Project Acronym: MEDIS

Project Title: A Methodology for the Formation of Highly Qualified Engineers at Masters Level in the Design and Development of Advanced Industrial Informatics Systems

Contract Number: 544490-TEMPUS-1-2013-1-ES-TEMPUS-JPCR

Starting date: 01/12/2013 **Ending date:** 30/11/2016

Deliverable Number: 4.1

Title of the Deliverable: Training AIISM – Design of training course
Task/WP related to the Deliverable: teachin material

Type (Internal or Restricted or Public): Internal

Author(s): Radu Dobrin and Sasikumar Punnekkat

Partner(s) Contributing: UPV, UTSTUTT, UP, TU-Sofia

Contractual Date of Delivery to the CEC: 30/09/2014

Actual Date of Delivery to the CEC: 30/09/2014

Context

Project Co-ordinator

Company name :	Universitat Politecnica de Valencia (UPV)
Name of representative :	Houcine Hassan
Address :	Camino de Vera, s/n. 46022-Valencia (Spain)
Phone number :	+34 96 387 7578
Fax number :	
E-mail :	husein@upv.es
Project WEB site address :	http://medis.upv.es/

WP 4	Design of the AIISM-PBL methodology
WPLeader	Mälardalen Univeristy (MDU)
Task 4.1	Training AIISM – Design of training course
Task Leader	MDU
Dependencies	UPV, MDU, TUSofia, USTUTT, UP
Starting date	01/03/2014
Release date	

Author(s)	Radu Dobrin and Sasikumar Punnekkat
Contributor(s)	
Reviewers	

History

Version	Date	Author	Comments
0.1	01/02/2016	Radu Dobrin and Sasikumar Punnekkat	Initial draft


MÄLARDALEN UNIVERSITY
A GREAT STUDY DESTINATION
IN THE HEART OF SWEDEN
medis
MÄLARDALEN UNIVERSITY SWEDEN

Course module: mobile monitoring and control of embedded systems

Challenges in remote monitoring and control		

- Programming mobile monitoring and control applications
- Security of monitoring and control applications
- Reliability of monitoring and control applications

Week	Type	Topic
1 – Introduction		
1	Lecture	Fundamentals of Remote Monitoring and Control
1	Seminar	Research of Remote Monitoring and Control
1	Laboratory	Lab introduction - Liquid tank system I
1	Miniproject	Presentation of project goals
2	Lecture	Lab introduction - Liquid tank system
2	Seminar	Research of mobile communication
2	Laboratory	Lab introduction - Liquid tank system II
2	Miniproject	Control systems and mobile devices
2 – IDE		
3	Lecture	Dedicated IDE's for IOS and Android
3	Seminar	IDE for mobile devices
3	Laboratory	IDE introduction - instalation and usability
3	Miniproject	Desgin and structuring of control application
4	Lecture	Cross-platform developement tools (Titanium, PhoneGap, etc)
4	Seminar	State of the Art on development tools
4	Lab	Basic app development !
4	Miniproject	Testing of mobile apps and I/O address mapping
3 – Basic App Development		
5	Lecture	Inter-device communication 1
5	Seminar	Research inter-device communication
5	Lab	Basic app development II
5	Miniproject	Implement basic control logic
6	Lecture	Inter-device communication 1
6	Seminar	Research synchronisation in distributed systems.
6	Lab	GUI development and implementaion
6	Miniproject	Implement complex control logic
4 – Graphical User Interface (GUI)		
7	Lecture	Grapghical User Interface (GUI) I
7	Seminar	Structure of code for GUI I
7	Lab	GUI development and implementation I
7	Miniproject	Add control logic to GUI I
8	Lecture	Grapghical User Interface (GUI) II
8	Seminar	Structure of code for GUI II
8	Lab	GUI development and implementation II
8	Miniproject	Add control logic to GUI II
5 – Security		
9	Lecture	Security in mobile communication
9	Seminar	Research application areas of secure wireless communication
9	Lab	Access controll and synchronization mechanisms I
9	Miniproject	Build a library of functions to secure access
10	Lecture	Security and control
10	Seminar	Research on secure control systems
10	Lab	Access controll and synchronization mechanisms II
10	Miniproject	Secure sending and receiving of messages.
6 – Reliability		
11	Lecture	Reliability in mobilie communication
11	Seminar	Research methods of reliability
11	Lab	Implement reliable communication
11	Miniproject	Simple distributed reliable control application
12	Lecture	Fault tolerance
12	Seminar	Research one fault tolerance for mobile devices
12	Lab	Implement fault tolerant communication
12	Miniproject	Add synchronous data transfer to distributed application.
13	Lecture	Testing of reliable mobile applications
13	Seminar	Research on testing of mobile communcation
13	Laboratory	Testing approaches for mobile communication
13	Miniproject	Determine latency of trafic in miniproject.
7 – Research findings		
14	Lecture	Research on dependable mobile communication 1

Course outline

7 chapters

Chapter 1: Fundamentals of remote monitoring and control
- L1: introduction to embedded systems and mobile devices
- L2: introduction to the water tank controller

Chapter 2: Integrated Develomnet Environment
- L3: intro to mobile applications in Android, iOS, BlackBerry and Windows mobile
- L4: intro to cross platform development (PhoneGap, Appcelerator, Titanium and Xamarin)

Chapter 3: Basic app development
- L5: inter device communication for Android
- L6: inter device communication for iOS

Chapter 4: GUI
- L7: Intro to GUI and basic standards
- L8: GUI for Android and iOS

Chapter 5: Security
- L9: Intro to security in embedded systems
- L10: Ways to achieve security (e.g. cryptography, vpn, etc)

Chapter 6: Reliability
- L11: Reliability in the context of dependability
- L12: Means to achieve reliability: fault tolerance
- L13: Software testing

Chapter 7: Research in remote monitoring and control
- L14 Seminar 1
- L15 Seminar 2

Course outline

7 chapters

Chapter 1: Fundamentals of remote monitoring and control

- Seminar: the working of water tank controller
- Miniproject: report - basic principles in control theory

Chapter 2: Integrated Develomnet Environment

- Seminar: the working of water tank controller
- Miniproject: report - basic principles in control theory

Chapter 3: Basic app development

- Seminar: discuss steps in creating P2P connections by using wifi and bluetooth
- Miniproject: report – compare wireless connectivity in Android and iOS

Chapter 4: GUI

- Seminar: discuss how the application starts a new activity discuss how the application starts a new activity
- Miniproject: report - discuss Event Listeners and Event Handlers interfaces.

Chapter 5: Security

- Seminar: discuss seminar articles in the field, as well as the results of the labs
- Miniproject: 4 page report on how they can apply security concepts for embedded systems

Chapter 6: Reliability

- Seminar: the working of water tank controller
- Miniproject: report – on selected articles

Chapter 7: Research in remote monitoring and control

- L14 Seminar 1
- L15 Seminar 2

Outline

- Introduction
- Learning Activities
- Lectures
- Labs
- Seminars
- Mini-Projects
- Scheduling
- Human and Material Resources
- Evaluation
- Conlusions

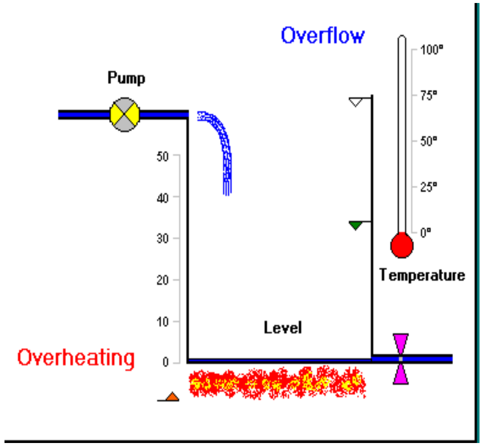
Introduction

Introduction:

- To detail the structure of the Advanced Industrial Informatics Specialization Modules (AIISM)
- Learn how to organize the AIISM methodology.
- The AIISM course uses a PBL (Problem Based Learning) methodology to instruct the design and implementation of II systems to control industrial processes
- The AIISM methodology is designed based on previous experiences of the EU universities on PBL and active learning techniques
- The purpose is to create a working environment for the students similar to the real environment in companies:
  - To guarantee fundamental knowledge of AIISM as basis for the development of further objectives
  - To accustom students work in teams when solving industrial problems
  - To encourage students to use practical skills to improve their problem solving abilities
  - To develop the capacity to adapt to any new computer based systems, due to rapid advances in this area.

Introduction:

- Other engineering transversal skills are gained during the course:
  - Teamwork
  - Technical competencies
  - Oral presentation
  - Budget management
  - Report redaction
  - Etc.



- The AIISM is structured with different activities of progressive complexity to facilitate the teams to develop their projects along the course.

- The activities are developed during 5 hours/day, one day of the week through a PBL methodology, using as a case study the example of the control of the liquids tank:

Learning activities

Learning activities description:

•Students have to apply the knowledge acquired from the lectures and the laboratory practices.

•The proposed learning activities are the following:

- **Lecture and problems**: lecturer presents main ideas of lecture contents and proposes some application problems which student solves individually (e. g., 1 h).
- **Laboratory session**: To implement (1 h 15'') a practical problem previously presented during lecture. Students work by teams of two students.
- **Seminars**: a panel discussion with student teams (4 students) lasting 45 minutes is proposed, consisting generally of solving a problem by means of PBL.
- **Mini-project**: dedicated to planning, design and development of the control system of the educational liquids tank. The mini-project is performed by teams of 4 students during 2 hours. Weekly, the mini-project is advanced progressively.

Lectures

Lectures: Introduction

- Lecture is the first step in the learning process for each of the topics in each course's module.
- The lecturer presents the main topics of the theme contents.
- It includes some application examples.

Lectures: Structure

During lecture, the teacher:

- PREVIOUSLY: plans the lecture session based on their professional experience.
- Presents the learning objectives of the subject.
- Contextualizes the subject within the module, the course and the career, based on the problems to be solved and the resources that will be proposed to solve these problems.
- Motivates the subject based on the importance of the problems coped by the subject.
- Lists and discusses the bibliographic resources that support the concepts that will be presented during the lecture and that students may use to deepen in their study.
- Presents the key concepts related to the subject, providing the needed details to properly understand them, and specifying the extra available resources and the learning process that students should follow to complete the knowledge's acquisition. Follows a logical order for the argumentation, so that students can acquire the knowledge progressively and uses illustrative examples to clarify the presented concepts.

Lectures: Structure (Cont.)

- Verifies that students correctly understand the presented concepts and adapts the speech if needed, allowing students to ask questions about the concepts that are not clear enough and observing the students' answers to the control questions.
- Informs about the skills related to the lecture that will be evaluated.
- Tells students about the practical application of the presented concepts that will be performed later during the seminars, laboratories and mini-project, and which are related to the lecture, encouraging students to study the issue with sufficient interest.
- After the lecture, keeps open communication channels with students, so they can make consultations before next lecture if necessary.
- LATER: should analyze the lecture session to improve their professional skills.

Lectures: Structure (Cont.)

Lectures: Structure (Cont.)

During lecture, the student:

- PREVIOUSLY: has studied the recommended previous readings.
- Is receptive to the teacher's presentation and is proactive.
- Takes notes to conceptualize what is being exposed to facilitate its further study.
- Asks for concepts' clarification if necessary.
- Answers the questions the teacher addresses to the audience.
- LATER: should follow the learning method proposed by the teacher

Lectures: Goals

- After the lecture the students will know what is the problem and its importance, how the problem is described, which available alternatives would solve it and the criteria to select an appropriate solving alternative even though they will not have yet the skills to apply the proposed methods.

- They will have a studying plan that will be organized in the form of a set of labs, seminars and mini-project.

Lectures: Prerequisites

- Previously to the lecture the student must be informed about the prerequisites to successfully follow the lecture.

- If the prerequisites aren't satisfied a previous to the lecture formation plan should be supplied.

Lectures: Example

“Reliability in remote monitoring and control of embedded systems”

This lecture will be on reliability of embedded systems and usage of fault tolerance and testing for dependable systems design. It will also cover the synchronization issues that can occur with multiple accesses during remote monitoring and control of these systems.

- Goal:

  To give the students an introduction to dependability in general, and reliability in particular for remote monitoring of embedded systems

Lectures: Example (Cont.)

- Outline
  - Introduction to dependability
  - Introduction to reliability
  - Examples
  - Reliability issues
  - Means to achieve reliability

- Bibliography

  - N. G. Leveson, “High-pressure steam engines and computer software,” in Proceedings of the 14th International Conference on Software Engineering, 1992, http://sunnyday.mit.edu/steam.pdf
  - What really happened on Mars? http://research.microsoft.com/en-us/um/people/mbj/Mars_Pathfinder/Mars_Pathfinder.html
  - N. G. Leveson, The role of software in spacecraft accidents, http://sunnyday.mit.edu/papers/jsr.pdf

Lectures: Recommendations

- During the lecture all the conversations in the classroom should be public, addressed to all of the members.
- The sequence of the lecture should be planned and predictable.
- The dynamics of the lecture should be attractive to the audience with short and direct explanations.
- The lecture shouldn't explain all the details of the concepts of the subject, it should give enough hints to let students autonomously complete the knowledge acquisition.
- The students must have the chance to formulate questions.
- Behavior rules must be defined to let the lecture to be productive and the corrective mechanisms should be effective and proportional.
- Potential distractors of the audience attention should be avoided or minimized.

Labs

Labs

The lab is the first practical exercise that students take to acquire a basic set of skills related to the topic presented in the lecture. The exercises in the labs solve specific and well-defined problems; they are guided, fully documented, and in progressively increasing complexity. The labs provide students with a set of tools and skills that can be used to solve more open problems during the seminars.

Labs: Structure

During lab, the teacher:

- PREVIOUSLY: plans the lab session based on his/her professional experience.
- Presents the learning goals of the lab.
- Contextualizes the lab within the subject.
- Motivates the practical exercise to be performed during the lab based on the importance of the problem it addresses.
- Lists and comments the equipment, the material and documentation resources needed to perform the lab.
- Describes the correct utilization of the lab equipment and warns about potential material and personal damage due to inappropriate use.
- Answers students questions during the practical exercise.
- LATER: should analyze the lab session to improve their professional skills.

Labs: Structure (Cont.)

During lab, the student:

- PREVIOUSLY: has studied the lab documentation, and has attended the related lectures.
- Is receptive to the teacher's indications and is proactive.
- Takes notes to remember the indications.
- Asks for concepts' clarification if necessary.
- Works in teams of two students on the practical exercises of the lab.
- Answers the questions of the teacher related to the exercise.
- LATER: should review and document the results of the practical exercises and eventually performs the extra optional exercises.

Labs: Structure (Cont.)

During lab, the technical assistant:

- PREVIOUSLY: Sets the necessary equipment for the lab in each of the workbenches based on the teacher's requests and his/her professional experience.
- Helps solving problems that could arise related to the equipment, power supply, communications and software, making diagnosis about the safeness and correctness operation of the equipment and replacing damaged components.
- LATER: should analyze the lab session to improve their professional skills.

Labs: Goal

- After the lab the student should have acquired the skills handle to assess the reliability of the app controlled embedded system framework. These skills will be useful in the next related seminars and mini-project exercises.

Labs: Prerequisites

- The student should have attended the related lecture and have read the recommended further lectures and the lab guide.
- Working teams of two-three people should have been set.

Labs: Lab Example

"Reliability" Lab

In this lab analog the students will answer the question on how the previously developed app for control of embedded systems behaves upon faults

In particular they will:

- Develop mechanisms to introduce or simulate faults in the app
    - modify the app such that it sends wrong values for temperature and pressure randomly
- Develop mechanisms to introduce or simulate faults in the water tank controller
    - modify the water tank controller such that it drops received values for temperature and pressure randomly
- Draw a graph that plots the expected temperature and pressure of the water tank controller vs. the actual temperature and pressure for *30* simulations
- Write a report that details your conclusions and reflections

Labs: Lab Example (Cont.)

- Motivation

  The importance of the analog signals in the control of physical processes is presented.

- Contents

  The lab documentation starts with and introduction, the general problem specification, the necessary hardware equipment – DAC and temperature sensor, a description of the transfer function of the sensor and an introduction to the DAC's library API's. It follows with a set of exercises.

Labs: Recommendations

- The lab should be fully documented.
- The lab guide should start with an introduction that remarks to the concepts of the lecture that are going to be applied in the practical exercise, the goals definition and a list of the material and documental resources that will be needed.
- The guide should continue with a definition of the practical activities in the following phases of progressive and increasing of complexity: introductory, reinforcement, advanced and optional.
- The teacher should supervise the practical exercises of the students, answering their questions, guiding them and providing enough hints to let the students find solutions by them shelf.
- Recurrent errors and problems, and interesting student's designs during the lab should be shared with the whole group. Remote desktop sessions of the workbench computer screen could be presented on the slide projector.

Seminars

Seminars

During the seminars the students must solve problems on the topic of the lecture. They have already exercised on related tools and procedures in the previous laboratories.

The type of the activities in the seminar has to do with the application of those tools and procedures to design solutions to specific problems. These problems are manageable parts of more complex problems. The activities involve information searching over the Internet, designs and calculations.

Students work in groups of 4 people, but they must explain and share their experiences with the whole group during the seminar meetings.

In the seminars of the course different problems and sub-problems related to the design and programming of physical processes controllers are analyzed from the perspective of the personal computer control platform studied in this modules.

Seminars: Structure

- The seminar starts with a teacher speech remembering about the key concepts presented during the lecture, and the proposition of several related design problems.
- The problems are decomposed in smaller parts, and the connections between the different sub-problems are discussed.
- Some of the problems are proposed to the whole group to start a debate, meanwhile other problems are proposed to be selected and solved by the different working teams during the seminar.
- After the investigation and design activities developed by each of the working teams the spokespeople share the obtained conclusions with the whole group and a second debate starts.
- The seminar ends with a resume by the teacher and the homework definition for each of the working teams that will prepare the next seminar session.

Seminars: Goals

- To practice the concepts presented in the related lectures and verify that these concepts have been assimilated correctly.
- To relate the previous concepts with other technical concepts that usually are studied in different subjects, in a contextualized way.
- Acquire team-working skills.
- Acquire documentation and presentation skills.
- Acquire critical searching of information skills.

Seminars: Prerequisites

- The student should have attended the related lecture and have read the further recommended readings.

- Working teams of four people should have been set. A balanced team is recommended with a similar level of initial knowledge for each of its members. One of the members' team will act as spokesperson.

Seminars: Example

- The students are expected to
    - Read the following articles and write a short summary along with your reflections.
    - Discuss the articles in class.

N. G. Leveson, “High-pressure steam engines and computer software,” in Proceedings of the 14th International Conference on Software Engineering, 1992, http://sunnyday.mit.edu/steam.pdf

What really happened on Mars? http://research.microsoft.com/en-us/um/people/mbj/Mars_Pathfinder/Mars_Pathfinder.html

N. G. Leveson, The role of software in spacecraft accidents, http://sunnyday.mit.edu/papers/jsr.pdf

Mini-project

Mini-project

During the mini-project students apply the knowledge and skills that they have acquired in the lectures, labs and seminars to develop in an integral way the controller of a physical process.

The problem of the mini-project is the highest complexity problem in the course.

The working teams in the mini-project are the same as in the seminars.

The designs developed by the teams during the seminars are used as components of the mini-project's problem's solution.

The teams can use in their own mini-projects, seminar designs that other teams have shared.

Mini-project: Goals

- After the mini-project, the students should be able to integrate the tools and protocols practiced during the course to develop simple complexity and medium sized applications to control physical process.

- They should be able to document and present the mini-project process and outcome.

Mini-project: Prerequisites

- The students should have attended the lectures and completed the practical exercises of the labs and seminars that are related to the module of the mini-project that is going to be developed.

Mini-project: Example

The students are expected to

- Submit a report summarizing the articles and the discussions during the seminar. □
- The report should also include your reflections on the reliability aspects of mobile applications that control embedded systems.

Scheduling

Week	Type	Topic
1 – Introduction		
1	Lecture	Fundamentals of Remote Monitoring and Control
1	Seminar	Research of Remote Monitoring and Control
1	Laboratory	Lab introduction - Liquid tank system I
1	Miniproject	Presentation of project goals
2	Lecture	Lab introduction - Liquid tank system
2	Seminar	Research of mobile communication
2	Laboratory	Lab introduction - Liquid tank system II
2	Miniproject	Control systems and mobile devices
2 – IDE		
3	Lecture	Dedicated IDE's for IOS and Android
3	Seminar	IDE for mobile devices
3	Laboratory	IDE introduction - instalation and usability
3	Miniproject	Desgin and structuring of control application
4	Lecture	Cross-platform developement tools (Titanium, PhoneGap, etc)
4	Seminar	State of the Art on development tools
4	Lab	Basic app development !
4	Miniproject	Testing of mobile apps and I/O address mapping
3 – Basic App Development		
5	Lecture	Inter-device communication 1
5	Seminar	Research inter-device communication
5	Lab	Basic app development II
5	Miniproject	Implement basic control logic
6	Lecture	Inter-device communication 1
6	Seminar	Research synchronisation in distributed systems.
6	Lab	GUI development and implementaion
6	Miniproject	Implement complex control logic
4 – Graphical User Interface (GUI)		
7	Lecture	Grapghical User Interface (GUI) I
7	Seminar	Structure of code for GUI I
7	Lab	GUI development and implementation I
7	Miniproject	Add control logic to GUI I
8	Lecture	Grapghical User Interface (GUI) II
8	Seminar	Structure of code for GUI II
8	Lab	GUI development and implementation II
8	Miniproject	Add control logic to GUI II
5 – Security		
9	Lecture	Security in mobile communication
9	Seminar	Research application areas of secure wireless communication
9	Lab	Access controll and synchronization mechanisms I
9	Miniproject	Build a library of functions to secure access
10	Lecture	Security and control
10	Seminar	Research on secure control systems
10	Lab	Access controll and synchronization mechanisms II
10	Miniproject	Secure sending and receiving of messages.
6 – Reliability		
11	Lecture	Reliability in mobilie communication
11	Seminar	Research methods of reliability
11	Lab	Implement reliable communication
11	Miniproject	Simple distributed reliable control application
12	Lecture	Fault tolerance
12	Seminar	Research one fault tolerance for mobile devices
12	Lab	Implement fault tolerant communication
12	Miniproject	Add synchronous data transfer to distributed application.
13	Lecture	Testing of reliable mobile applications
13	Seminar	Research on testing of mobile communcation
13	Laboratory	Testing approaches for mobile communication
13	Miniproject	Determine latency of trafic in miniproject.
7 – Research findings		
14	Lecture	Research on dependable mobile communication 1

Human
and
Material Resources

Human and Material Resources: Introduction

This methodology requires classroom-laboratories with general and pertinent equipment: industrial computers, mobile devices, data acquisition cards, microcomputers, prototypes, virtual labs, simulators.

Open source software is used for programming the applications, which will allow reducing the cost of the project.

This deliverable describes the resources required for learning activities and the staff required for supporting this methodology.

Human and Material Resources: The classroom-laboratory

The methodology proposed for this project can be enhanced using a single fully equipped classroom-laboratory.

These classroom-laboratories are interesting in order to avoid differentiating between theory, problems, and the laboratory practice and providing flexibility in the session development.

Each type of learning activity has different requirements as explained in the following subsections.

Human and Material Resources: The classroom-laboratory

Lectures

To present the main ideas of the contents we require the following typical classroom equipment:

- Board.
- Professor personal computer.
- Digital projector.
- Office applications.
- Internet connection.

To enhance the learning experience, the following extra equipment is recommended

- Samples of real devices (sensors, actuators, etc.).
- Mobile devices (tablets/smartphones) provided by the institution or by the students.
- e-learning materials

Human and Material Resources: The classroom-laboratory

Seminars and problems

Seminars require the same equipment than the lectures because the main purpose is to solve problems and discuss possible solutions:

- Student personal computer or mobile device.
- Collaborative software environments.
- Office applications.
- Internet connection.
- Development software.

