the tools used to program autonomous robots, the correct use of these tools and the integration of control systems.

Course-related learning outcomes

Knowledge

The graduate has an advanced and enhanced knowledge of methods of analysis and design of control systems (K2_W7 [P7S_WG])

Has an organized and in-depth knowledge related to of control systems and control and measurement systems (K2_W11 [P7S_WG])

Has a basic knowledge of the life cycle of automation and robotics systems as well as control and measurement systems (K2_W13 [P7S_WG])

Skills
Is able to use information and communication technology (K2_U08 [P7S_UK])
The graduate is able to integrate and program specialized robotized systems (K2_U12 [P7S_UW])
Is able to critically analyse how control systems and robotics systems work; also have the ability to select automation systems using microprocessor controllers (K2_U19 [P7S_UW])
The graduate can construct an algorithm for solving a complex and unusual engineering task and a simple research problem, as well as implement, test and run it in a selected development environment for selected operating systems (K2_U25 [P7S_UW])

Social competences
Is aware of the necessity of a professional approach to technical tasks, in-depth familiarization with documentation and environmental conditions in which devices and their components may operate (K2_K4 [P7S_KR])

Methods for verifying learning outcomes and assessment criteria

Learning outcomes presented above are verified as follows:

Learning outcomes presented above are verified as follows:

The knowledge acquired during the lecture is verified by one 45-minute exam conducted in an examination session. The examination consists of 20-30 (test) questions and up to 5 open questions, differently scored. The credit threshold: 50% of points. Issues for the examination, on the basis of which the questions are developed are made available during the lecture.

Skills acquired during the laboratory classes are verified on the basis of a credit colloquium consisting of 20 questions and checking the practical implementation of the traffic planning problem. The credit threshold: 50% of points.

Programme content

Lecture:

- 1 System scripts in bash/python, cron, bashrc, services
- Programming the Discovery chip from Linux
3. udev rules, fixed names for USB devices, low latency for USB communication
4. ROS node for communication in USB and publishing data
5. concurrent processing in C++ (threads, processes)
- 6 CUDA (performing operations on a graphic card)
- 7 Tensorflow + ROS (start up the network detecting objects in ROS)
- 8 Remote Master (ROS on multiple computers)
- 9 ArUco object detection for calibration
- 10 ROS bags (collection of data from camera for calibration)
- 11 TFs in ROS (reading transformations from previously saved ROS-bags)
- 12 Calibration of easyHandEye cameras (on ROS-bags)

Laboratory:

- 1 System scripts in bash/python, cron, bashrc, services
- Programming the Discovery chip from Linux
3. udev rules, fixed names for USB devices, low latency for USB communication
4. ROS node for communication in USB and publishing data
5. concurrent processing in C++ (threads, processes)
- 6 CUDA (performing operations on a graphic card)
- 7 Tensorflow + ROS (start up the network detecting objects in ROS)
- 8 Remote Master (ROS on multiple computers)
- 9 ArUco object detection for calibration
- 10 ROS bags (collection of data from camera for calibration)
- 11 TFs in ROS (reading transformations from previously saved ROS-bags)
- 12 Calibration of easyHandEye cameras (on ROS-bags)

Course topics

none

Teaching methods

1. lecture: multimedia presentation, illustrated with examples given on the board.
2. laboratory exercises: instructions carried out on computers and robots available in the laboratory

Bibliography

Basic

Mark Mitchell, Jeffrey Oldham, Alex Samuel, Advanced Linux Programming, New Riders Publishing
Robot Operating System (ROS), Springer 2016

Additional

M. Galewski, STM32. Aplikacje i ćwiczenia w języku C, Wydawnictwo BTC, Legionowo 2011

Breakdown of average student's workload

	Hours	ECTS
Total workload	100	4,00
Classes requiring direct contact with the teacher	40	1,50
Student's own work (literature studies, preparation for laboratory classes/ tutorials, preparation for tests/exam, project preparation)	60	2,50