T-411-MECH **MECHATRONICS I** **6 ECTS**

Year of study: 3rd year BSc / 1st year MSc.
Semester: Fall.
Level of course: 3. First cycle, advanced / 4. Second cycle, introductory.
Type of course: Core for BSc Mechatronics Engineering, elective for other programs.
Mandatory Prerequisites: Programming in C++ (T-208-FOR2) or Programming (T-111-PROG).
Recommended prerequisites: Electric circuits (T-306-RAS1); Statics and Mechanics of Materials (T-106-BURD); Computer architecture (T-107-TOLH, may be taken in the same semester).
Schedule: Runs for 12 weeks - 6 teaching hours a week. A combination of lectures and labs.
Supervisor: Joseph Timothy Foley.
Lecturer: Joseph Timothy Foley.

Learning outcome:

Knowledge:

At the end of the class, students should have good knowledge of:

- Data encoding formats and binary arithmetic
- Different electrical, mechanical, and software components
- Commonly used sensors and actuators in smart devices
- Concepts of modularity, independence, information, and robustness

Skills:

At the end of the class, students should know how to:

- Operate oscilloscopes, multimeters, soldering irons, and benchtop power-supplies
- Create schematics, layout PCB boards, and solder components to build working devices
- Program a microcontroller to read sensors and control actuators. For example: An Arduino with C++ or a Raspberry Pi with python
- Apply the Axiomatic Design methodology to design modular and robust mechatronic systems.
- Understand digital and analog communication interfaces such as wireless networking, ethernet, and Internet of Things
- Understand actuator and electronics specification sheets
- Measure the accuracy, repeatibility, and resolution of a sensor
- Record communication and data into a design notebook properly for international-quality research
- Write lab reports and conference papers using LaTeX and Overleaf

Competences:

At the end of the class, students will be able to:

- Develop an application using a linux single-board-computer and real-time microcontroller
- Design, build, and test advanced circuits with active elements
- Choose the best components for a design
- Improve upon existing mechatronic devices
- Build actuator or control systems with feedback
- Debug electronic, software, and mechanical issues efficiently
- Write a conference-quality paper for a final report of a project

Content:

All course descriptions may be subject to change. Updated information on the course schedule, reading material, teaching and learning activities, and assessment methods will be introduced in the learning management system Canvas at the beginning of the semester.

This is an introduction to Mechatronics, the technique of interfacing software, electronics, and mechanical components. We will be utilizing the Raspberry Pi single-board linux microcontroller and Arduino as our focus. Students will have pay a fee for their personal lab kit which includes some shared parts for team-based labs. The chosen textbook is required for the course and critical to completing the Lab assignments.

We will begin with an introduction to linux and software engineering. This includes C++, python, and Subversion (for collaboration). We will then shift to electronics design, implementation, and testing. We will cover both analog and digital electronics with a focus on interfacing to sensors and actuators. Students will be designing and building PCB boards using Altium to integrate the electronics being developed. Students will choose a final mechatronics team project to be presented at the end of the semester. This project should involve manufacturing mechanical elements and interfacing them with the microcontrollers to demonstrate their mastery of the subject. Students will be spending a good deal of time in the Electronics Lab in V207 and Machine Shop building projects. This means that each student should have gone through safety training in the labs as a prerequisite. If you are an exchange student, contact the Teaching Assistants about setting up a safety training session so you can use the facilities.

Textbook: *Exploring Raspberry Pi: Interfacing to the Real World with Embedded Linux* by Derek Molloy, published by John Wiley and Sons in 2016.

Teaching and learning activities: Communication and rigor are critical to proper mechatronic design. Students will be shown how to use a research notebook and expected to keep it up to date as part of their grade. Proper citation of included internet and written material must be performed. Each subject will consist of lectures and related labs or projects. Significant student participation and interaction in lecture discussions is expected.

Many assignments are to be done in teams. Collaboration on individual assignments is expected, but each student must do their own writeup (no copying). Document assignments will use LaTeX templates that will be provided.

Students are expected to make use of the Machine shop and Electronics lab, taking appropriate International safety precautions where applicable. The instructor will give guidelines on these procedures and assist in execution.

Assessment methods: There is no final exam. Assignments will consist of a mixture of group lab assignments and individual competency tests. Students must be able to effectively communicate their ideas through written and oral methods. Students will each have a design notebook which must be used on a regular basis on topics relating to the class, team efforts, and analysis. The notebook will graded periodically.

Students are required to keep at least a 67% attendance grade in order to complete the course. Students are also expected to assist in cleanup of the mechatronics lab at the end of the course in order to receive a grade.

Proper citation is a requirement in this class, without exemptions. All material from an outside source (ideas, text, pictures) must include a proper citation. IEEE is the preferred format. Failure to include citations will result in a 0 for the assignment and considered plagarism which will be reported to the academic office. Improperly cited material will be assessed a lesser penalty depending upon the assignment. You are explicitly given permission to use the RU logo on your reports and presentations without citation because you are enrolled at our university.

All course descriptions may be subject to change. Updated information on the course schedule, reading material, teaching and learning activities, and assessment methods will be introduced in the learning management system Canvas at the beginning of the semester.

Late work will be penalized according to the degree of lateness: 10% per day that work is late to a maximum of 5 points (for 5+ days late). Assigments may only be submitted up to one week late. Software with code that was checked in to SVN at the due date/time can be checked off at the next session at full credit. If any changes are made after that time, the late work penalty will be applied. For LaTeX documents and presentations, the CANVAS submission time will be used to assess lateness. Always check the assignment for the appropriate submission procedure.

Whenever possible, the evaluation sheet for a given assignment will be provided before the start of the assignment. Reports and papers may be resubmitted up to a week after they are returned for regrading. These grades will be averaged for the new grade(40%/60%).

Language of instruction: English.

All course descriptions may be subject to change. Updated information on the course schedule, reading material, teaching and learning activities, and assessment methods will be introduced in the learning management system Canvas at the beginning of the semester.