To enhance the learning experience, it is recommended the following extra equipment

- Samples of real devices (sensors, actuators, etc.).
- Mobile devices (tablets/smartphones) provided by the institution or by the students.

For example, for code development, it is possible to use code sharing tools such as http://pastebin.com service in order to implement this idea.

Human and Material Resources: The classroom-laboratory

Laboratory

The resources required for the “remote monitoring of embedded sysems” module are:

- Student personal computer.
- Tablet
- Scale models, for example a “liquids tank” scale model.
- Simulators.
- Internet access.

Human and Material Resources: The classroom-laboratory

Miniprojects

This activity is dedicated to the planning, design and development of a framework for remote monitoring of embedded systems

Besides the equipment already mentioned in the previous section, for the miniproject, the following additional equipment will be required:

- Student personal computer or mobile device.
- Collaborative software environments.
- Office applications.
- Internet connection.
- Development software.

To enhance the hands on experience, it is recommended the following extra equipment

- Samples of real devices (sensors, actuators, etc.).
- Mobile devices (tablets/smartphones) provided by the institution or by the students.

Human and Material Resources: Human resources

Teachers

Regarding teachers, the assistance consists of resolving questions related to the teaching structure and the synchronization of learning sessions (lectures, labs, seminars) as well as questions of the specific contents of the course.

Teachers will follow a course about the implementation of the PBL methodology, the organization of the different learning units, the evaluation system for the students.

Part of the training course will deal with how to teach a class applying the proposed methodology to a small group of students. This is a pilot course for testing the proposals.

Human and Material Resources: Human Resources

Technicians

Laboratory technician is required to set-up all the hardware and software tools. Technicians will obtain support on aspects related to the installation and configuration of the software and hardware used to develop the laboratories and miniprojects.

Regarding the technicians, the training course shows how to use the necessary development tools (hardware and software) and their installation and configuration for the laboratory and mini-project activities.

Human and Material Resources: Human Resources

Administrative

The administrative staff will have support to help them translate the EU evaluation marks to the PC evaluation system as well as regarding the transference credits system.

The training of the administrative staff explains the use of the ECTS credits and its transference to the PC credit system as well as the grades equivalence among EU and the different PC systems.

Evaluation

Evaluation: Introduction

- A very important aspect of the learning process as it will allow us to determine the level of assimilation of knowledge and skills by students.
- Not only be focused on the technical knowledge of the subject but should also include assessment of those skills and competencies that students must acquire.
- Pay attention to how they have developed cognitive skills (analysis, synthesis, application, evaluation, and critique)
- Action skills (organizing time, resources, coordination, negotiating, tolerating)
- The assessment must take into account how students are acquiring the knowledge, skills and competencies and ensure that those who pass the course have appropriate capabilities
- Problem solving related to real world problems is motivating for students as they see direct application and better assimilate concepts
- Students identify the problem, research on how to solve it applying concepts and principles. If they work in teams, develop communication skills and collaborative work, developing analytic skills
- During the evaluation process large amount of information will be collected

Evaluation: Formation of Teams

- Teams of 2 or 3 students.
- Groups should remain invariant throughout the course.
- Students who form a team have a similar level of knowledge.
- To set the level of students, the first day of class an objective type test will be performed, with around 25 questions, each one with 3 possible answers and only one answer will be correct. In this way we can know the real starting level of the students. This test should contain questions on mobile app development for control of embedded sytems.
- Questions should be aimed to collect general aspects of the concepts required as prerequisites to begin to pursue the matter. For example:
    - Basic Engineering concepts
    - Programming (Software)
    - Reliability/security issues

Evaluation: Student evaluation methodology

Level One: Attitude (Student engagement)

- Motivation that students have within the course.
- keep an ongoing dialogue with different student groups of the course, this should be maintained by the lecturer throughout the entire course.
- Special attention to students deliver activities in the time and manner agreed as it is a clear indicator of student motivation
- Deadline for delivery of the different activities that students set and meet deadlines. This should be something to evaluate.
- Tis level of evaluation should be a part of the final mark. This part can be 10% of the final overall rating.
- Identify those students with a special motivation for the subject because their attitude is above average
- It is important that students meet requirements of work deadlines in the subject. This will facilitate the professional future of the graduates in order to work in environments with strict deadlines.

Evaluation: Student evaluation methodology

Level Two: Learning

We determine the acquisition of knowledge and skills that students have acquired throughout the course.

Problems:

•Lecturer presents main ideas of lecture contents and proposes some application problems which student solves individually.

•The teachers throughout the course individually assess how the student has solved different problems.

•For example, through the problems we can assess how students interpret a transfer function of a sensor/actuator. How to interpret the voltage as a physical quantity. The analytical representation by an equation. We can see if the student can interpret the current temperature if the input value is X volts, etc.

•For the evaluation of the problem must take into account: Approach resolution procedures, steps followed in the resolution, final result, method, clarity of presentation and approach, inclusion of units of measure, focus on the important issues facing superfluous. In the event that the final result is not correct,

Evaluation: Student evaluation methodology

Level Two: Learning

Laboratory:

- A practical problem previously presented during lecture.
- Students work by teams of two/three students.
- During the lab sessions students will show the teacher how they are solving the proposed activities
- Teacher will make questions about how is the resolution of the activity.
- At the end of each lab session the teacher will rate each group based on the work done and the objectives achieved.
- For evaluation can take into account:
    - Introduction phase: will reflect 20% of the grade.
    - Reinforcement Phase: will reflect 40% of the grade.
    - Advanced stage: Will represent a 40% of the grade.

Evaluation: Student evaluation methodology

Level Two: Learning

Seminars:

- A panel discussion with student teams (around six students) is proposed, consisting generally of solving a problem by means of PBL.
- The teachers will meet with each of the groups who will present how they have raised the issue, what options for the resolution are viable and which ones have been taken.
- It is important that the teacher dialogue with all members of the group to identify how well attended and have acquired the relevant knowledge.
- Some aspects to be taken into account for the assessment:
  - Level of responsibility among group members.
  - Number of studied and analysis in terms of advantages and disadvantages of each possible solution solutions.
  - Quality technical report writing as to the work done
  - Defense of ideas and how to defend against constructive criticism
  - Interaction of knowledge of this area with other
  - Management of bibliographic sources
  - Extra work done with respect to the requested
  - Robustness of the proposed solution adopted
  - Etc.

Evaluation: Student evaluation methodology

Level Two: Learning

Mini-project:

- Dedicated to planning, design and development of the control system of a real problem design.
- The developed project will presented publicly to a jury composed of three lecturers.
- The jury assesses the following aspects including some transversal skills:
    - Report: Maximum score of this part is 25% of the total mark.
    - Oral presentation: The score of this part is 10% maximum. The team presents the work during maximum of 20 minutes.
    - Implementation: The score assigned to this part is 65% of total mark. After the oral presentation, the team shows the project application.

Evaluation: Student evaluation methodology

Level Two: Learning

Mini-project (Cont.):

In each of these sections in addition to the assessment of knowledge, the teacher should take into consideration and evaluate all the important skills and transversal skills for engineers:

- Cognitive skills: Analysis, synthesis, application, evaluation, critique, etc.
- Action skills: Organizing time, resources, coordination, negotiating, tolerating, etc.

The advantage of the methodology is that it allows including other skills that assessment allow an integrated formation of the student: *competition, working in teams, cooperation, oral presentations, budget management, report redaction, etc.*

The rating of these skills should be included in the appropriate rating to each of the evaluation issues associated at this level.

Evaluation: Student evaluation methodology

Level Three: Grading (outcome)

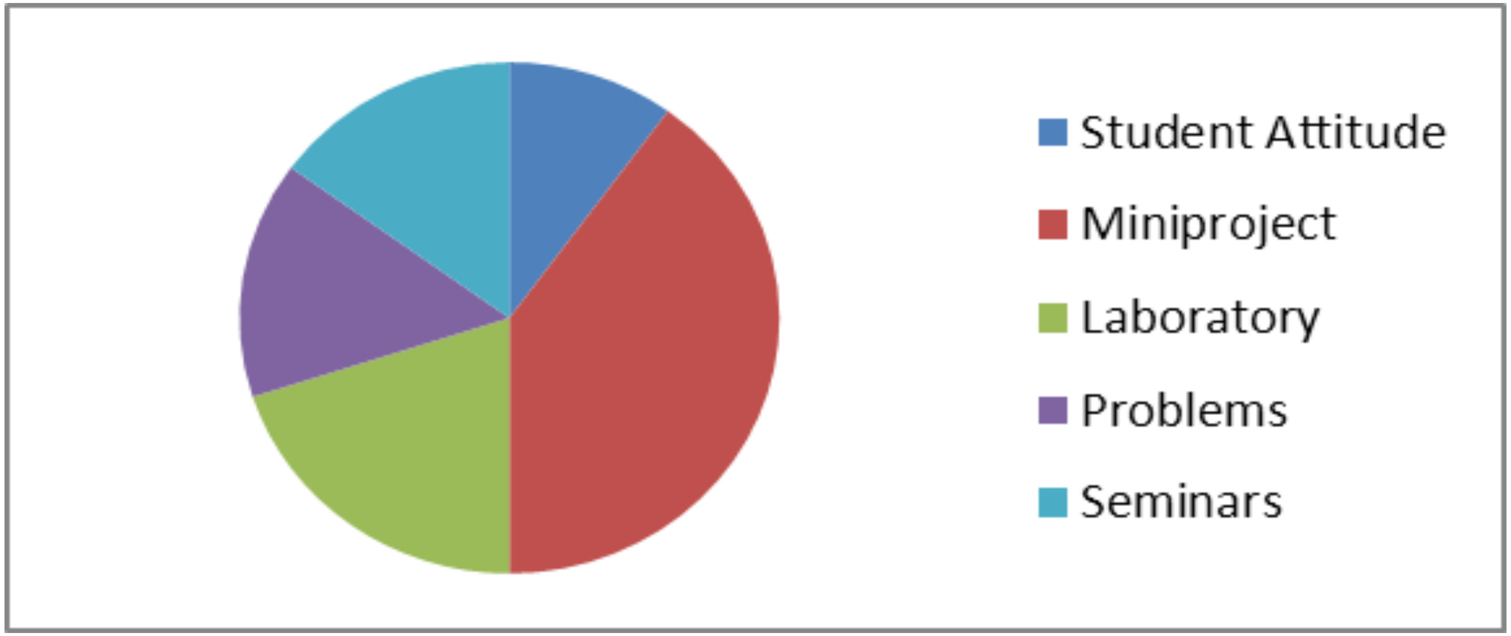
- We will collect all grades earned along the continuous assessment developed along the course
- Proceed to obtain the final grade for the course.
- We establish a just proportion between the different sections to ensure a fair rating for differentiating their individual acquisition of knowledge and skills against the student group work.
- The proposal will apply as follows:
    - The evaluation of the student attitude (A) a 10% of the final score.
    - The evaluation of the miniproject represents (MP) a 40% of the final score.
    - The evaluation of the Laboratory (L): 20% of the final score.
    - The evaluation of the Problems (P): 15% of the final score.
    - The evaluation of the Seminars (S): 15% of the final score.

Evaluation: Student evaluation methodology

Level Three: Grading (outcome) (Cont.)

For the calculation of the final grade (FG) can be followed as the following equation:

$$FG= A*0.1+MP*0.4+L*0.2+P*0.15+S*0.15$$



Evaluation: Assessment of the methodology system

- Evaluation of the system used for teaching this subject.
- Important to know the opinion of students and teachers involved to find out what has been done well and what parts could be improved
- In this sense the evaluation board system from two points of view:
    - Student point of view:
        - A survey among students to have information about the acceptance of the course.
        - Students can give their opinion at the end of the course and before obtaining their qualifications
        - Design a survey in a way that can be simple and easy to answer. For instance, can be made based on 6 questions with 5 possible answers ("A": Strongly Agree; "B": Agree; "C": Unsure; "D": Disagree; "E": Strongly Disagree) for each, rated from A to E marks.

Evaluation: Assessment of the methodology system (Cont.)

- The questions can be of the type:
  - Has the subject methodology facilitated your learning process?
  - Has every important concept of the subject been addressed in the miniproject?
  - Has the complexity level of every part of the subject been reasonable?
  - Has the activities promoted cooperation skills as in real work environments?
  - Have you felt motivated during the learning process?
  - Would you recommend taking this course to other students?

- Lecturer viewpoint:
  - The opinion of teachers is important to make an overall assessment of how the course has worked and what aspects should be improved.
  - Teachers should maintain an open dialogue throughout the course and at the end make the balance.
  - The aspects to be evaluated are for example:
    - Ratio of approved students
    - Quality of ratings
    - Amount of work done by teachers
    - Problems that have arisen and how they have been resolved
    - Possible updating of content,
    - Duplication and overlap with other subjects, etc.

Thanks!

MÄLARDALENS HÖGSKOLA
ESKILSTUNA VÄSTERÅS

UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

medis

Introduction

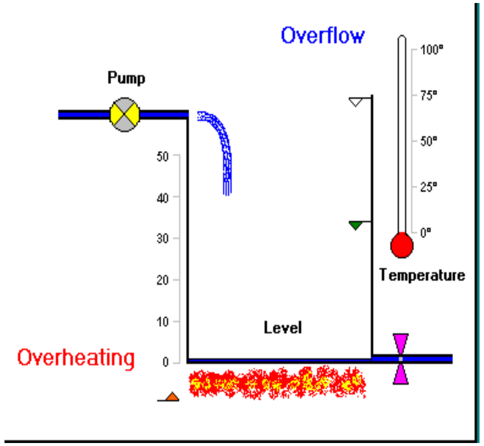
Introduction:

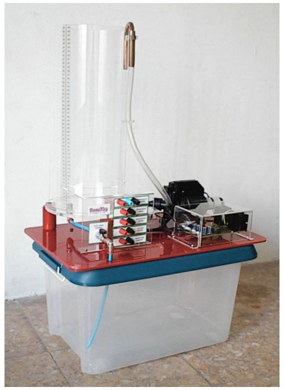
- To detail the structure of the Advanced Industrial Informatics Specialization Modules (AIISM)
- Learn how to organize the AIISM methodology.
- The AIISM course uses a PBL (Problem Based Learning) methodology to instruct the design and implementation of II systems to control industrial processes
- The AIISM methodology is designed based on previous experiences of the EU universities on PBL and active learning techniques
- The purpose is to create a working environment for the students similar to the real environment in companies:
    - To guarantee fundamental knowledge of AIISM as basis for the development of further objectives
    - To accustom students work in teams when solving industrial problems
    - To encourage students to use practical skills to improve their problem solving abilities
    - To develop the capacity to adapt to any new computer based systems, due to rapid advances in this area.

- Other engineering transversal skills are gained during the course:
  - Teamwork
  - Technical competencies
  - Oral presentation
  - Budget management
  - Report redaction
  - Etc.

- The AIISM is structured with different activities of progressive complexity to facilitate the teams to develop their projects along the course.

- The activities are developed during 5 hours/day, one day of the week through a PBL methodology, using as a case study the example of the control of the liquids tank:





Learning activities

Learning activities description:

- Students have to apply the knowledge acquired from the lectures and the laboratory practices.
- The proposed learning activities are the following:
  - **Lecture and problems**: lecturer presents main ideas of lecture contents and proposes some application problems which student solves individually (e. g., 1 h).
  - **Laboratory session**: To implement (1 h 15”) a practical problem previously presented during lecture. Students work by teams of two students.
  - **Seminars**: a panel discussion with student teams (4 students) lasting 45 minutes is proposed, consisting generally of solving a problem by means of PBL.
  - **Mini-project**: dedicated to planning, design and development of the control system of the educational liquids tank. The mini-project is performed by teams of 4 students during 2 hours. Weekly, the mini-project is advanced progressively.

Lectures

Lectures: Introduction

- Lecture is the first step in the learning process for each of the topics in each course’s module.
- The lecturer presents the main topics of the theme contents.
- It includes some application examples.

Lectures: Structure

During lecture, the teacher:

- PREVIOUSLY: plans the lecture session based on their professional experience.
- Presents the learning objectives of the subject.
- Contextualizes the subject within the module, the course and the career, based on the problems to be solved and the resources that will be proposed to solve these problems.
- Motivates the subject based on the importance of the problems coped by the subject.
- Lists and discusses the bibliographic resources that support the concepts that will be presented during the lecture and that students may use to deepen in their study.
- Presents the key concepts related to the subject, providing the needed details to properly understand them, and specifying the extra available resources and the learning process that students should follow to complete the knowledge's acquisition. Follows a logical order for the argumentation, so that students can acquire the knowledge progressively and uses illustrative examples to clarify the presented concepts.

Lectures: Structure (Cont.)

- Verifies that students correctly understand the presented concepts and adapts the speech if needed, allowing students to ask questions about the concepts that are not clear enough and observing the students' answers to the control questions.
- Informs about the skills related to the lecture that will be evaluated.
- Tells students about the practical application of the presented concepts that will be performed later during the seminars, laboratories and mini-project, and which are related to the lecture, encouraging students to study the issue with sufficient interest.
- After the lecture, keeps open communication channels with students, so they can make consultations before next lecture if necessary.
- LATER: should analyze the lecture session to improve their professional skills.

Lectures: Structure (Cont.)

During lecture, the student:

- PREVIOUSLY: has studied the recommended previous readings.
- Is receptive to the teacher's presentation and is proactive.
- Takes notes to conceptualize what is being exposed to facilitate its further study.
- Asks for concepts' clarification if necessary.
- Answers the questions the teacher addresses to the audience.
- LATER: should follow the learning method proposed by the teacher

Lectures: Goals

- After the lecture the students will know what is the problem and its importance, how the problem is described, which available alternatives would solve it and the criteria to select an appropriate solving alternative even though they will not have yet the skills to apply the proposed methods.
- They will have a studying plan that will be organized in the form of a set of labs, seminars and mini-project.

Lectures: Prerequisites

- Previously to the lecture the student must be informed about the prerequisites to successfully follow the lecture.
- If the prerequisites aren't satisfied a previous to the lecture formation plan should be supplied.

Lectures: Example

"Process Interface"

In this lecture the communication between the control program and the physical process to control is discussed.

- Goal:
  To learn methods to observe the current situation of the physical process under control and manipulate it, in order to make control decisions and apply them to influence the dynamic evolution of the process.

- Contextualization:
  The general concept of the process interface is particularized for the specific characteristics of the personal computer control platform. Hardware devices involved in the process interface are presented. The architecture and electronic technology aspects of these devices will be studied in other courses, their programming aspects will be studied in this course though.

Lectures: Example (Cont.)

- Motivation

  The process interface is presented as an essential module in the control program.
  The correct observation of the current situation is necessary to make accurate control decisions.
  Students already know how to communicate their computer programs with the real world; they have defined user interfaces in previous courses, where users were able to pass input data to the program and get the results using interfaces like keyboards and screens. The process interface they are going to develop in this course is an improving extension to that communication capability.

- Bibliography
  - References to transducer devices: sensors and motors.
  - References to analog-to-digital and digital-to-analog converters.
  - References to data acquisition cards.
  - References to APIs of data acquisition libraries.

Lectures: Example (Cont.)

- Concepts
  - The necessity of observation and manipulation of the physical process is presented.
  - The concepts of sensor and actuator and their transfer functions are presented.
  - The different types of signals: input and output, digital and analog, their codification and their interpretation as physical magnitudes are presented.
  - The available methods of signal acquisition are presented and their programming aspects and the response time requirements are discussed.
  - C/C++ programming concepts useful in the data acquisition programming: the use of bitwise operators and masks in the interpretation of the I/O lectures.

.

Lectures: Example (Cont.)

- Examples
  - The signal from the sensor of the current liquids level in the tank is used as an example of analog input signal.
  - The alarm signal from the overflowing sensor is used as an example of digital input signal.
  - The signal to the heater in the tank, which is manipulated on an ON/OFF basis, is used as an example of digital output signal.
  - The signal to the liquids input pump in the tank is used as an example of analog output signal.

- Control Questions and Recommended Further Reading
  - Question Example:
  - Do you know which external representations for integral numbers are allowed in C/C++ language? References to octal, decimal and hexadecimal representations in C/C++.
  - Do you know how a literal integral number in a C/C++ expression is internally represented in memory? References to internal data representations in C/C++.

Lectures: Recommendations

- During the lecture all the conversations in the classroom should be public, addressed to all of the members.
- The sequence of the lecture should be planned and predictable.
- The dynamics of the lecture should be attractive to the audience with short and direct explanations.
- The lecture shouldn't explain all the details of the concepts of the subject, it should give enough hints to let students autonomously complete the knowledge acquisition.
- The students must have the chance to formulate questions.
- Behavior rules must be defined to let the lecture to be productive and the corrective mechanisms should be effective and proportional.
- Potential distractors of the audience attention should be avoided or minimized.

Labs

Labs

The lab is the first practical exercise that students take to acquire a basic set of skills related to the topic presented in the lecture. The exercises in the labs solve specific and well-defined problems; they are guided, fully documented, and in progressively increasing complexity. The labs provide students with a set of tools and skills that can be used to solve more open problems during the seminars.

Labs: Structure

During lab, the teacher:

- PREVIOUSLY: plans the lab session based on his/her professional experience.
- Presents the learning goals of the lab.
- Contextualizes the lab within the subject.
- Motivates the practical exercise to be performed during the lab based on the importance of the problem it addresses.
- Lists and comments the equipment, the material and documentation resources needed to perform the lab.
- Describes the correct utilization of the lab equipment and warns about potential material and personal damage due to inappropriate use.
- Answers students questions during the practical exercise.
- LATER: should analyze the lab session to improve their professional skills.

Labs: Structure (Cont.)

During lab, the student:

- PREVIOUSLY: has studied the lab documentation, and has attended the related lectures.
- Is receptive to the teacher's indications and is proactive.
- Takes notes to remember the indications.
- Asks for concepts' clarification if necessary.
- Works in teams of two students on the practical exercises of the lab.
- Answers the questions of the teacher related to the exercise.
- LATER: should review and document the results of the practical exercises and eventually performs the extra optional exercises.

Labs: Structure (Cont.)

During lab, the technical assistant:

- PREVIOUSLY: Sets the necessary equipment for the lab in each of the workbenches based on the teacher's requests and his/her professional experience.
- Helps solving problems that could arise related to the equipment, power supply, communications and software, making diagnosis about the safeness and correctness operation of the equipment and replacing damaged components.
- LATER: should analyze the lab session to improve their professional skills.

Labs: Goal

- After the lab the student should have acquired the skills to develop a basic data acquisition system. These skills will be useful in the next related seminars and mini-project exercises.

Labs: Prerequisites

- The student should have attended the related lecture and have read the recommended further lectures and the lab guide.
- Working teams of two people should have been set.

Labs: Lab Example

"Process Interface" Subject - "Analog Input: Temperature Sensor" Lab

In this lab analog input signals are acquired using a Data Acquisition Card by National Instruments. The current temperature is observed.

- Goal

  To learn how to acquire analog input signals using API libraries of commercial DACs.

- Contextualization

  The general concept of the process interface is particularized for the specific characteristics of the personal computer control platform studied in this module of the course.

  Hardware devices involved in the process interface are presented. The architecture and electronic technology aspects of these devices will be studied in other courses, their programming aspects will be studied in this course though.

Labs: Lab Example (Cont.)

- Motivation

  The importance of the analog signals in the control of physical processes is presented.

- Contents

  The lab documentation starts with and introduction, the general problem specification, the necessary hardware equipment – DAC and temperature sensor, a description of the transfer function of the sensor and an introduction to the DAC's library API's. It follows with a set of exercises.

Labs: Lab Example (Cont.)

- List of exercises:

1. Introductory phase - basic and completely guided exercise
    - The process to program the DAC to acquire the signal of the temperature sensor and show the current temperature of the laboratory room on the computer screen is fully described.
2. Reinforcement phase
    - The student is requested to acquire a different analog signal, coming it from a different analog input port, and to interpret this new signal with a different transfer function. The student must develop the exercise without the previous guide. The physical signal is simulated using a signal generator in this case. The frequency of the signal acquisition is set constant.
3. Advanced phase
    - The student is requested to combine the code that was developed in the previous exercises with the developed code in a previous lab. The goal is to acquire and plot on the computer screen the temperature of the room and interpret a digital input signal as an overheating alarm sensor.
4. Optional phase
    - The student is requested to analyze the effects on the observed signal depending on the frequency of the input signal, which simulated with the generator, and the frequency of the signal acquisition loop.

Labs: Recommendations

- The lab should be fully documented.
- The lab guide should start with an introduction that remarks to the concepts of the lecture that are going to be applied in the practical exercise, the goals definition and a list of the material and documental resources that will be needed.
- The guide should continue with a definition of the practical activities in the following phases of progressive and increasing of complexity: introductory, reinforcement, advanced and optional.
- The first exercise in the introductory phase should be described step by step.
- The second exercise in the reinforcement should practice the same concepts and method than before but on a different set of problem data and without the help of the guide
- The third exercise in the advanced phase should practice the application of the previously acquired tools, protocols and skills to solve a small sized and small complexity application problem.
- A fourth exercise should be defined in an optional final phase to let advanced students to consider further technical questions related to the topic of the lab.
- The teacher should supervise the practical exercises of the students, answering their questions, guiding them and providing enough hints to let the students find solutions by them shelf.
- Recurrent errors and problems, and interesting student's designs during the lab should be shared with the whole group. Remote desktop sessions of the workbench computer screen could be presented on the slide projector.

Seminars

Seminars

During the seminars the students must solve problems on the topic of the lecture. They have already exercised on related tools and procedures in the previous laboratories.

The type of the activities in the seminar has to do with the application of those tools and procedures to design solutions to specific problems. These problems are manageable parts of more complex problems. The activities involve information searching over the Internet, designs and calculations.

Students work in groups of 4 people, but they must explain and share their experiences with the whole group during the seminar meetings.

In the seminars of the course different problems and sub-problems related to the design and programming of physical processes controllers are analyzed from the perspective of the personal computer control platform studied in this modules.

Seminars: Structure

- The seminar starts with a teacher speech remembering about the key concepts presented during the lecture, and the proposition of several related design problems.

- The problems are decomposed in smaller parts, and the connections between the different sub-problems are discussed.

- Some of the problems are proposed to the whole group to start a debate, meanwhile other problems are proposed to be selected and solved by the different working teams during the seminar.

- After the investigation and design activities developed by each of the working teams the spokespeople share the obtained conclusions with the whole group and a second debate starts.

- The seminar ends with a resume by the teacher and the homework definition for each of the working teams that will prepare the next seminar session.

Seminars: Goals

- To practice the concepts presented in the related lectures and verify that these concepts have been assimilated correctly.
- To relate the previous concepts with other technical concepts that usually are studied in different subjects, in a contextualized way.
- Acquire team-working skills.
- Acquire documentation and presentation skills.
- Acquire critical searching of information skills.

Seminars: Prerequisites

- The student should have attended the related lecture and have read the further recommended readings.

- Working teams of four people should have been set. A balanced team is recommended with a similar level of initial knowledge for each of its members. One of the members' team will act as spokesperson.

Seminars: Example

- Problem: Integration of the following C++ mini-project modules for the implementation of the liquids tank level and temperature regulators: Process Interface Module, Common Variables Module and Control Module.

- Sub-problems:
  - Simulation of the physical problem’s dynamics (level and temperature of the liquid in the tank) to validate the controller.
  - Simulation of the behavior of the physical process under control due to expected perturbations (aperture of the output valve, change of the ambient temperature and change of the input liquid temperature).
  - Observations of the continuous signals (level and temperature).
  - Alarm handling (overflowing and overheating digital sensors).
  - Acting (input liquid pump and heater).
  - Design of the automata (operating modes and operating phases in the liquid service).
  - Design of the regulators (level and temperature control).
  - Programming of the automata.
  - Programming of the regulators.
  - Coupling issues between controlled variables (level and temperature).
  - Validation of the controller.

Mini-project

Mini-project

During the mini-project students apply the knowledge and skills that they have acquired in the lectures, labs and seminars to develop in an integral way the controller of a physical process.

The problem of the mini-project is the highest complexity problem in the course.

The working teams in the mini-project are the same as in the seminars.

The designs developed by the teams during the seminars are used as components of the mini-project’s problem’s solution.

The teams can use in their own mini-projects, seminar designs that other teams have shared.

Mini-project: Goals

- After the mini-project, the students should be able to integrate the tools and protocols practiced during the course to develop simple complexity and medium sized applications to control physical process.

- They should be able to document and present the mini-project process and outcome.

Mini-project: Prerequisites

- The students should have attended the lectures and completed the practical exercises of the labs and seminars that are related to the module of the mini-project that is going to be developed.

Mini-project: Example

To develop of a complete controller of a liquids tank, integrating the following application modules:

- Common Variables Module.
- User Interface Module.
- Process Interface Module.
- Tasks Module.
- Control Module.

Scheduling

Scheduling

Week	Chapter	Type	Topic
	1	INTRODUCTION	
1	1	Lecture	Introduction to industrial informatics
1	1	Lab	Development environment - Programming the "Hello World"
1	1	Seminar	C programming (1) - Basic resources
1	1	Miniproject	Presentation of the problem to solve
2	1	Lecture	Structure and design of industrial informatics systems
2	1	Lab	Event oriented programming
2	1	Seminar	C programming (2) - Programming tools
2	1	Miniproject	Analysis of the project requirements
	2	COMPUTER	
3	2	Lecture	Computer architecture
3	2	Lab	Using libraries in C
3	2	Seminar	C programming (3) - Libraries
3	2	Miniproject	Project formal specification
	3	PROJECT PLANNING	
4	3	Lecture	Project management (1)
4	3	Lab	Tools for project management
4	3	Seminar	Discussing cases of project management systems
4	3	Miniproject	Project planning
	4	PROGRAMMING + DATA	
5	4	Lecture	Modular programming
5	4	Lab	Modular programming in C
5	4	Seminar	Modular programming resources
5	4	Miniproject	Modular decomposition of the program
6	4	Lecture	Data representation and sharing
6	4	Lab	Data sharing between C modules
6	4	Seminar	Chosing the appropiate data representation
6	4	Miniproject	Implementation of the shared data module
	5	PROCESS INTERFACE	
7	5	Lecture	Process interface (1) - Introduction and digital input
7	5	Lab	Digital input
7	5	Seminar	DAQ card (1) - Introduction and digital input
7	5	Miniproject	Implementation of the process interface module (1) - DI
8	5	Lecture	Process interface (2) - Digital output
8	5	Lab	Digital output
8	5	Seminar	DAQ card (2) - Digital output
8	5	Miniproject	Implementation of the process interface module (2) - DO
9	5	Lecture	Process interface (3) - Analog input and output
9	5	Lab	Analog input and output
9	5	Seminar	DAQ card (3) - Analog input and output
9	5	Miniproject	Implementation of the process interface module (3) - AIO
	6	USER INTERFACE	
10	6	Lecture	Graphical user interface (1) - Introduction
10	6	Lab	Programming GUI controls
10	6	Seminar	Graphical user interface for the industry (1) - Basic
10	6	Miniproject	Implementation of the user interface module (1) - Basic
11	6	Lecture	Graphical user interface (2) - Advanced resources
11	6	Lab	Programming a GUI for an industrial application
11	6	Seminar	Graphical user interface for the industry (2) - Advanced
11	6	Miniproject	Implementation of the user interface module (2) - Advanced
	7	TASKS	
12	7	Lecture	Task scheduling
12	7	Lab	Basic scheduler
12	7	Seminar	Scheduling strategies
12	7	Miniproject	Implementation of the task scheduler module
	8	REGULATION	
13	8	Lecture	Foundations and continuous control
13	8	Lab	Programming regulation strategies (1) cc
13	8	Seminar	Control strategies (1) cc
13	8	Miniproject	Implementation of the regulator module (1) cc
14	8	Lecture	Event-driven control
14	8	Lab	Programming regulation strategies (2) edc
14	8	Seminar	Control strategies (2) edc
14	8	Miniproject	Implementation of the regulator module (2) edc
	10	PROJECT (2) ENDING	
15	10	Lecture	Project documentation and presentation
15	10	Lab	Tools for project documentation
15	10	Seminar	Project documentation strategies
15	10	Miniproject	Documentation and presentation of the project

Human and Material Resources

Human and Material Resources: Introduction

This methodology requires classroom-laboratories with general and pertinent equipment: industrial computers, mobile devices, data acquisition cards, microcomputers, prototypes, virtual labs, simulators.

Open source software is used for programming the applications, which will allow reducing the cost of the project.

This deliverable describes the resources required for learning activities and the staff required for supporting this methodology.

Human and Material Resources: The classroom-laboratory

The methodology proposed for this project can be enhanced using a single fully equipped classroom-laboratory.

These classroom-laboratories are interesting in order to avoid differentiating between theory, problems, and the laboratory practice and providing flexibility in the session development.

Each type of learning activity has different requirements as explained in the following subsections.

Human and Material Resources: The classroom-laboratory

Lectures

To present the main ideas of the contents we require the following typical classroom equipment:

- Board.
- Professor personal computer.
- Digital projector.
- Office applications.
- Internet connection.

To enhance the learning experience, the following extra equipment is recommended

- Samples of real devices (sensors, actuators, etc.).
- Mobile devices (tablets/smartphones) provided by the institution or by the students.
- e-learning materials

Human and Material Resources: The classroom-laboratory

Seminars and problems

Seminars require the same equipment than the lectures because the main purpose is to solve problems and discuss possible solutions:

- Student personal computer or mobile device.
- Collaborative software environments.
- Office applications.
- Internet connection.
- Development software.

To enhance the learning experience, it is recommended the following extra equipment

- Samples of real devices (sensors, actuators, etc.).
- Mobile devices (tablets/smartphones) provided by the institution or by the students.

For example, for code development, it is possible to use code sharing tools such as http://pastebin.com service in order to implement this idea.

Human and Material Resources: The classroom-laboratory

Laboratory

The resources required for the “Industrial computers” module are:

- Student personal computer.
- Power source.
- Oscilloscope.
- Multimeter.
- Scale models, for example a “liquids tank” scale model.
- DAQ boards.
- Testing board.
- Qt development environment software.
- Simulators.
- Internet access.

Miniprojects

This activity is dedicated to the planning, design and development of an industrial control system.

Besides the equipment already mentioned in the previous section, for the miniproject, the following additional equipment will be required:

- Student personal computer or mobile device.
- Collaborative software environments.
- Office applications.
- Internet connection.
- Development software.

To enhance the hands on experience, it is recommended the following extra equipment

- Samples of real devices (sensors, actuators, etc.).
- Mobile devices (tablets/smartphones) provided by the institution or by the students.

Human and Material Resources: Human resources

Teachers

Regarding teachers, the assistance consists of resolving questions related to the teaching structure and the synchronization of learning sessions (lectures, labs, seminars) as well as questions of the specific contents of the course.

Teachers will follow a course about the implementation of the PBL methodology, the organization of the different learning units, the evaluation system for the students.

Part of the training course will deal with how to teach a class applying the proposed methodology to a small group of students. This is a pilot course for testing the proposals.

Human and Material Resources: Human resources

Technicians

Laboratory technician is required to set-up all the hardware and software tools. Technicians will obtain support on aspects related to the installation and configuration of the software and hardware used to develop the laboratories and miniprojects.

Regarding the technicians, the training course shows how to use the necessary development tools (hardware and software) and their installation and configuration for the laboratory and mini-project activities.

Human and Material Resources: Human resources

Administrative

The administrative staff will have support to help them translate the EU evaluation marks to the PC evaluation system as well as regarding the transference credits system.

The training of the administrative staff explains the use of the ECTS credits and its transference to the PC credit system as well as the grades equivalence among EU and the different PC systems.

Evaluation

Evaluation: Introduction

- A very important aspect of the learning process as it will allow us to determine the level of assimilation of knowledge and skills by students.
- Not only be focused on the technical knowledge of the subject but should also include assessment of those skills and competencies that students must acquire.
- Pay attention to how they have developed cognitive skills (analysis, synthesis, application, evaluation, and critique)
- Action skills (organizing time, resources, coordination, negotiating, tolerating)
- The assessment must take into account how students are acquiring the knowledge, skills and competencies and ensure that those who pass the course have appropriate capabilities
- Problem solving related to real world problems is motivating for students as they see direct application and better assimilate concepts
- Students identify the problem, research on how to solve it applying concepts and principles. If they work in teams, develop communication skills and collaborative work, developing analytic skills
- During the evaluation process large amount of information will be collected

Evaluation: Formation of Teams

- Teams of 2 or 3 students.
- Groups should remain invariant throughout the course.
- Students who form a team have a similar level of knowledge.
- To set the level of students, the first day of class an objective type test will be performed, with around 25 questions, each one with 3 possible answers and only one answer will be correct. In this way we can know the real starting level of the students. This test should contain questions on microprocessor-based hardware and programming system.
- Questions should be aimed to collect general aspects of the concepts required as prerequisites to begin to pursue the matter. For example:
    - Basic Engineering concepts
    - Programming (Software)

Evaluation: Student evaluation methodology

Level One: Attitude (Student engagement)

- Motivation that students have within the course.
- keep an ongoing dialogue with different student groups of the course, this should be maintained by the lecturer throughout the entire course.
- Special attention to students deliver activities in the time and manner agreed as it is a clear indicator of student motivation
- Deadline for delivery of the different activities that students set and meet deadlines. This should be something to evaluate.
- Tis level of evaluation should be a part of the final mark. This part can be 10% of the final overall rating.
- Identify those students with a special motivation for the subject because their attitude is above average
- It is important that students meet requirements of work deadlines in the subject. This will facilitate the professional future of the graduates in order to work in environments with strict deadlines.

Evaluation: Student evaluation methodology

Level Two: Learning

We determine the acquisition of knowledge and skills that students have acquired throughout the course.

Problems:

- Lecturer presents main ideas of lecture contents and proposes some application problems which student solves individually.
- The teachers throughout the course individually assess how the student has solved different problems.
- For example, through the problems we can assess how students interpret a transfer function of a sensor/actuator. How to interpret the voltage as a physical quantity. The analytical representation by an equation. We can see if the student can interpret the current temperature if the input value is X volts, etc.
- For the evaluation of the problem must take into account: Approach resolution procedures, steps followed in the resolution, final result, method, clarity of presentation and approach, inclusion of units of measure, focus on the important issues facing superfluous. In the event that the final result is not correct,

Evaluation: Student evaluation methodology

Level Two: Learning

Laboratory:

- A practical problem previously presented during lecture.
- Students work by teams of two/three students.
- During the lab sessions students will show the teacher how they are solving the proposed activities
- Teacher will make questions about how is the resolution of the activity.
- At the end of each lab session the teacher will rate each group based on the work done and the objectives achieved.
- For evaluation can take into account:

    - Introduction phase: will reflect 20% of the grade.
    - Reinforcement Phase: will reflect 40% of the grade.
    - Advanced stage: Will represent a 40% of the grade.

Evaluation: Student evaluation methodology

Level Two: Learning

Seminars:

- A panel discussion with student teams (around six students) is proposed, consisting generally of solving a problem by means of PBL.
- The teachers will meet with each of the groups who will present how they have raised the issue, what options for the resolution are viable and which ones have been taken.
- It is important that the teacher dialogue with all members of the group to identify how well attended and have acquired the relevant knowledge.
- Some aspects to be taken into account for the assessment:
    - Level of responsibility among group members.
    - Number of studied and analysis in terms of advantages and disadvantages of each possible solution solutions.
    - Quality technical report writing as to the work done
    - Defense of ideas and how to defend against constructive criticism
    - Interaction of knowledge of this area with other
    - Management of bibliographic sources
    - Extra work done with respect to the requested
    - Robustness of the proposed solution adopted
    - Etc.

Evaluation: Student evaluation methodology

Level Two: Learning

Mini-project:

- Dedicated to planning, design and development of the control system of a real problem design.
- The developed project will presented publicly to a jury composed of three lecturers.
- The jury assesses the following aspects including some transversal skills:
    - Report: Maximum score of this part is 25% of the total mark.
    - Oral presentation: The score of this part is 10% maximum. The team presents the work during maximum of 20 minutes.
    - Implementation: The score assigned to this part is 65% of total mark. After the oral presentation, the team shows the project application.

Evaluation: Student evaluation methodology

Level Two: Learning

Mini-project (Cont.):

In each of these sections in addition to the assessment of knowledge, the teacher should take into consideration and evaluate all the important skills and transversal skills for engineers:

- Cognitive skills: Analysis, synthesis, application, evaluation, critique, etc.
- Action skills: Organizing time, resources, coordination, negotiating, tolerating, etc.

The advantage of the methodology is that it allows including other skills that assessment allow an integrated formation of the student: *competition, working in teams, cooperation, oral presentations, budget management, report redaction, etc*.

The rating of these skills should be included in the appropriate rating to each of the evaluation issues associated at this level.

Evaluation: Student evaluation methodology

Level Three: Grading (outcome)

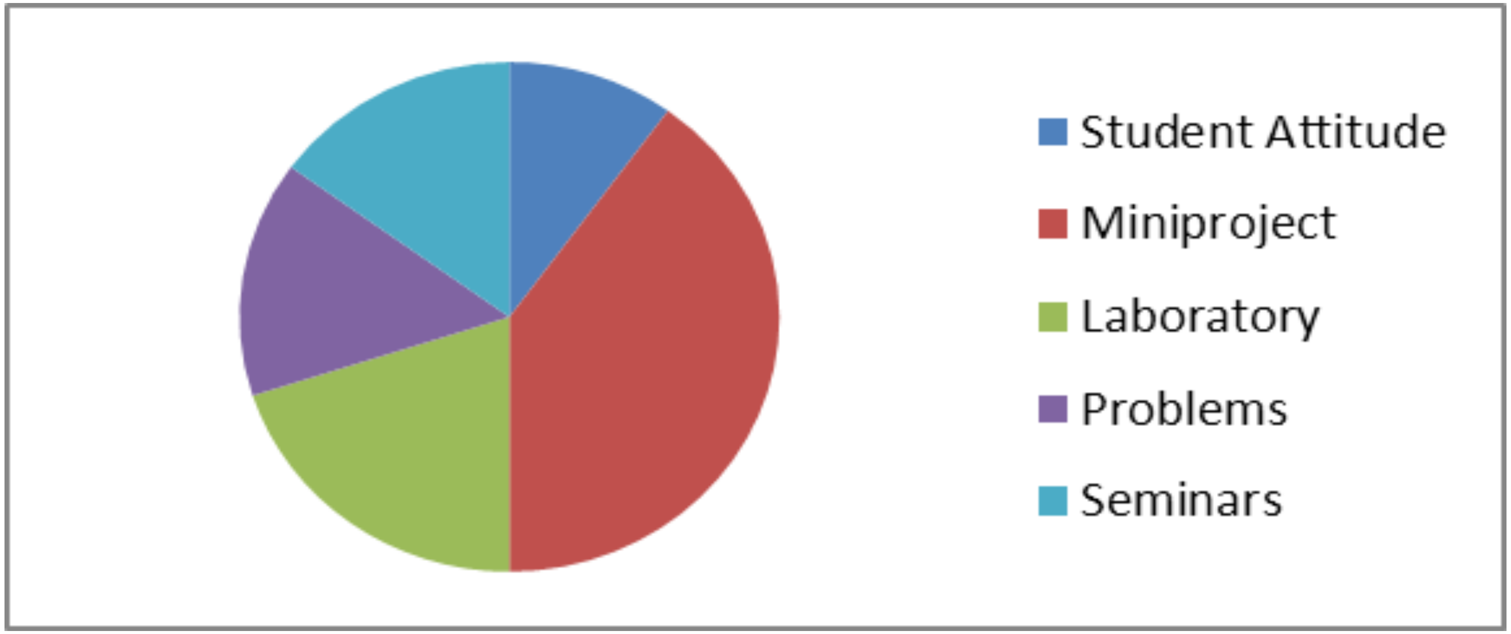
- We will collect all grades earned along the continuous assessment developed along the course
- Proceed to obtain the final grade for the course.
- We establish a just proportion between the different sections to ensure a fair rating for differentiating their individual acquisition of knowledge and skills against the student group work.
- The proposal will apply as follows:
    - The evaluation of the student attitude (A) a 10% of the final score.
    - The evaluation of the miniproject represents (MP) a 40% of the final score.
    - The evaluation of the Laboratory (L): 20% of the final score.
    - The evaluation of the Problems (P): 15% of the final score.
    - The evaluation of the Seminars (S): 15% of the final score.

Evaluation: Student evaluation methodology

Level Three: Grading (outcome) (Cont.)

For the calculation of the final grade (FG) can be followed as the following equation:

$$FG = A*0.1+MP*0.4+L*0.2+P*0.15+S*0.15$$



Evaluation: Assessment of the methodology system

- Evaluation of the system used for teaching this subject.
- Important to know the opinion of students and teachers involved to find out what has been done well and what parts could be improved
- In this sense the evaluation board system from two points of view:
  - Student point of view:
    - A survey among students to have information about the acceptance of the course.
    - Students can give their opinion at the end of the course and before obtaining their qualifications
    - Design a survey in a way that can be simple and easy to answer. For instance, can be made based on 6 questions with 5 possible answers ("A": Strongly Agree; "B": Agree; "C": Unsure; "D": Disagree; "E": Strongly Disagree) for each, rated from A to E marks.

Evaluation: Assessment of the methodology system (Cont.)

- The questions can be of the type:
  - Has the subject methodology facilitated your learning process?
  - Has every important concept of the subject been addressed in the miniproject?
  - Has the complexity level of every part of the subject been reasonable?
  - Has the activities promoted cooperation skills as in real work environments?
  - Have you felt motivated during the learning process?
  - Would you recommend taking this course to other students?

- Lecturer viewpoint:
  - The opinion of teachers is important to make an overall assessment of how the course has worked and what aspects should be improved.
  - Teachers should maintain an open dialogue throughout the course and at the end make the balance.
  - The aspects to be evaluated are for example:
    - Ratio of approved students
    - Quality of ratings
    - Amount of work done by teachers
    - Problems that have arisen and how they have been resolved
    - Possible updating of content,
    - Duplication and overlap with other subjects, etc.

Thanks!

MEDIS Project

Advanced Industrial Informatics Specialization Modules

Industrial Computers Module

Training

Saint Petersburg

Outline

1. The Problem
2. Project

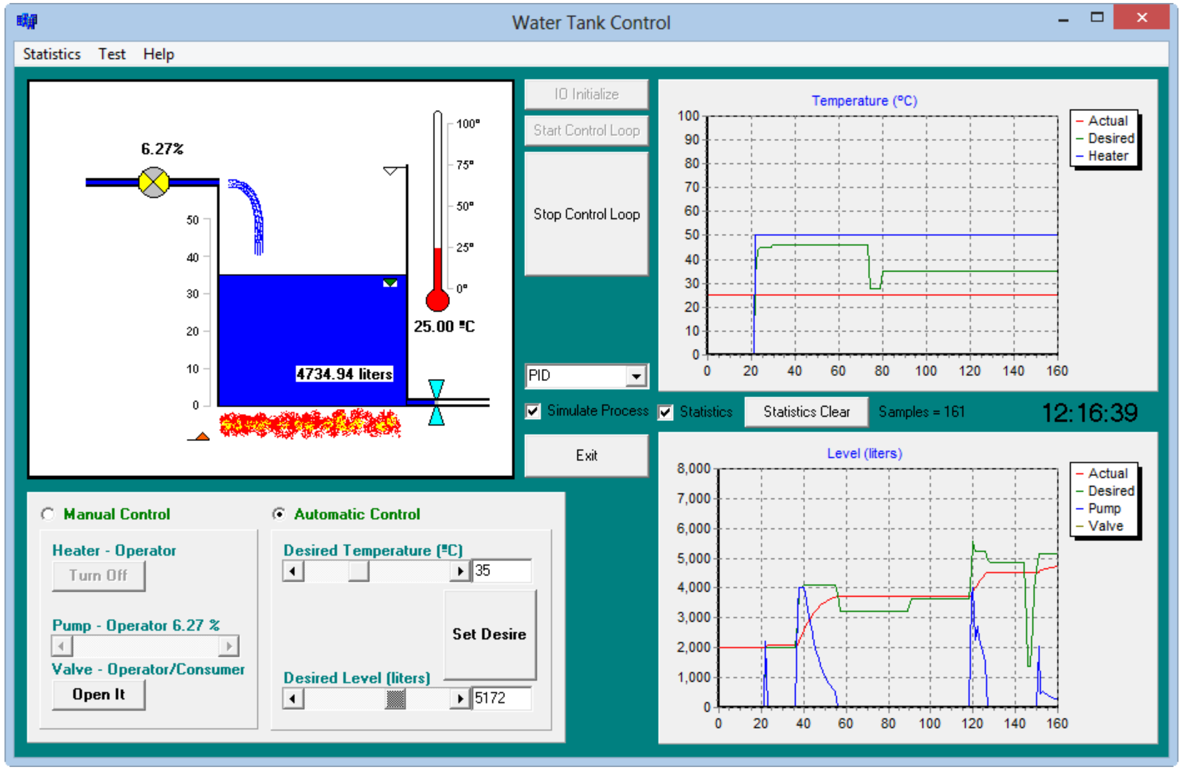
The Problem

The Problem – The Water Tank Controller

Teaching Philosophy – Problem Based Learning

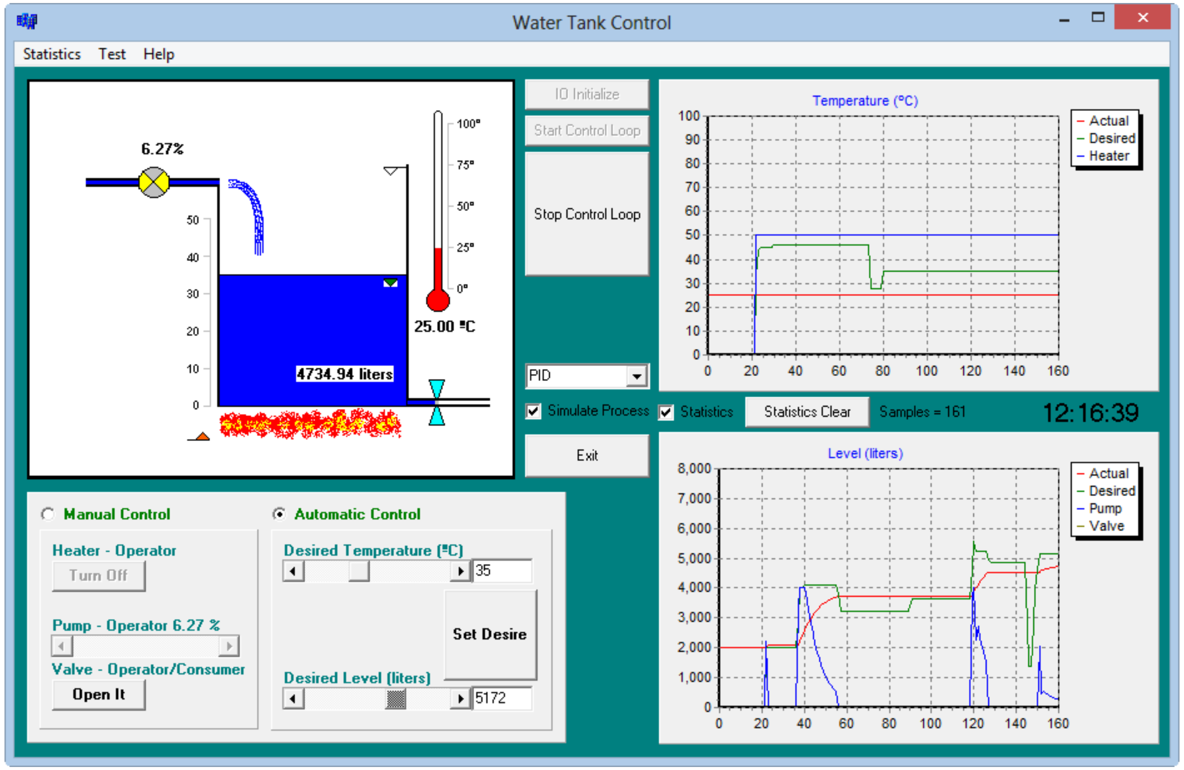
Water Tank

Water Tank Controller – Industrial Computer - based



Specification – Functionality

- Control of two variables: Water Temperature and Level
- Control Strategies: OnOff, PID, …
- Manual / Automatic Control
- Simulated / Real Process



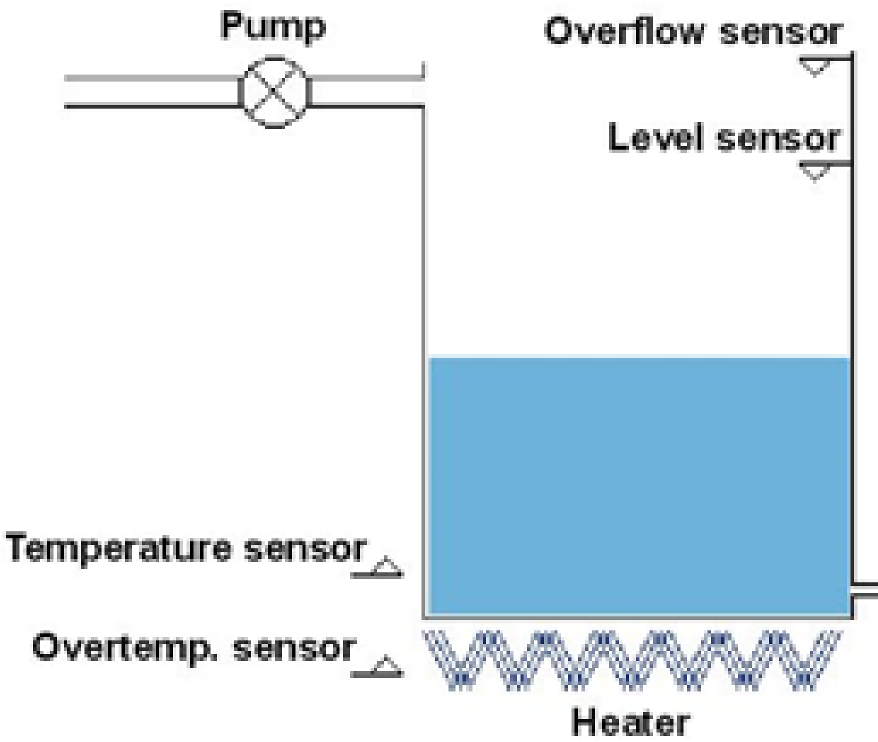
Specification – Process Interface

Sensors (Inputs):

- Water Level (Analog)
- Water Temperature (Analog)
- Overflow Alarm (Digital)
- Empty/Overheating Alarm (Digital)

Motors (Outputs):

- Water Input Flow - Pump (Analog)
- Water Output Flow - Valve (Digital)
- Water Heater (Analog)



Specification – Event-based Control

Hot Water Service

After the initialization, when user orders a new service,

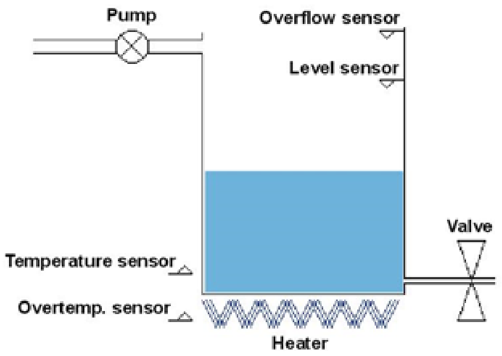
the tank is filled with cold water, using the input pump, until a predefined level,

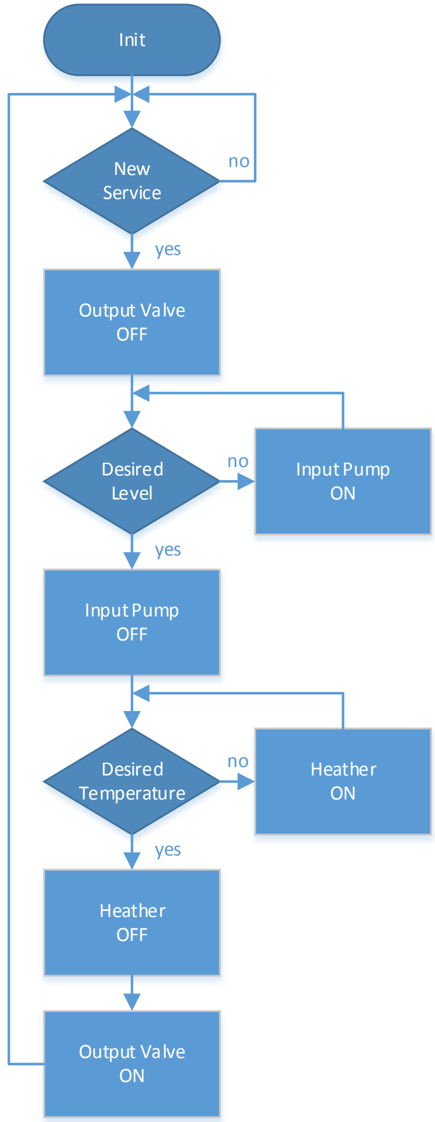
then

the water is heated, using the heather, until a predefined temperature,

finally

the hot water is served through the output valve, the controller waits until a new service request.





Specification – Continuous Control

Control Alternatives

Type of Control	Type of Driving Signal
On/Off	Digital (2 states)
	Analog (2 states)
P-discrete	Digital (N states)
	Analog (N states)
P	Digital PWM
	Analog
PID	Digital PWM
	Analog

Process Interface Resources

Data Acquisition Card – NI USB-6008

The National Instruments - NI USB-6008 (and USB-6009) are low-cost DAQ devices with easy screw connectivity and a small form factor.

With plug-and-play USB connectivity, these devices are simple enough for quick measurements but versatile enough for more complex measurement applications.

Web Site
http://sine.ni.com/nips/cds/view/p/lang/en/nid/201986

Operating System
Mac OS X, Windows 2000/XP, 7, 8, CE, Mobile, Vista

Driver
NI-DAQmx

NI USB-6008 – Characteristics

8 analog inputs

at 12 or 14 bits

up to 48 KSamples/s

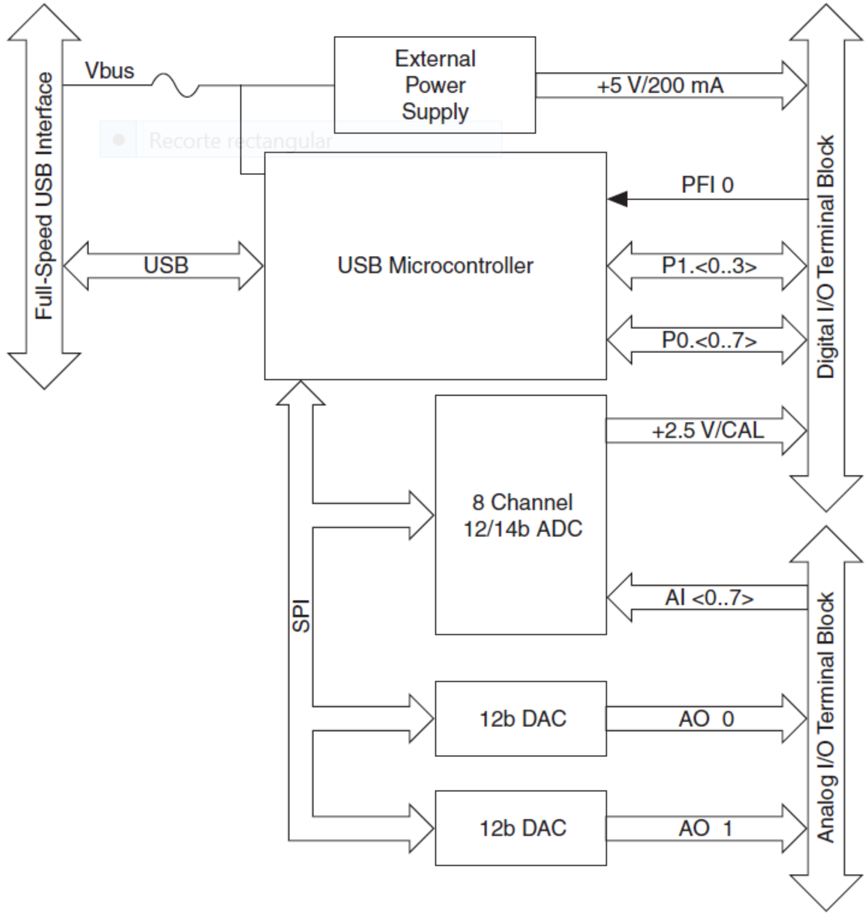
2 analog outputs

at 12 bits

software-timed

12 TTL/CMOS digital I/O lines

One 32-bit, 5 MHz counter



NI USB-6008 – Extensions – Simulators of Signals

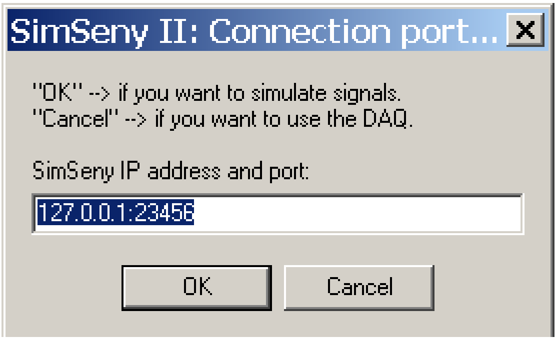
2 buttons
simulate sensors – digital inputs

2 LEDs
simulate motors – digital outputs

1 LM-335
temperature sensor – analog input

SimSeny

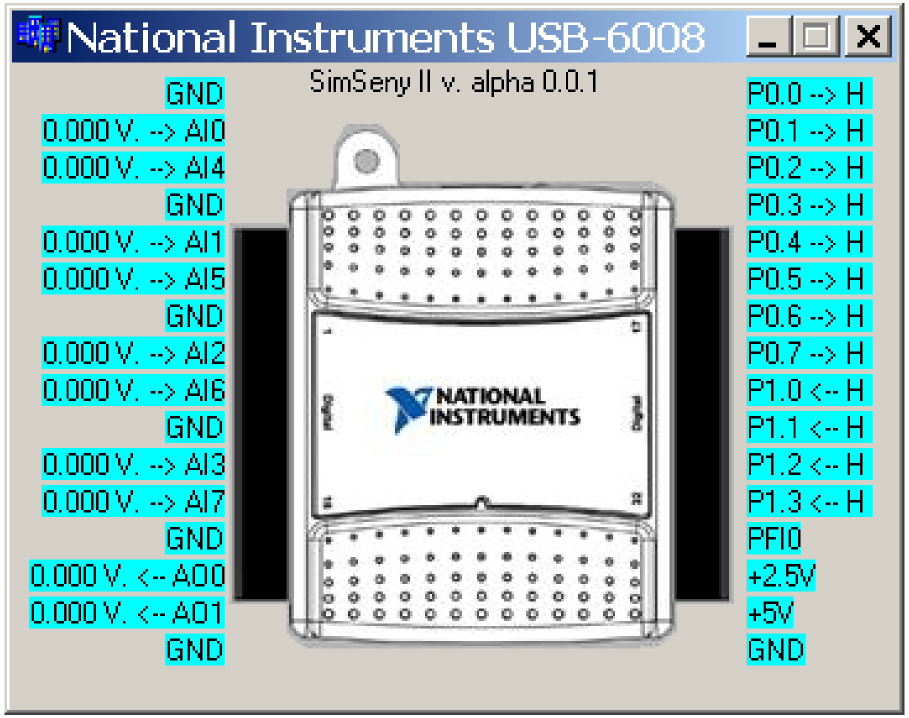
- SimSeny simulates at signal level
- Re-implements the API of a data acquisition card (e.g. NI USB-6008)
- It can be executed anyplace in the net (on a personal computer)

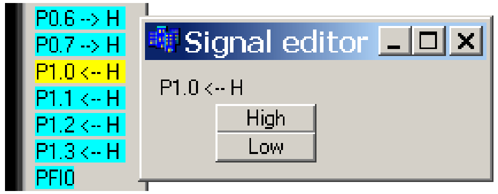


Web site:

http://www.disca.upv.es/aperles/simseny2/us_simseny2.html

SimSeny – Interface





Project

Program

Chapter 1 – Introduction to Industrial Informatics
Chapter 2 – Computer

Chapter 3 – Project Planning

Chapter 4 – Common Variables
Chapter 5 – Process Interface
Chapter 6 – User Interface
Chapter 7 – Tasks
Chapter 8 – Control

Chapter 9 – Project Integration

Project Phases

Software Engineering Techniques – *General Problem Solving Techniques applied to Software Development.*

Phases

recursively…

- Plan
- Design
- Develop

During Planning – Set Goals and Resources (*Requirements and Specifications)*

During Design – Improve the Solution (*Models)*

During Development – Apply and Validate the Solution (*Programs)*

Design Strategies

Two <u>Simultaneous</u> Design Strategies

- Top-Down
- Bottom-Up

Top-Down – Problem Decomposition (Modular Programming, Structured Programming)

Modular Decomposition
Criteria…

- Group Components by Functionality
- Try to Minimize the Number of Interfaces
- Decompose the Problems till a Tractable Level

Bottom-Up – Integrating Components

During Labs Exercises – Work at Components Level
During Project Exercises – Work at System Level (Integrate the Components)

Proposed Modules

Common Variables	• Defines a data structure that represents the state of the process under control. • It is implemented as a "blackboard", where different processes, knowledge's producers and knowledge's consumers, share global information. • Helps to minimize the number of interfaces of the set of modules.
User Interface	• Defines the interface with the user of the application. • Implements the inputs of the parameters and the goals. • Implements a readable output of the state of the process.
Process Interface	• Defines the interface with the process under control. • Performs the inputs from the sensors. • Performs the outputs to the motors.
Simulated Process Interface	• Defines dynamic models of the physical process, which are used to simulate the process and test the controller.
Control	• Makes the decisions about the control actions. • Applies different control strategies. • Defines a machine of states. • Defines continuous regulators.
Tasks	• Synchronizes the execution of the different tasks of the controller. • Implements a tasks dispatcher.

Implementation of the Modules

Each module is implemented as two files:

- Header File (Module Interface) – File Name Extension: H
- Implementation File – File Name Extension: CPP

Students Apply

Laboratories about Modular Decomposition of the Project

Design of the Modules – Sequence

1. Common Variables
2. User Interface
3. Process Interface
4. Tasks
5. Control

Common Variables Module

- The Common Variables Module is a central module, because it minimizes the number of interfaces in the system (the number of interdependences between the other modules)
- This simplifies the design, and helps the maintenance of the program.
- Once the variables that describe the problem are defined, then it is easier to define the functions that generate these variables.

Students Apply

Laboratories about Basic Programming in C++

Common Variables Module

Design Steps

- Analysis of the Problem
- Identification of the Key Variables that Describe the Problem
- Definition of the Variables using appropriate C++ Data Types

  The Data Structure is hidden into the Implementation of the Module

  Groups of variables:

  - Observations (from the Sensors)
  - Actions (to the Motors)
  - Desires (goals)
  - Decisions
  - Parameters of the controller (modes, control strategies, control state)
- Definition of pairs of Access Functions (Read and Write) for each of the variables

  Defined in the implementation file

  Declared (published) in the header file

UnitCommonVariables.h

```
// Overflow Sensor (Digital 2-States)

enum EnumOverflow {OVERFLOW_NO, OVERFLOW_YES};

EnumOverflow ReadOverflow(void);

void         WriteOverflow(EnumOverflow new_value);
```

UnitCommonVariables.cpp

```
#include "UnitCommonVariables.h"

// Overflow Sensor (Digital 2-States)

static EnumOverflow Overflow = OVERFLOW_NO;

EnumOverflow ReadOverflow(void)
    {
    return Overflow;
    }

void WriteOverflow(EnumOverflow new_value)
    {
    Overflow = new_value;
    }
```

UnitCommonVariables.h

```
// Temperature Sensor (Analog)
double ReadTemperature (void);
void   WriteTemperature(double new_value);
```

UnitCommonVariables.cpp

```
static double Temperature;
double ReadTemperature(void){return Temperature;}

void WriteTemperature(double new_value) {
   if( (new_value >= TEMP_MIN) && (new_value <= TEMP_MAX) )
      {
      Temperature = new_value;
      }
   else
      {
      emit TempERROR(); //Raises a signal
      }
```

Common Variables – List of Variables

UnitCommonVariables.cpp

```
                         // SENSORS //

// Overflow Sensor (Digital 2-States)
static EnumOverflow Overflow = OVERFLOW_NO;

// Overheating Sensor (Digital 2-States)
static EnumOverheating Overheating = OVERHEATING_NO;

// Stop Sensor (Digital 2-States)
static EnumStop Stop = STOP_NO;

// Level Sensor (Analog)
static double Level;

// Temperature Sensor (Analog)
static double Temperature;
```

UnitCommonVariables.cpp

```
// MOTORS //

// Heater Motor (Digital 2-States)
static EnumHeater Heater = HEATER_OFF;

// Valve Motor (Digital 2-States)
static EnumValve Valve = VALVE_CLOSED;

// Pump Motor (Analog)
static double Pump;
```

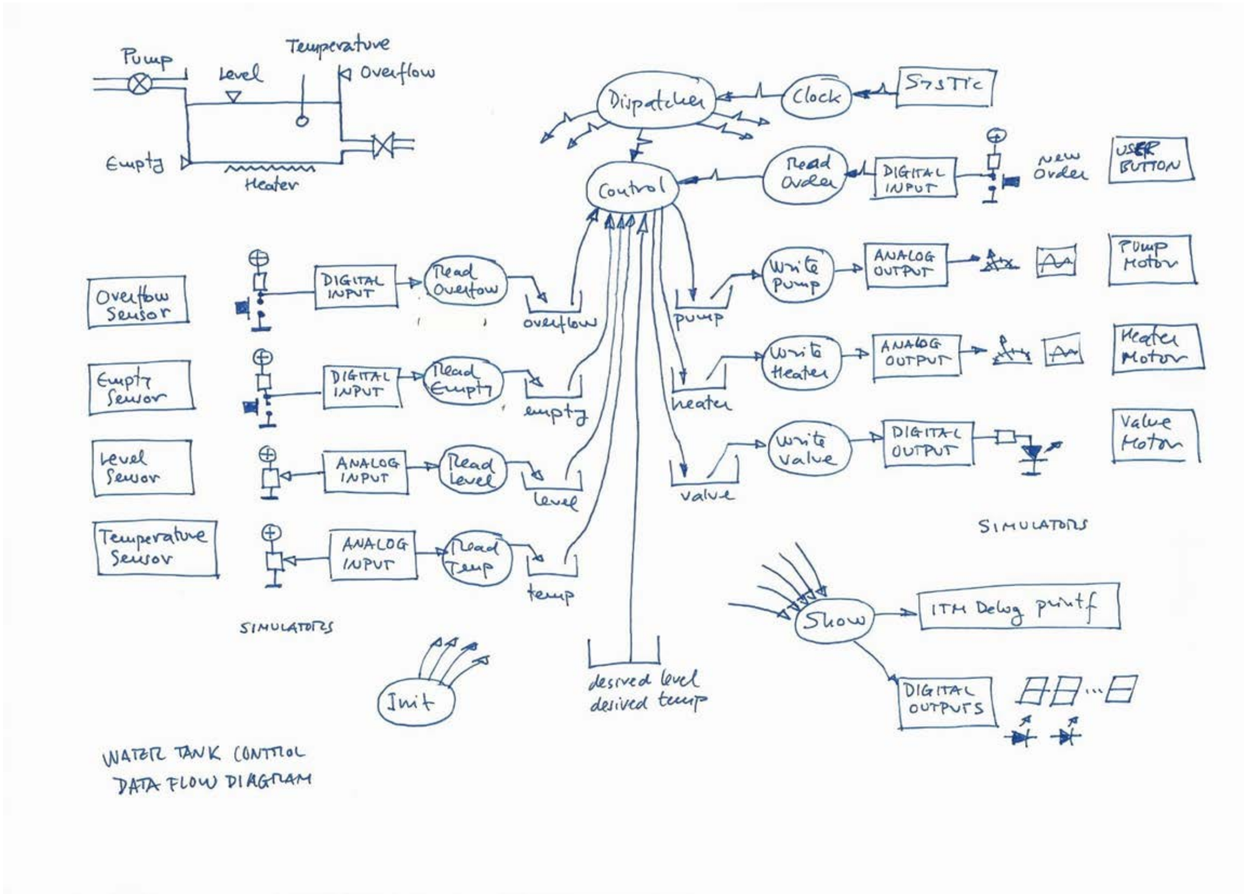
UnitCommonVariables.cpp

```
// DESIRES (GOALS) AND PARAMETERS //

// Level Desire (Analog)
static double LevelDesired;

// Temperature Desire (Analog)
static double TemperatureDesired;

// ControlMode (Digital N-States 2-States Now)
static EnumControlMode ControlMode = CONTROLMODE_MANUAL;

// ControlStrategy (Digital N-States 5-States Now)
static EnumControlStrategy ControlStrategy = CONTROLSTRATEGY_PID;

// ControlTimePeriod
static double ControlTimePeriod = DEFAULT_PERIOD;
```

UnitCommonVariables.cpp

```
// CONTROL AND SIMULATION //

// ControlState
static enum EnumControlState ControlState = CONTROLSTATE_OFF;

// Simulate Process
static bool SimulateProcess = true;

// Statistics
static bool Statistics = true;

// Samples
static unsigned long int Samples = 0;
```

Common Variables - Data Flow Diagram Example



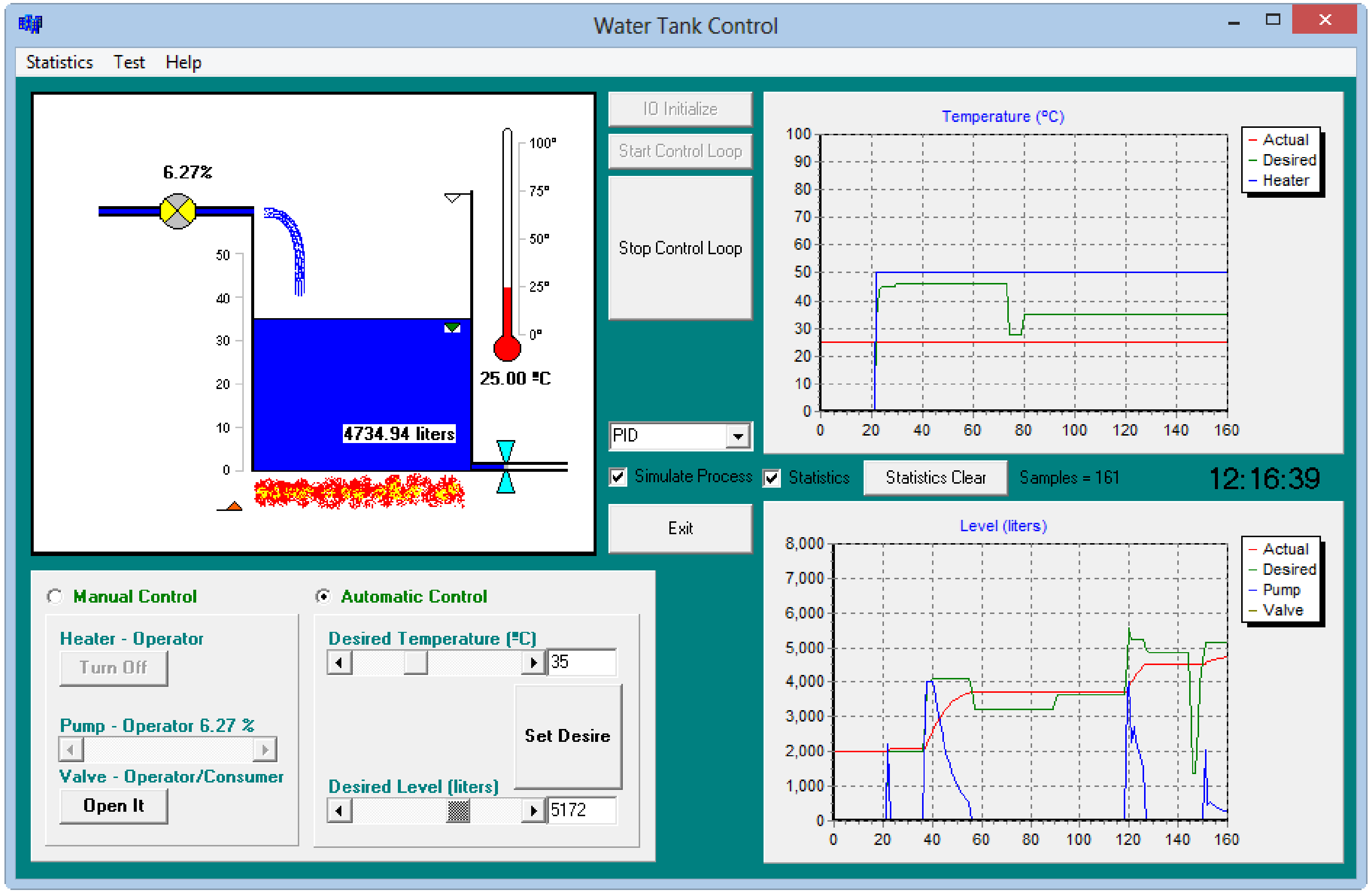
User Interface Module

- After the Common Variables Module, we continue with the User Interface, because the control elements of the user interface help to describe the specification of the controller (as a visual reminder of the actions to be performed)
- At the beginning of the User Interface Module implementation, most of the events from the user interface are not handled yet. As additional parts of the program are developed, the user interface becomes active.

Students Apply

Laboratories about Windows Events Loop, and about Design of Graphical User Interfaces.

User Interface Module - Example



Process Interface Module

- After the User Interface Module we continue with the Process Interface, implementing the sensors inputs and the motors outputs.
- The module defines Configuration functions and Operation (Input or Output) function.
- The functions of this module use the NIDAQmx library.

Students Apply

Laboratories about C bitwise operators and masks, and about the Data Acquisition Card, Digital Inputs, Digital Outputs, Analog Inputs, and Analog Outputs.

Simulated Process Interface Module

- To help validating the controller, students develop simulators of the physical process.

Students Apply

Laboratories about C++ programming, and about Timers.

UnitProcessInterface.h

```
// IO SYSTEM CONTROL //

void IO_Initialize (void);

bool IO_Initialized(void);
void IO_Finalize   (void);
```

UnitProcessInterface.h

```
// SENSORS (INPUTS) //

// Overflow Sensor (Digital Input 2-States)
EnumOverflow GetOverflow(void);

// Overheating Sensor (Digital Input 2-States)
EnumOverheating GetOverheating(void);

// Stop Sensor (Digital Input 2-States)
EnumStop GetStop(void);

// Level Sensor (Analog Input)
double GetLevel(void);

// Temperature Sensor (Analog Input)
double GetTemperature(void);
```

UnitProcessInterface.h

```
// MOTORS //

// Heater Motor (Digital Output 2-States)
void SetHeater(EnumHeater new_value);

// Valve Motor (Digital Output 2-States)
void SetValve(EnumValve new_value);

// Pump Motor (Analog Output)
void SetPump(double new_value);

// StopState
void SetStopState(void);
```

Unit**Simulated**ProcessInterface.h

```
void Simulate(void);

EnumOverflow    GetSimualtedOverflow   (void);
EnumOverheating GetSimulatedOverheating(void);

double GetSimulatedLevel      (void);
double GetSimulatedTemperature(void);
```

UnitProcessInterface.cpp

```
#define SS_FALSE_DAQ    //SimSeny: to avoid the needing of drivers

#include "ss_nidaqmx.h" //SimSeny: replaces "nidaqmx.h"
//#include "nidaqmx.h"  //NiDAQ real card

#include <stdio.h>
#include <stdlib.h>
#include <vcl.h>

#include "UnitProcessInterface.h"
```

UnitProcessInterface.cpp

```
#define IO_CARD_NAME                   "InfiDAQ"

#define DIGITAL_INPUT_PORT             "port0"
#define BIT_SENSOR_OVERFLOW            0
#define BIT_SENSOR_OVERHEATING         1
#define BIT_SENSOR_STOP                2

#define DIGITAL_OUTPUT_PORT            "port1"
#define BIT_MOTOR_VALVE                0
#define BIT_MOTOR_HEATER               1

#define CHANNEL_MOTOR_PUMP             "ao0"
#define CHANNEL_SENSOR_TEMPERATURE "ai0"
#define CHANNEL_SENSOR_LEVEL           "ai1"
```

UnitProcessInterface.cpp

```
static TaskHandle TaskDigitalOutputs; //Heater, Valve
static TaskHandle TaskDigitalInputs;  //Overflow, Overheating, Stop
static TaskHandle TaskAnalogOutput_Pump;
static TaskHandle TaskAnalogInput_Level;
static TaskHandle TaskAnalogInput_Temperature;

static bool Initialized = false;

static int LastDigitalOutput;
```

UnitProcessInterface.cpp

```
void IO_Error(char *message)
   {
   ShowMessage(message);
   exit(1);
   }

void IO_Test(int error_code, int line)
   {
   char message[2000];
   char buffer [1000];
   if(error_code != 0)
      {
      DAQmxGetErrorString(error_code, buffer, 1000);
      sprintf(message,
              "NiDAQmx Error\nLine: %d\nReason: %s\n",
              line, buffer);
      IO_Error(message); }}
```

UnitProcessInterface.cpp

```
void IO_Initialize(void)
   {
   IO_Test( DAQmxCreateTask("Digital Inputs Task",
                            &TaskDigitalInputs),
            __LINE__);

   IO_Test( DAQmxCreateDIChan(TaskDigitalInputs,
                              IO_CARD_NAME/**/"/"/**/DIGITAL_INPUT_PORT,
                              "",
                              DAQmx_Val_ChanForAllLines),
            __LINE__);
```

Continues…

UnitProcessInterface.cpp

```
IO_Test( DAQmxCreateTask("Digital Outputs Task ",
                         &TaskDigitalOutputs),
         __LINE__);

IO_Test( DAQmxCreateDOChan(TaskDigitalOutputs,
                           IO_CARD_NAME/**/"/"/**/DIGITAL_OUTPUT_PORT,
                           "",
                           DAQmx_Val_ChanForAllLines),
         __LINE__);

LastDigitalOutput = 0x0000;
```

Continues…

UnitProcessInterface.cpp

```
IO_Test( DAQmxCreateTask("Pump Task",
                         &TaskAnalogOutput_Pump), __LINE__);

IO_Test( DAQmxCreateAOVoltageChan(TaskAnalogOutput_Pump,
                           IO_CARD_NAME/**/"/"/**/CHANNEL_MOTOR_PUMP,
                                  "Pump Channel",
                                  0.0,
                                  5.0,
                                  DAQmx_Val_Volts,
                                  NULL),
         __LINE__);
```

Continues…

UnitProcessInterface.cpp

```
IO_Test( DAQmxCreateTask("Temperature Task",
                         &TaskAnalogInput_Temperature),
         __LINE__);

IO_Test( DAQmxCreateAIVoltageChan(TaskAnalogInput_Temperature,
                   IO_CARD_NAME/**/"/"/**/CHANNEL_SENSOR_TEMPERATURE,
                                  "",
                                  DAQmx_Val_RSE,
                                  0.0,
                                  10.0,
                                  DAQmx_Val_Volts,
                                  NULL),
          __LINE__);
```

Continues…

UnitProcessInterface.cpp

```
IO_Test( DAQmxCreateTask("Level Task",
                        &TaskAnalogInput_Level),
         __LINE__);

IO_Test( DAQmxCreateAIVoltageChan(TaskAnalogInput_Level,
                      IO_CARD_NAME/**/"/"/**/CHANNEL_SENSOR_LEVEL,
                                  "",
                                  DAQmx_Val_RSE,
                                  0.0,
                                  6.0,
                                  DAQmx_Val_Volts,
                                  NULL),
         __LINE__);

Initialized = true;
}
```

UnitProcessInterface.cpp

```
bool IO_Initialized(void)
   {
   return Initialized;
   }

void IO_Finalize(void)
   {
   if(Initialized)
      {
      SetPump  (0.0);
      SetValve (VALVE_CLOSED);
      SetHeater(HEATER_OFF);
      }
   }
```

UnitProcessInterface.cpp

```
EnumOverflow GetOverflow(void)   {
   uInt32 data;

   IO_Test( DAQmxStartTask(TaskDigitalInputs),
            __LINE__);
   IO_Test( DAQmxReadDigitalScalarU32(TaskDigitalInputs,
                                          0.0,
                                          &data,
                                          NULL),
            __LINE__);
   IO_Test( DAQmxStopTask(TaskDigitalInputs),
                          __LINE__);

   if((data & (1 << BIT_SENSOR_OVERFLOW)) == 0)  //0=YES, 1=NO
      return(OVERFLOW_YES);
   else
      return(OVERFLOW_NO);   }
```

UnitProcessInterface.cpp

```
EnumOverheating GetOverheating(void){…}
EnumStop GetStop(void){…}
```

UnitProcessInterface.cpp

```
double GetLevel(void)   {
   float64 volts;
   double liters;

   IO_Test( DAQmxStartTask(TaskAnalogInput_Level),
            __LINE__);
   IO_Test( DAQmxReadAnalogScalarF64(TaskAnalogInput_Level,
            1.0,
            &volts,
            NULL),
            __LINE__);
   IO_Test( DAQmxStopTask(TaskAnalogInput_Level),
            __LINE__);

   liters = (6.0 - volts) * (3800.0 / 6.0); // 6V=0liters, 0V=3800liters
   return(liters);   }
```

UnitProcessInterface.cpp

```
double GetTemperature(void){…}
```

UnitProcessInterface.cpp

```
void SetHeater(EnumHeater new_value)
   {
   if(new_value == HEATER_ON)
      LastDigitalOutput = LastDigitalOutput | (1 << BIT_MOTOR_HEATER);
   else
      LastDigitalOutput = LastDigitalOutput & ~(1 << BIT_MOTOR_HEATER);

   IO_Test( DAQmxStartTask(TaskDigitalOutputs), __LINE__);
   IO_Test( DAQmxWriteDigitalScalarU32(TaskDigitalOutputs,
                                       false,
                                       0.0,
                                       LastDigitalOutput,
                                       NULL),
            __LINE__);
   IO_Test( DAQmxStopTask(TaskDigitalOutputs),
            __LINE__);
   }
```

UnitProcessInterface.cpp

```
void SetValve(EnumValve new_value) {…}
```

UnitProcessInterface.cpp

```
void SetPump(double new_value)   {
   double volts;

   volts = new_value * (5.0/100.0); //0V=0%, 5V=100%

   IO_Test( DAQmxStartTask(TaskAnalogOutput_Pump),
            __LINE__);
   IO_Test( DAQmxWriteAnalogScalarF64(TaskAnalogOutput_Pump,
                                      false,
                                      1.0,
                                      volts,
                                      NULL),
            __LINE__);
   IO_Test( DAQmxStopTask(TaskAnalogOutput_Pump),
            __LINE__);
   }
```

Tasks Module

- After the Process Interface Module, we continue with the Tasks Module, defining a Clock and synchronizing the tasks of the control loop.
- Since the dynamics of the process is slow, we use a Software Timer and handle its Ticks events to iterate the control loop.

Students Apply

Laboratories about the Handling Events and Software Timers.

UnitTasks.h

```
void ControlLoopStart  (void);
void ControlLoopStop   (void);
void ControlLoopIterate(void);
```

UnitTasks.cpp

```
void ControlLoopIterate(void)   //Task Sequence
   {
   if(ReadControlState() == CONTROLSTATE_ON)
      {
      if(ReadSimulateProcess() == true)
         {
         SimulatedObserve();
         }
      else
         {
         Observe();
         }
      Desire();
      Decide();
      Act();
      }
   Inform();
   }
```

UnitTasks.cpp

```
void ControlLoopStart(void)
   {
   //Initialize in Safe State
   WriteStopState();
   SetStopState();
   //Control State ON
   WriteControlState(CONTROLSTATE_ON);
   }

void ControlLoopStop(void)
   {
   //Keep in Safe State
   WriteStopState();
   SetStopState();
   //Control State OFF
   WriteControlState(CONTROLSTATE_OFF);
   }
```

UnitTasks.cpp

```
void Observe(void)
   {
   //The Observe function establishes the current state of the problem
   //It reads from the Sensors and writes on the Common Variables
   WriteStop         ( GetStop    () );
   WriteOverflow     ( GetOverflow() );
   WriteOverheating( GetOverheating() );
   WriteLevel        ( GetLevel        () );
   WriteTemperature( GetTemperature() );
   }
```

UnitTasks.cpp

```
void SimulatedObserve(void)
   {
   Simulate(); //Step of simulation

   //From Simulated Sensors to Common Variables
   WriteStop        ( GetStop()                  ); //Not Simulated
   WriteOverflow    ( GetSimualtedOverflow()     );
   WriteOverheating( GetSimulatedOverheating() );
   WriteLevel       ( GetSimulatedLevel()        );
   WriteTemperature( GetSimulatedTemperature() );
   }
```

UnitTasks.cpp

```
void Desire(void)
   {
   //The Desire function establishes the control goals

   //In this implementation
   //   this function doesn't do anything in a direct way
   //   because the operator inputs and selections on the user interface
   //   establish the desired levels
   //The event handlers of the user interface
   //   perform that storing process
   };
```

UnitTasks.cpp

```
void Decide(void)
   {
   //Applies Control Strategies

   switch( ReadControlMode() )
      {
      case CONTROLMODE_MANUAL: //The Control is Manual
         //Nothing is done from here directly
         //User operates the process through the user interface controls
         //We keep this case for conceptual convenience
         break;

      case CONTROLMODE_AUTOMATIC: //The Control is Automatic
         Control(); // Control function
         break;
      }
   }
```

UnitTasks.cpp

```
void Act(void)
   {
   //From Common Variables to Motors

   SetPump  (ReadPump()  );
   SetValve (ReadValve() );
   SetHeater(ReadHeater());
   }
```

UnitTasks.cpp

```
void Inform(void)
   {
   if(ReadStatistics() && (ReadControlState() == CONTROLSTATE_ON))
      {
      WriteSamples(ReadSamples() + 1);
      CollectStatistics();
      }
   ShowTank();
   }
```

Control Module

- After the Tasks Module, we complete the system with the Control Module, implementing different control strategies.

Students Apply

Laboratories about C++ programming, and about Timers.

UnitControl.h

```c
void Control(void);
```

UnitControl.cpp

```
#include <stdlib.h>
#include "UnitCommonVariables.h"
#include "UnitControl.h"

void Control(void)
   {
   //Alarming Situations
   if((ReadOverflow()    == OVERFLOW_YES)||
      (ReadOverheating() == OVERHEATING_YES))
      {
      WriteHeater(HEATER_OFF);
      WriteValve (VALVE_CLOSED);
      WritePump  (0.0);
      emit LevelALARM();
      return;
      }

//Continues…
```

```
switch(ReadControlStrategy())
   {
   case CONTROLSTRATEGY_ONOFF:
      ControlOnOff();
      break;
   case CONTROLSTRATEGY_ONOFF_H:
      ControlOnOffHysteresis();
      break;
   case CONTROLSTRATEGY_P:
      ControlP();
      break;
   case CONTROLSTRATEGY_PI:
      ControlPI();
      break;
   case CONTROLSTRATEGY_PID:
      ControlPID();
      break;
   }
}
```

UnitControl.cpp

```
void ControlOnOff(void)
   {
   ControlOnOffLevel();
   ControlOnOffTemperature();
   }

void ControlOnOffHysteresis(void){…}
void ControlP(void){…}
void ControlPI(void){…}
void ControlPID(void){…}
```

UnitControl.cpp

```
void ControlOnOffLevel(void)
   {
   if( ReadLevel() < ReadLevelDesired() )
      {
      WritePump(100.0);
      }
   else
      {
      WritePump(0.0);
      }
   }
```

MEDIS Project

Advanced Industrial Informatics Specialization Modules

Industrial Computers Module

Training

Saint Petersburg

Outline

1. Introduction
2. Aims of the Industrial Computers Module, Learning Outcomes and Prerequisites
3. PBL – Case of Study
4. Learning Resources in the Laboratory

1 Introduction

Training for Technicians

A laboratory technician is required to set-up all the hardware and software tools.

Technicians will provide support on the installation and configuration of the software and hardware used to develop the laboratories and projects.

Regarding the technicians, the training course shows how to use the necessary development tools (hardware and software) and their installation and configuration for the laboratory and project activities.

2 Industrial Computers Module

Goal

Development of a Controller based on Industrial Computers

Learning Outcomes

Specific Skills

- Programing Techniques and Tools
- Design, Development and Validation of an Computer-based Controller (Programming Aspects)
- Project Management and Documentation

General Skills

- Problem Solving
- Collaborative Learning – Work in Team
- Negotiation and Presentation

Prerequisites

- Basic C Programming
- Basic knowledge on Digital Electronics and Automatic Control

Program

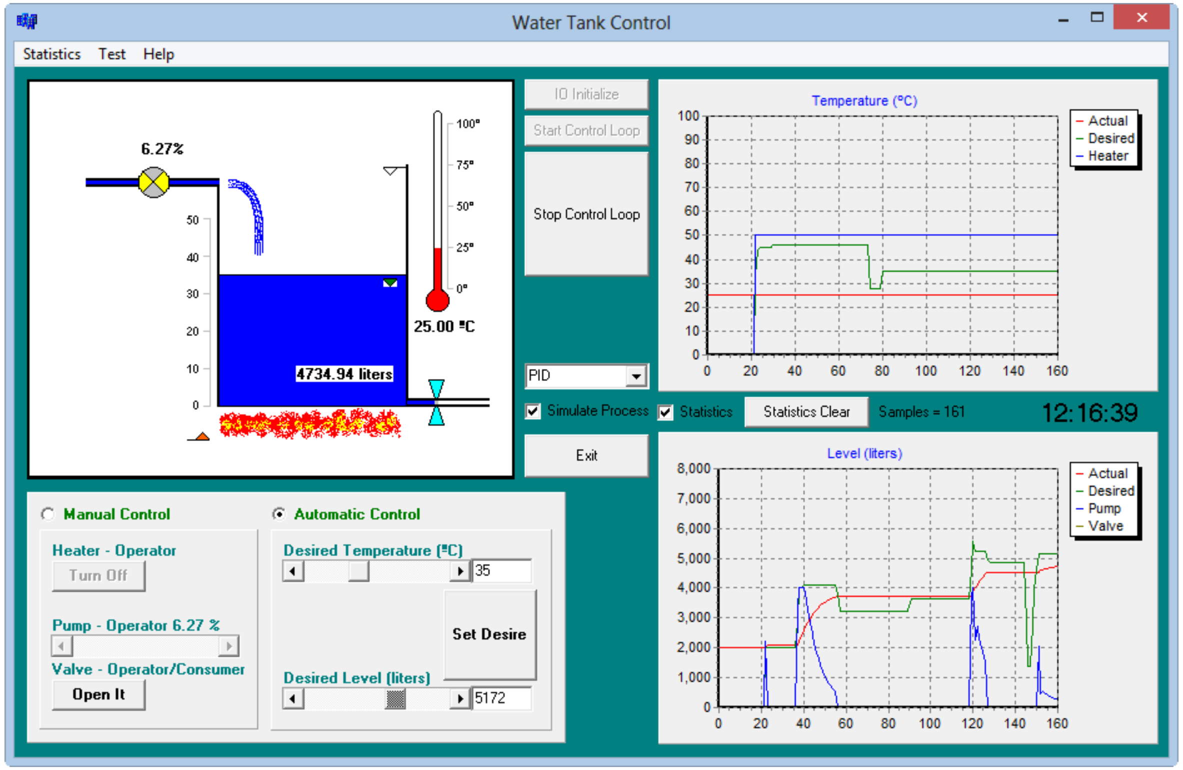
Chapter 1 – Introduction to Industrial Informatics
Chapter 2 – Computer
Chapter 3 – Project Planning
Chapter 4 – Programming + Common Variables
Chapter 5 – Process Interface
Chapter 6 – User Interface
Chapter 7 – Tasks
Chapter 8 – Control
Chapter 9 – Project Integration

3 PBL – Case of Study – The Water Tank Controller

Teaching Philosophy – Problem Based Learning

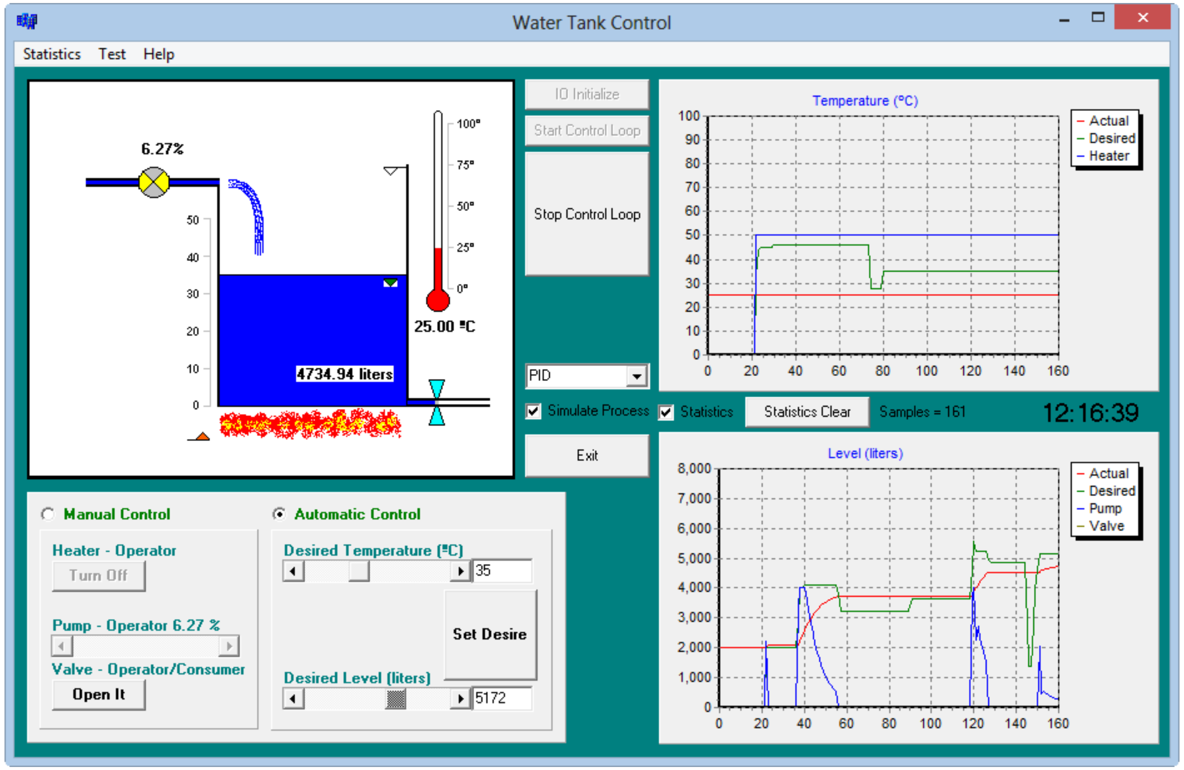
Water Tank

Water Tank Controller – Industrial Computer - based



Specification – Functionality

- Control of two variables: Water Temperature and Level
- Control Strategies: OnOff, PID, …
- Manual / Automatic Control
- Simulated / Real Process



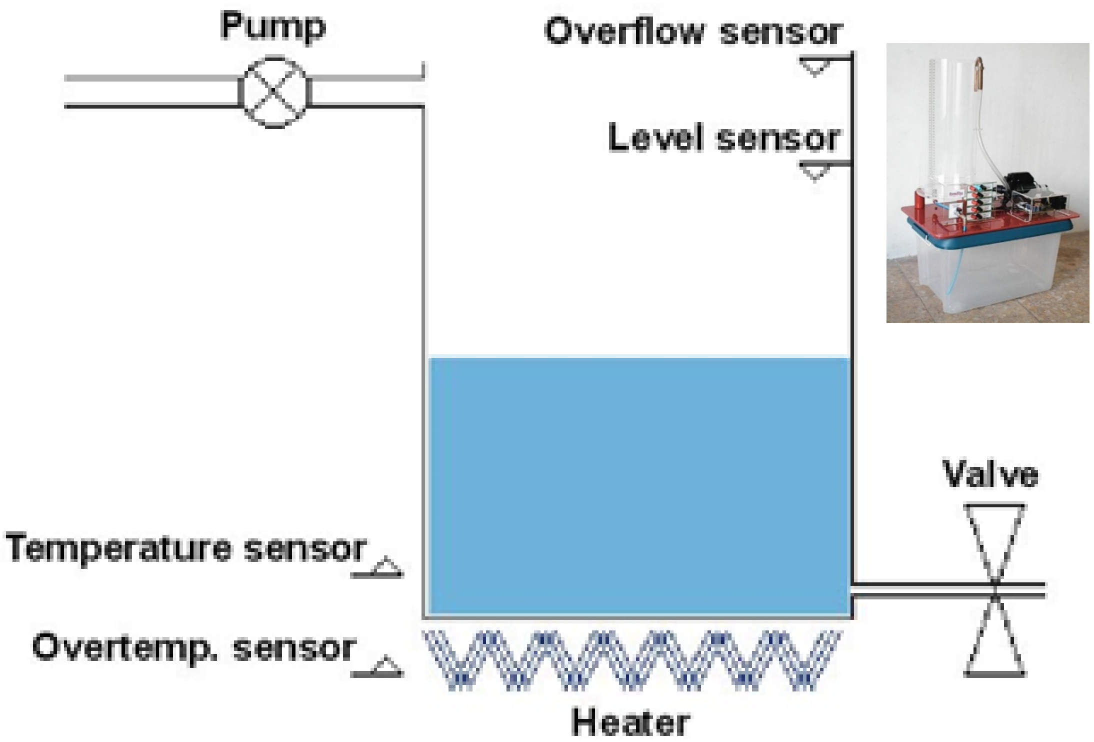
Specification – Process Interface

Sensors (Inputs):

Water Level (Analog)
Water Temperature (Analog)
Overflow Alarm (Digital)
Empty/Overheating Alarm (Digital)

Motors (Outputs):

Water Input Flow - Pump (Analog)
Water Output Flow - Valve (Digital)
Water Heater (Analog)



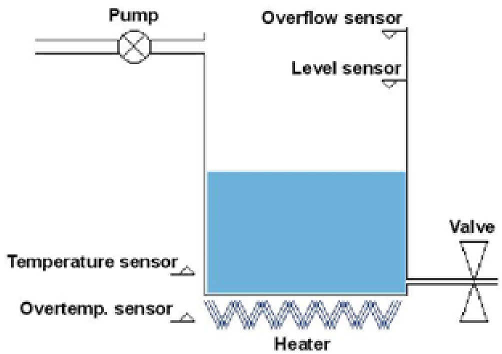
Specification – Event-based Control

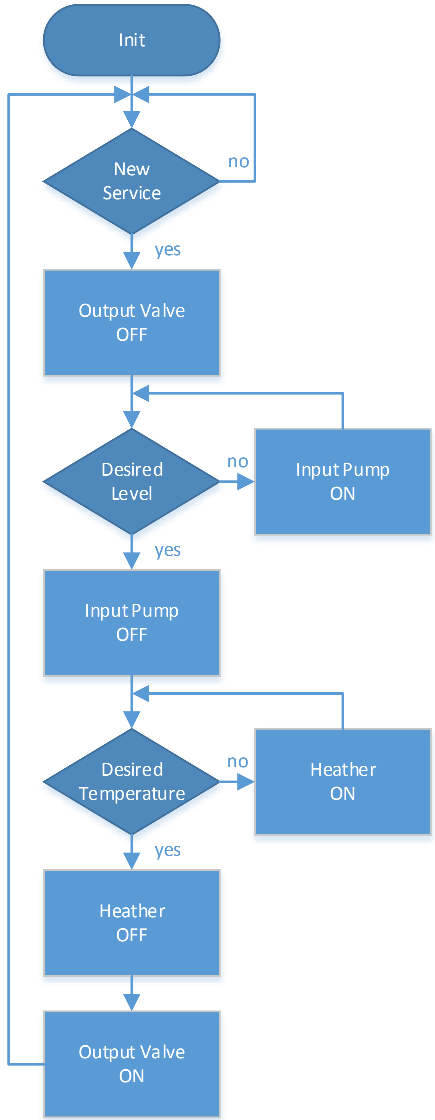
Hot Water Service

After the initialization, when user orders a new service,
the tank is filled with cold water, using the input pump,
until a predefined level,

then
the water is heated, using the heather,
until a predefined temperature,

finally
the hot water is served through the output valve,
the controller waits until a new service request.





Specification – Continuous Control

Control Alternatives

Type of Control	Type of Driving Signal
On/Off	Digital (2 states)
	Analog (2 states)
P-discrete	Digital (N states)
	Analog (N states)
P	Digital PWM
	Analog
PID	Digital PWM
	Analog

4 Learning Resources in the Laboratory

Essential Resources

The resources absolutely necessary to conduct the learning activities of the Industrial Computers Module

Optional Resources

Resources that make the learning process easier

Essential Resources

- Physical Process – Water Tank
  Real Process, Simulated Process, Simulated Process' Signals – Symseny

- Industrial Computer (General Purpose Computing Platform) – Personal Computer
  Operating System – Windows (XP, 7, 8)

- I/O System – Data Acquisition Card – National Instruments USB-6008

- Integrated Development Environment (IDE) – Qt Creator
  Programming Language / Compiler / Libraries – C++, MinGW, Qt

- Documentation
  Reference Manuals, User Manuals

Optional Resources

- Third Party Libraries (Application Development)
  e.g. OpenSceneGraph (3D), OpenCV (Computer Vision), …

- Tools for Sharing and Communicating (Collaborative Learning)
  Remote Desktop – Windows Remote Desktop, TeamView
  Source Code Sharing - Pastebin
  Versions Control - GitHub
  Subject Web Site - PoliformaT (UPV Sakai System)

- Tools for Documenting
  Text – MS-Office/OpenOffice
  Bitmaps – GIMP
  Vector Graphics – InkScape

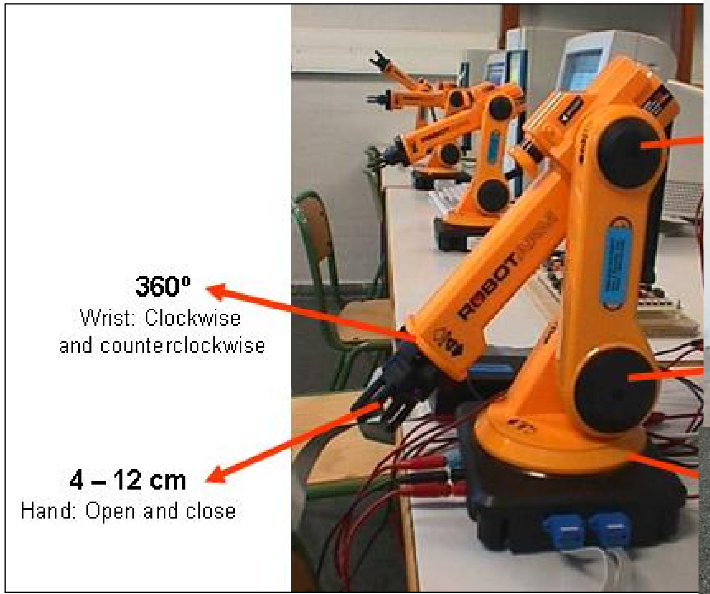
Physical Process – The Water Tank (or other Processes)

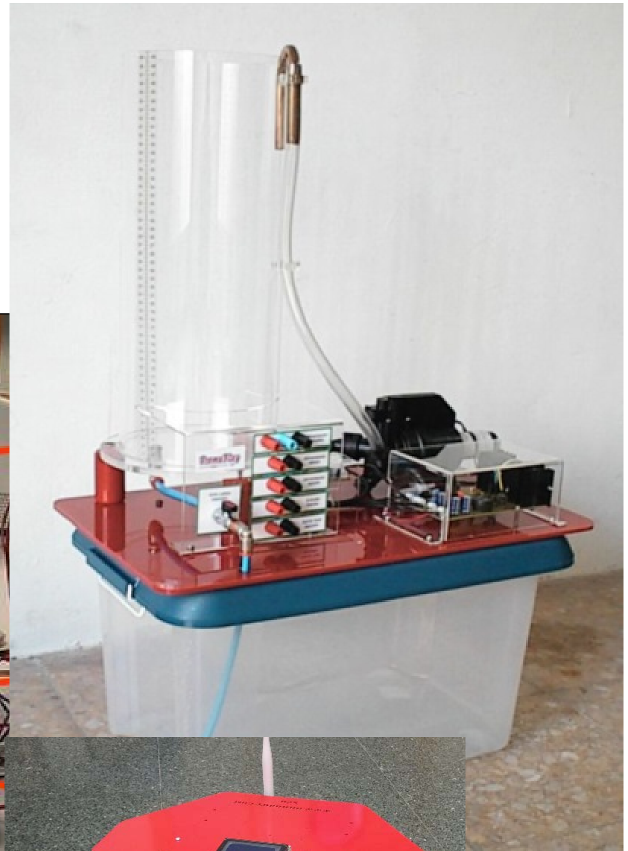
PBL: Control Problem

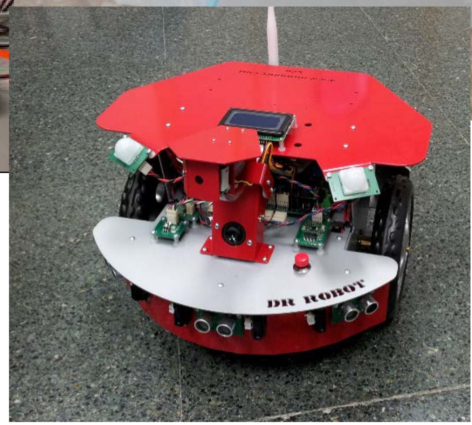
Real/Reduced-size Physical Processes vs Simulated

- More Expensive
- Room Needed

+ Let Address Real Issues
+ More Tangible (Motivating)







Technician Duties

- Installation
- Maintenance

Simulated Process – The Water Tank (or other Processes)

Simulating at physical model level

- As part of the Project activities, students develop simulators of the physical process' dynamics
- They use the simulators to test the controller
- The simulators run on the computer

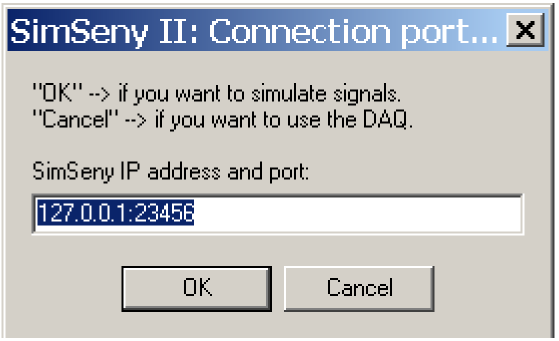
Simulating at signal level

- SimSeny simulator

SimSeny

Simulation Tool Developed by Prof. Angel Perles

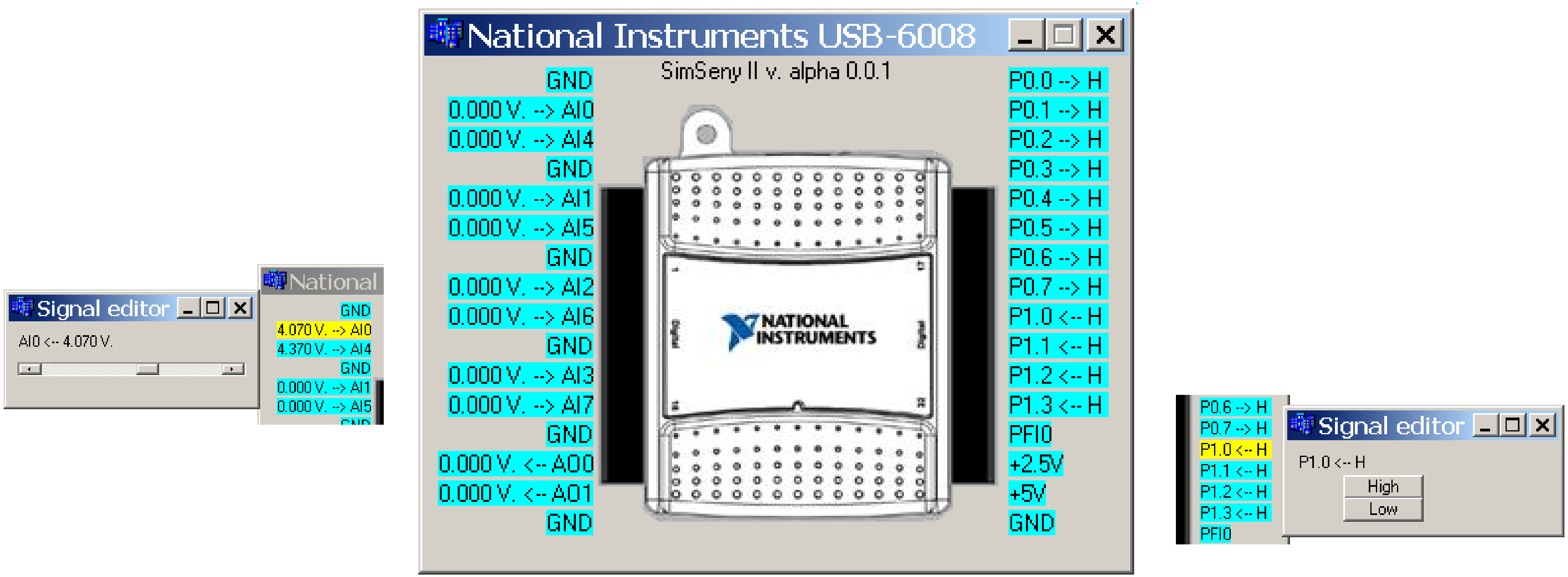
- SimSeny simulates at signal level
- Re-implements the API of a data acquisition card (e.g. NI USB-6008)
- It can be executed anyplace in the net (on a personal computer)



Web site:

http://www.disca.upv.es/aperles/simseny2/us_simseny2.html

SimSeny – Interface



Industrial Computer (Personal Computer)

PC Requirements

- Basic Configuration e.g. Dual Core
- USB frontal ports
- LAN and Internet Access

PCs in the laboratory

Technician Duties (installation, configuration, maintenance)

Students Laptop

Technician Duties (to provide recommendations and advice)

Data Acquisition Card – NI USB-6008

The National Instruments - NI USB-6008 (and USB-6009) are low-cost DAQ devices with easy screw connectivity and a small form factor.

With plug-and-play USB connectivity, these devices are simple enough for quick measurements but versatile enough for more complex measurement applications.

Web Site
http://sine.ni.com/nips/cds/view/p/lang/en/nid/201986

Operating System
Mac OS X, Windows 2000/XP, 7, 8, CE, Mobile, Vista

Driver
NI-DAQmx

NI USB-6008 – Characteristics

8 analog inputs

at 12 or 14 bits

up to 48 KSamples/s

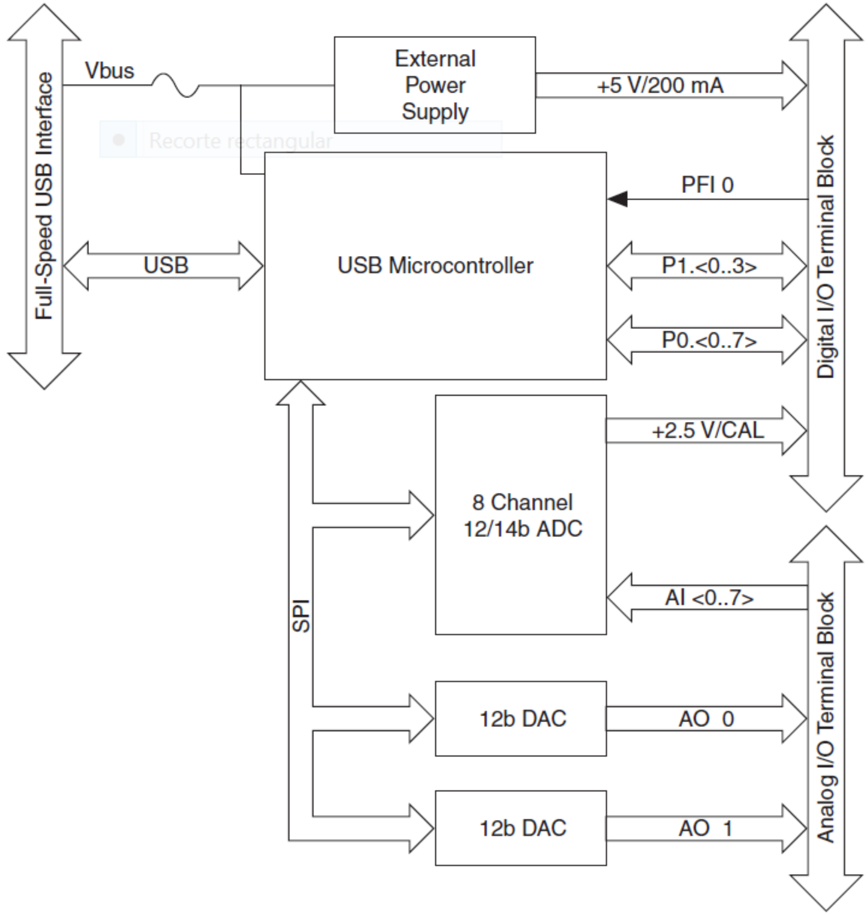
2 analog outputs

at 12 bits

software-timed

12 TTL/CMOS digital I/O lines

One 32-bit, 5 MHz counter

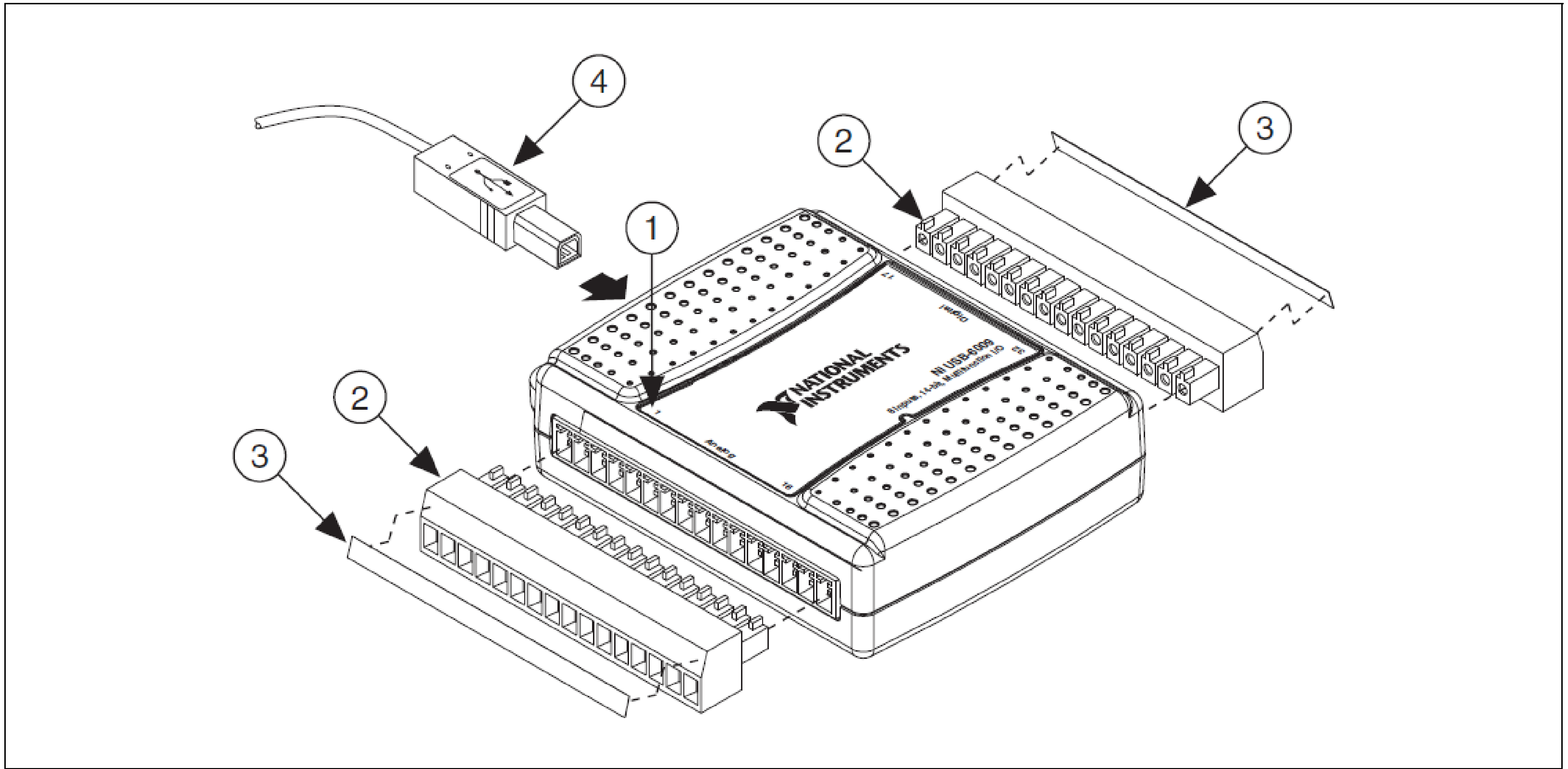


NI USB-6008 vs NI USB-6009

Comparison Tables

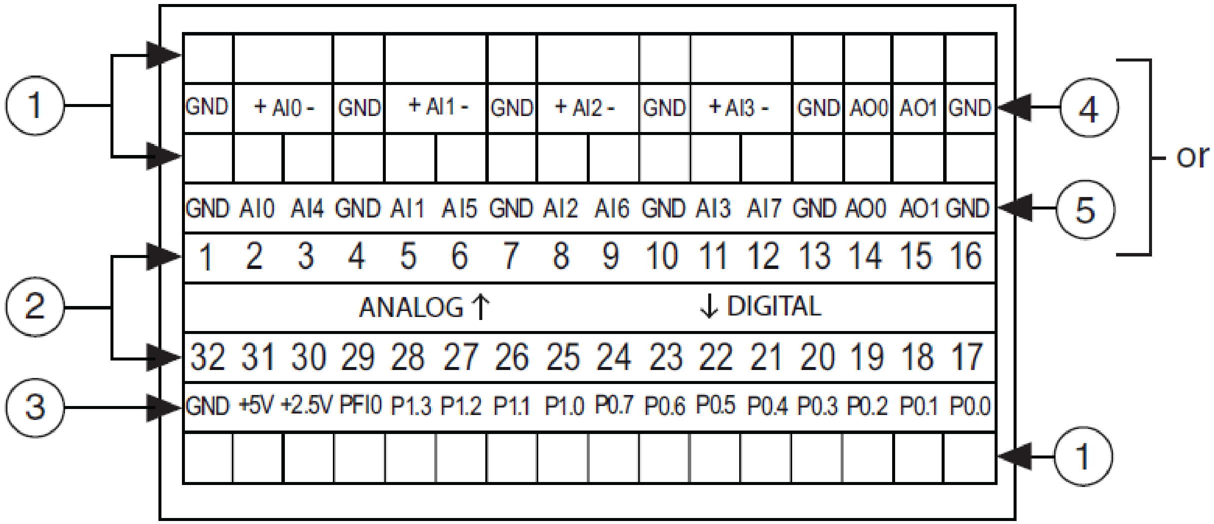
Product	Analog Inputs	Input Resolution	Max Sampling Rate (kS/s)	Analog Outputs	Output Resolution	Output Rate (Hz)	Digital I/O Lines	32-Bit Counter	Triggering
USB-6008	8 single-ended/4 differential	12	10	2	12	150	12	1	Digital
USB-6009	8 single-ended/4 differential	14	48	2	12	150	12	1	Digital

NI USB-6008 – Components



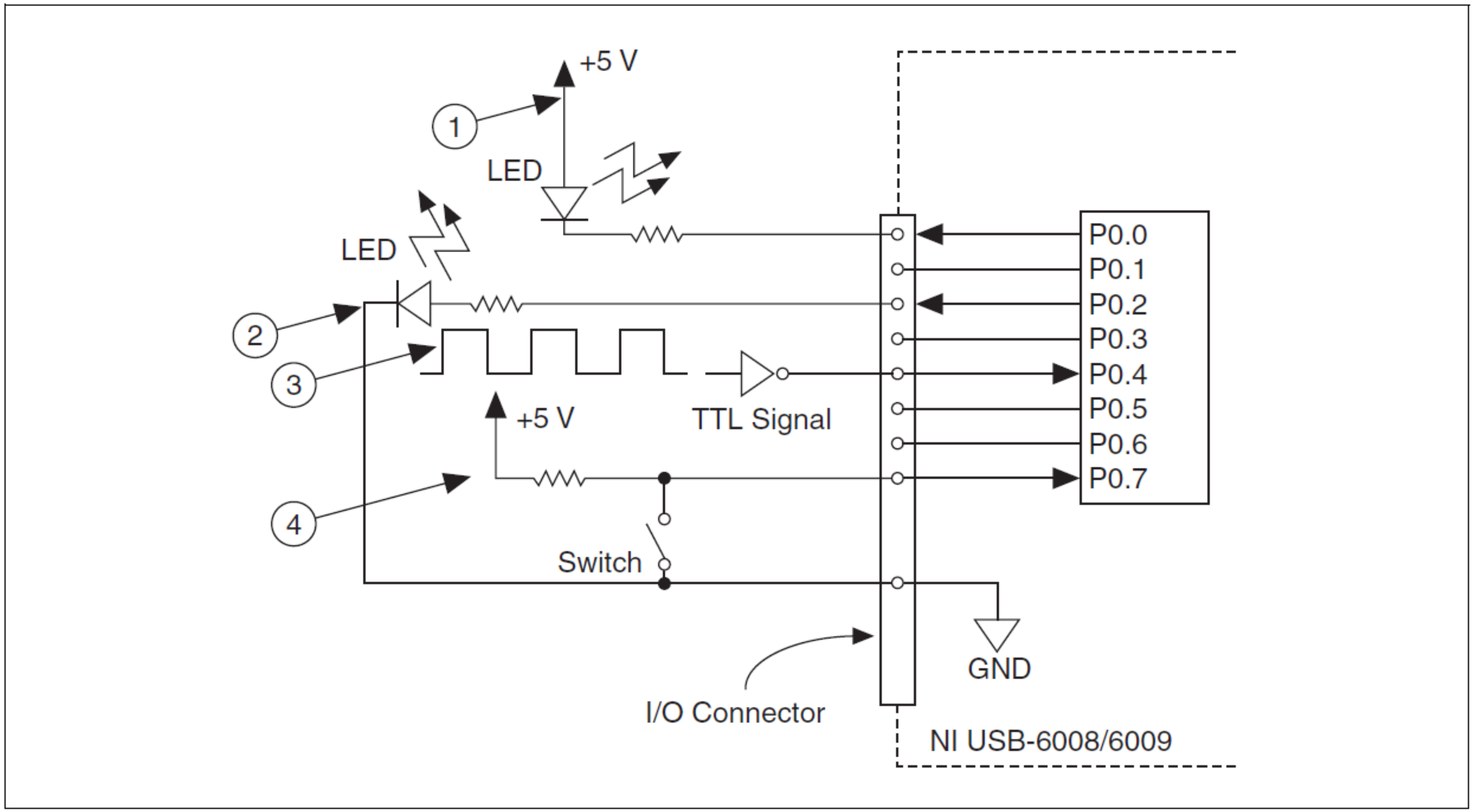
1	Overlay Label with Pin Orientation Guides	3	Signal Label
2	Screw Terminal Connector Plug	4	USB Cable

NI USB-6008 – Labels

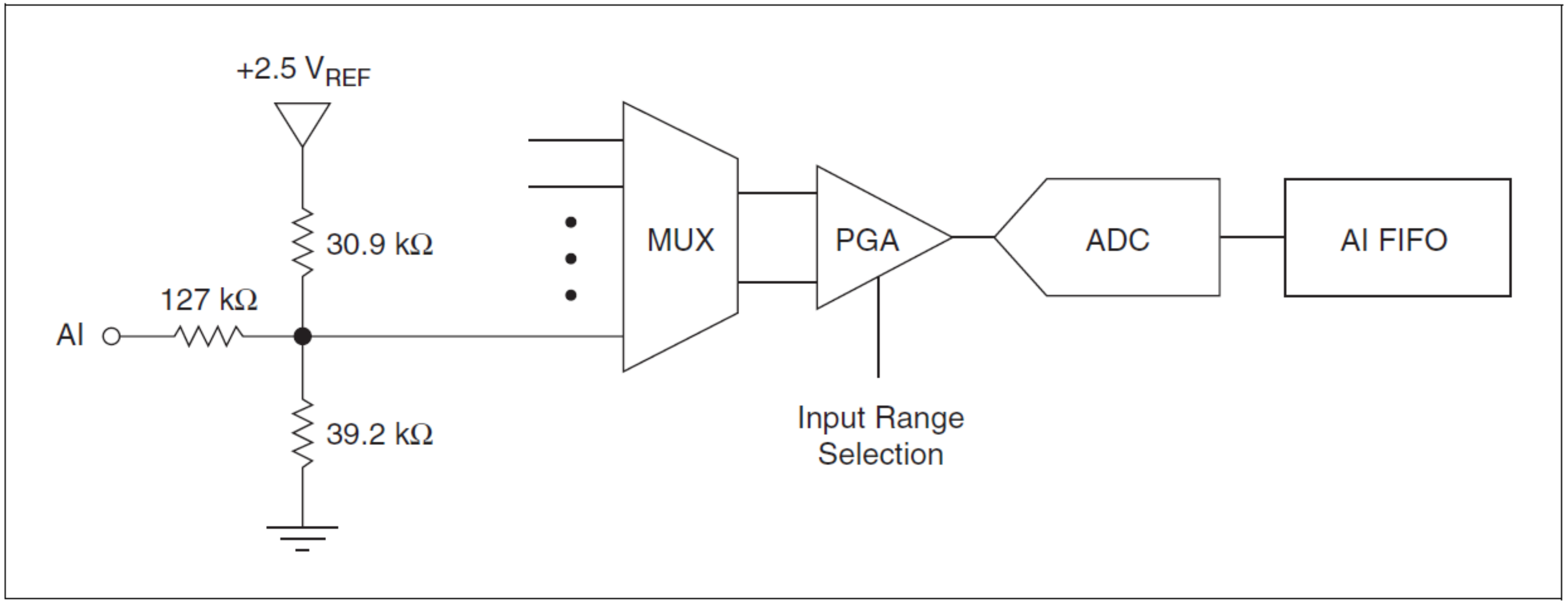


1	User-Defined Custom Label	4	Analog Input Differential Signal Name Label
2	Terminal Number Label	5	Analog Input Single-Ended Signal Name Label
3	Digital I/O Label		

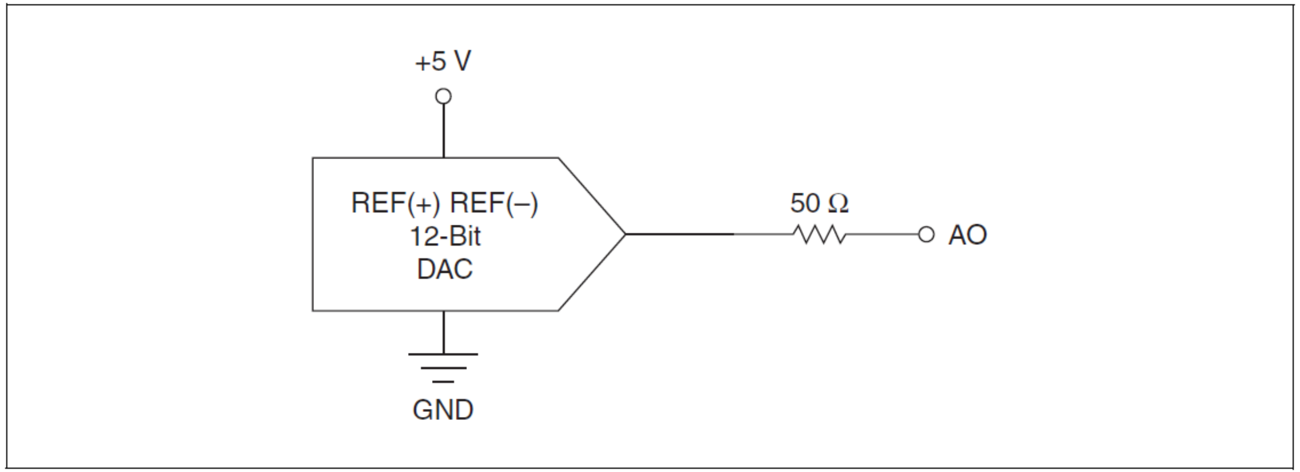
NI USB-6008 – Digital Input/Output



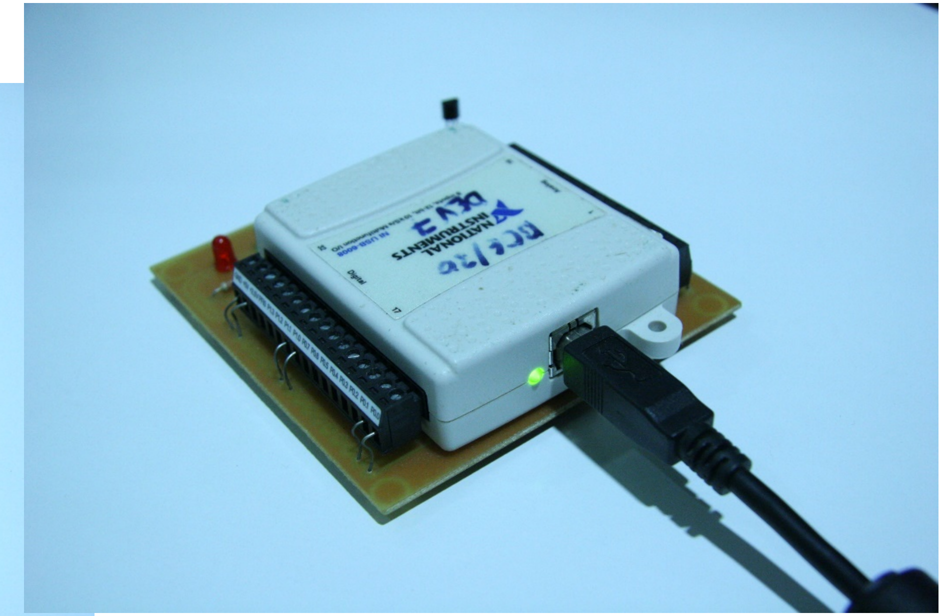
NI USB-6008 – Analog Input



NI USB-6008 – Analog Output



NI USB-6008 – Extensions – Simulators of Signals



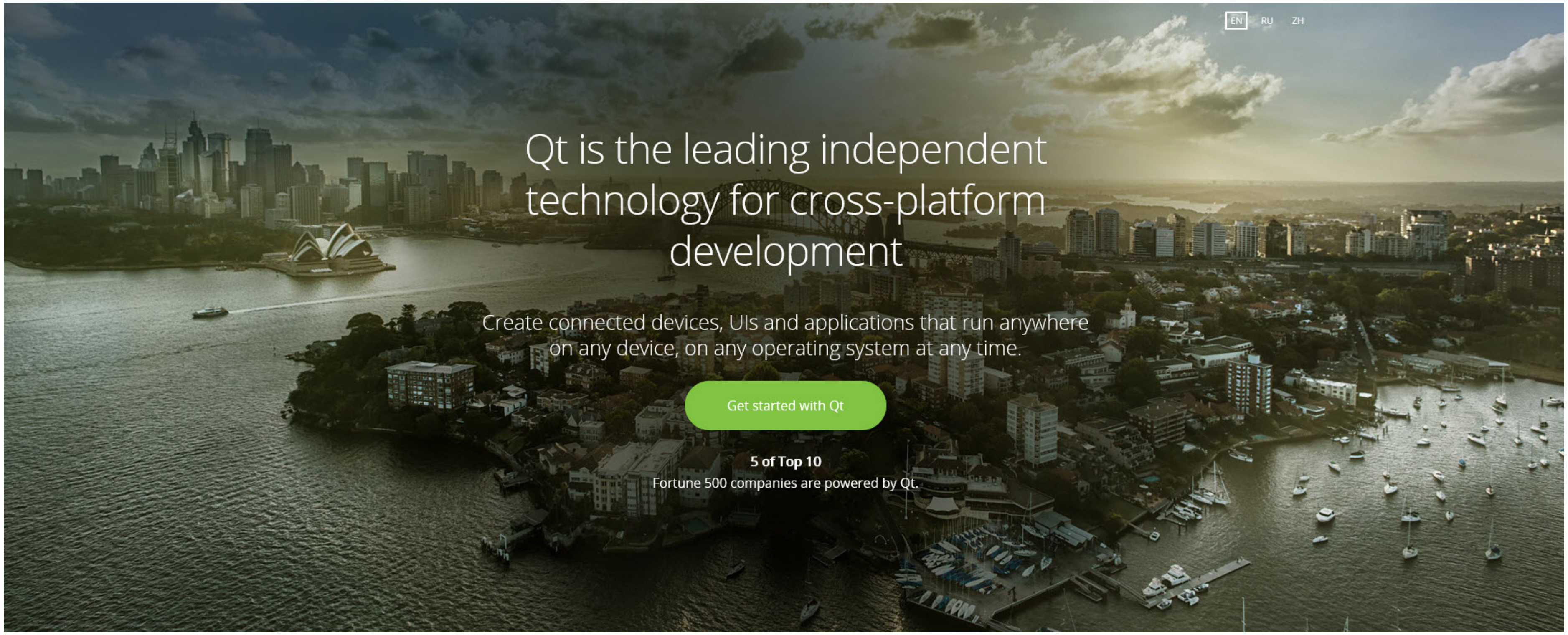
NI USB-6008 – Extensions – Simulators of Signals

2 buttons
simulate sensors – digital inputs

2 LEDs
simulate motors – digital outputs

1 LM-335
temperature sensor – analog input

IDE – Qt Creator



Web site:

https://www.qt.io/

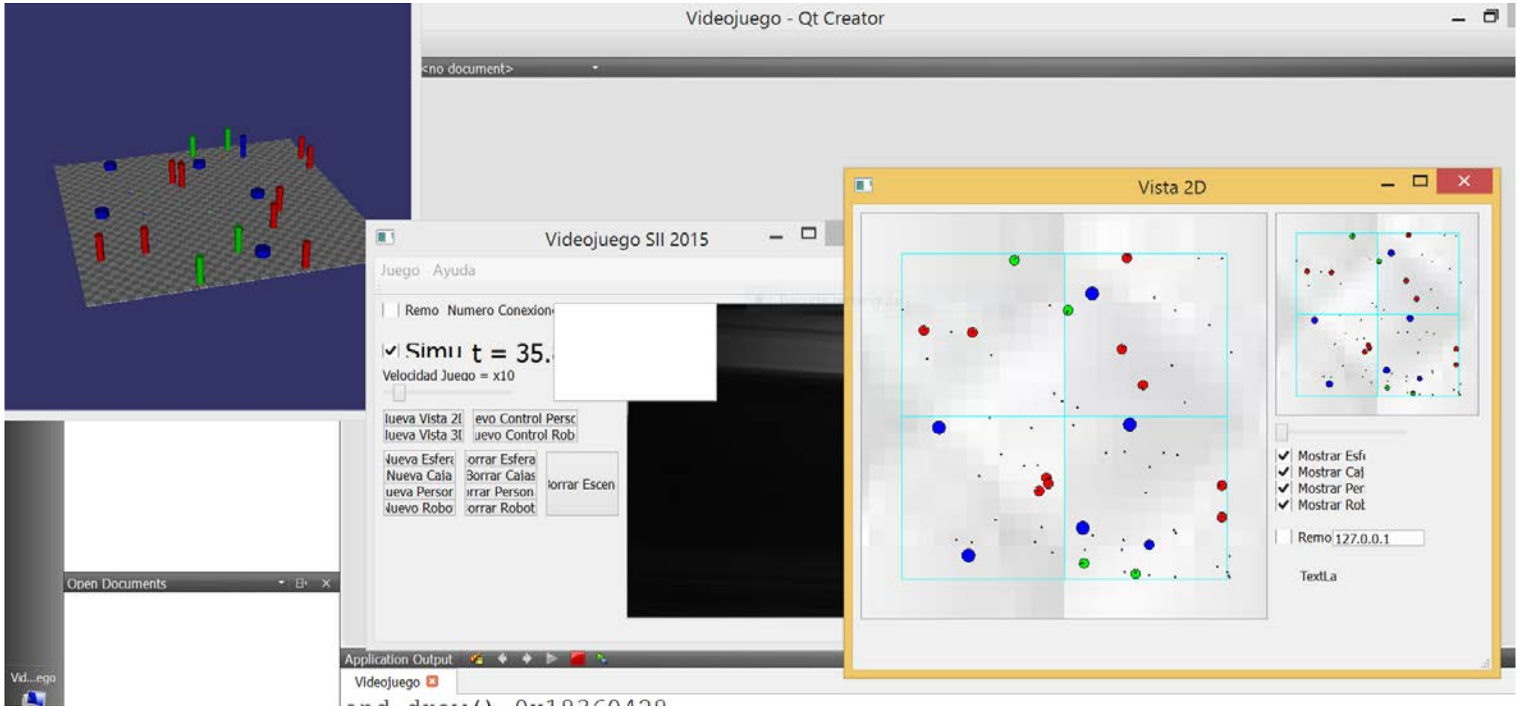
Qt Framework

Reasons for Qt

- Open Software
- Multiplatform
- C++ Native Code
- Big Community
- Many Success Applications

Optional Resources

- Third Party Libraries (Application Development)
  e.g. OpenSceneGraph (3D), OpenCV (Computer Vision), …



Optional Resources

- Tools for Sharing and Communicating (Collaborative Learning)
  Remote Desktop – Windows Remote Desktop, TeamView - https://www.teamviewer.com/
  Source Code Sharing – Pastebin - http://pastebin.com/
  Versions Control – GitHub - https://github.com/
  Subject Web Site - PoliformaT (UPV Sakai System)

- Tools for Documenting
  Text – MS-Office/OpenOffice
  Bitmaps – GIMP - http://www.gimp.org/
  Vector Graphics – InkScape - https://inkscape.org/

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

www.upv.es

U.PORTO

FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

MIEEC

Integrated Master in Electrical and Computer Engineering

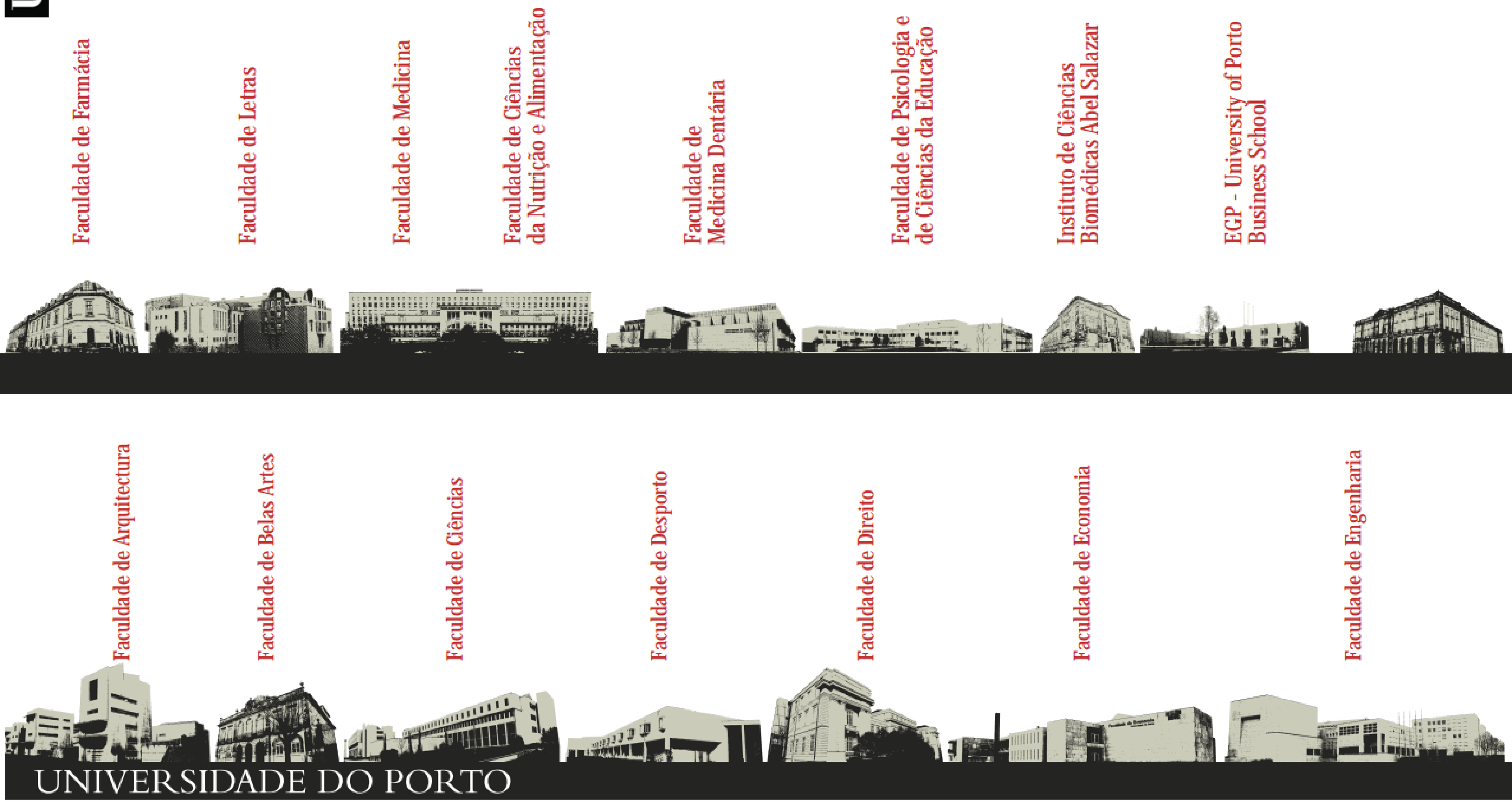
MEDIS (TEMPUS) meeting, St. Petersburg, Russia, 2015-may-18-22

The city of Porto - Portugal

U.PORTO 100

University of Porto

Três pólos
na cidade do Porto
acolhem as
14 faculdades,
a business school e perto de
70 unidades
de investigação da
Universidade do Porto.



Faculty of Engineering

- Unique in the country…
- Largest within UP…
- Several Rectors came from FEUP…

- Mobility: In => **241** / out => **319**
- Registered students: **8199**
- Total campus area: **93 918 m2**
- Teaching labs: **191**
- Computers for teaching: **1560**
- Lab investiment: **1,21 M€**

(23/05/2014)

Electrical & Computer Engineering

- Largest department in FEUP
  - **80** professors + **15** technicians
  - **1360** students (all 3 cycles)
- Admitted students: **200**
- Lowest grade: **14.7/20**

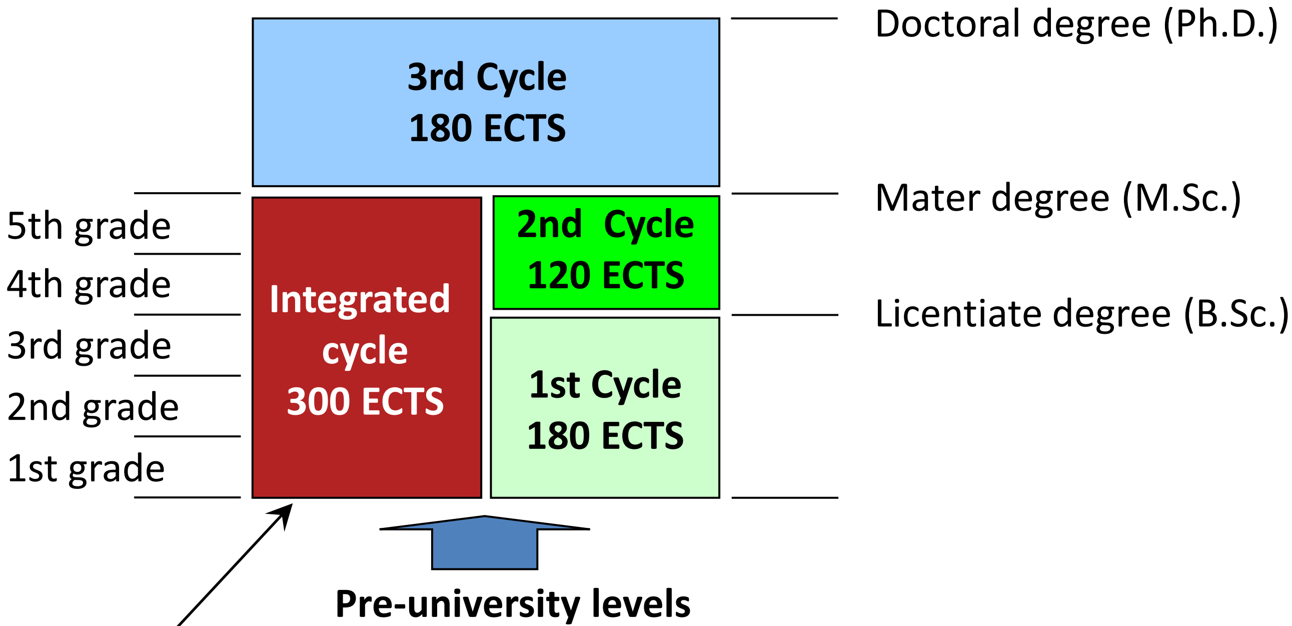
- Employment
  - **36%** had a job arrangement before finishing the degree
  - **80%** employed after 4 months
  - **3,6%** unemployed after 1 year

(2013/14)

Immersion in local industry

- Technology innovators
- Systems design
- Exploitation
- Maintenance
- Management

Our degrees within the Bologna system



<u>MIEEC</u>
Integrated Master
in Electrical and Computer Engineering

MIEEC – Global view of degree structure

5 years

MIEEC – Global view of degree structure

5 years

Entrance:

Number of vacancies: 200

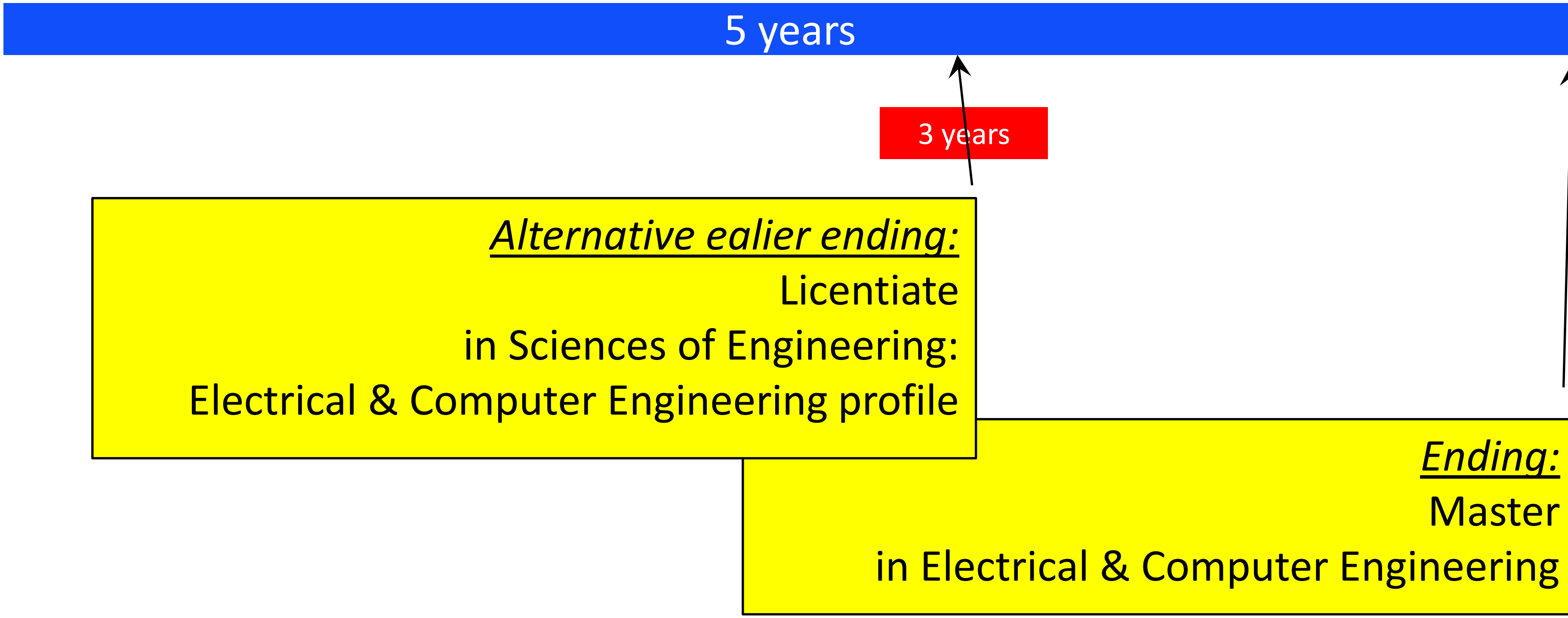
Lowest grade (2014/15): 14,8/20

MIEEC – Global view of degree structure

5 years

<u>Ending:</u>
Master
in Electrical & Computer Engineering

MIEEC – Global view of degree structure



MIEEC – Global view of degree structure

5 years

2 years

Fundamentals

- Mathematics
- Physics
- Programming
- Fundamentals of electricity
- Electronics

Common for all MIEEC students

MIEEC – Global view of degree structure

5 years

2 years	3 years
Fundamentals	Branches

Fundamentals:

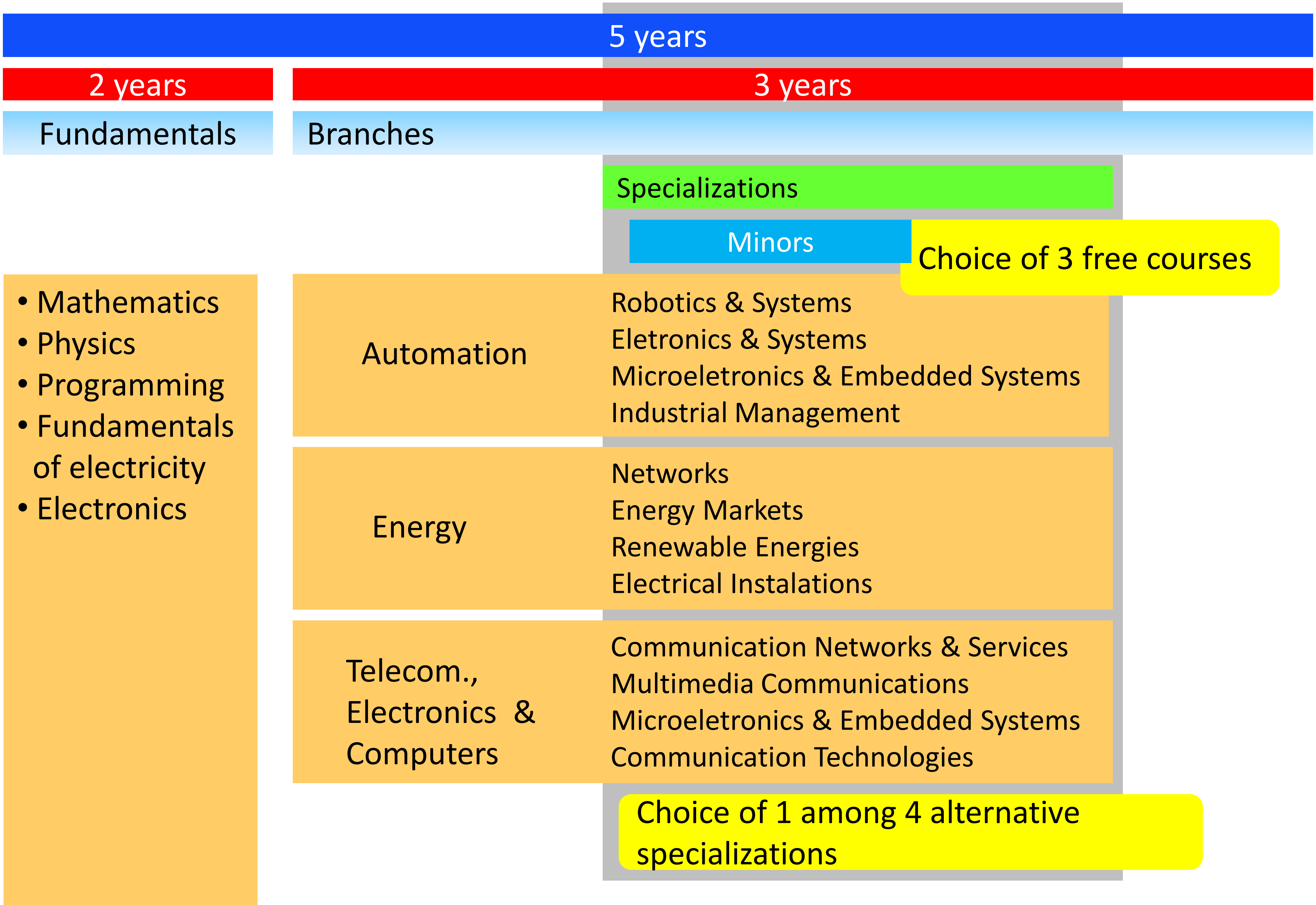
- Mathematics
- Physics
- Programming
- Fundamentals of electricity
- Electronics

Branches:

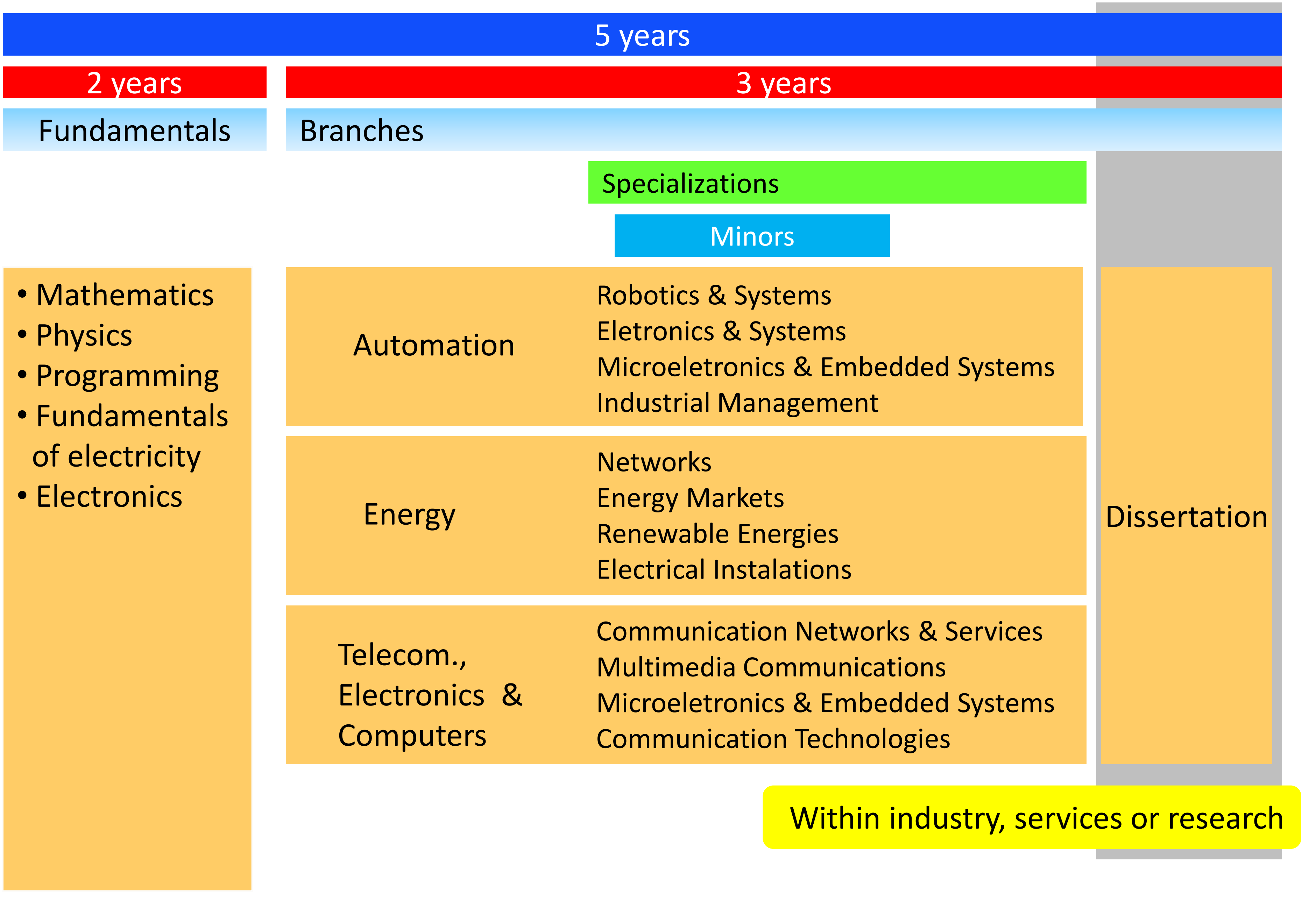
- Automation
- Energy
- Telecommunications, Electronics & Computers

Choice of 1 among 3 alternative branches

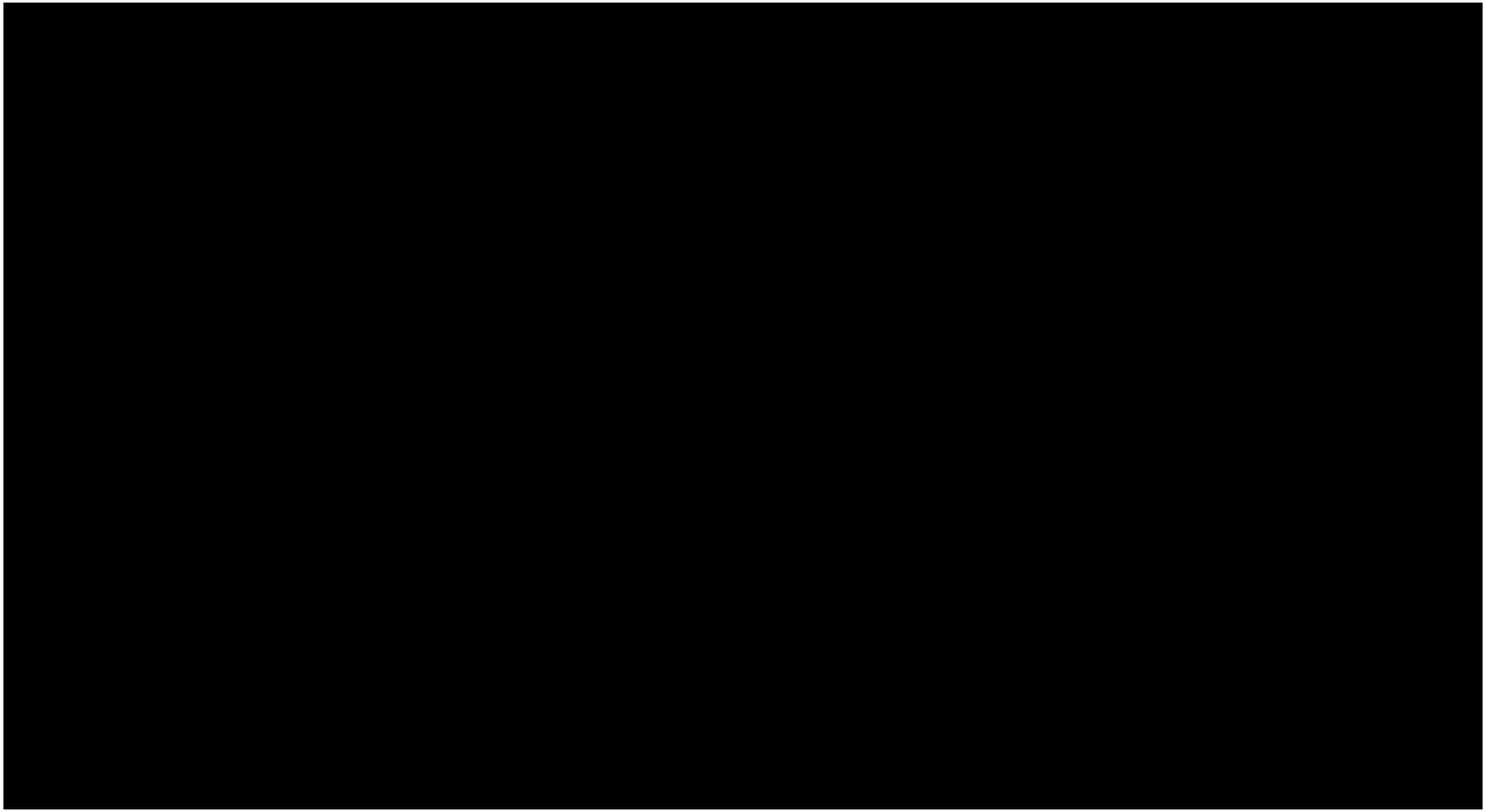
MIEEC – Global view of degree structure



MIEEC – Global view of degree structure



Project-based learning



MIEEC – The Automation branch

Mandatory courses

Systems and Automation (2nd grade)
Automation Technology (3rd grade)
Industrial Informatics (4th grade)
Industrial Communications (4th grade)

MEDIS

MIEEC – The Automation branch

Years: 2, 3, 4

- Systems and Automation
- Automation Technology
- Industrial Informatics
- Industrial Communications

Course outline:

- Discrete control systems
- State machine approach to programming (Grafcet, SFC)
- PLC technology and programming

MIEEC – The Automation branch

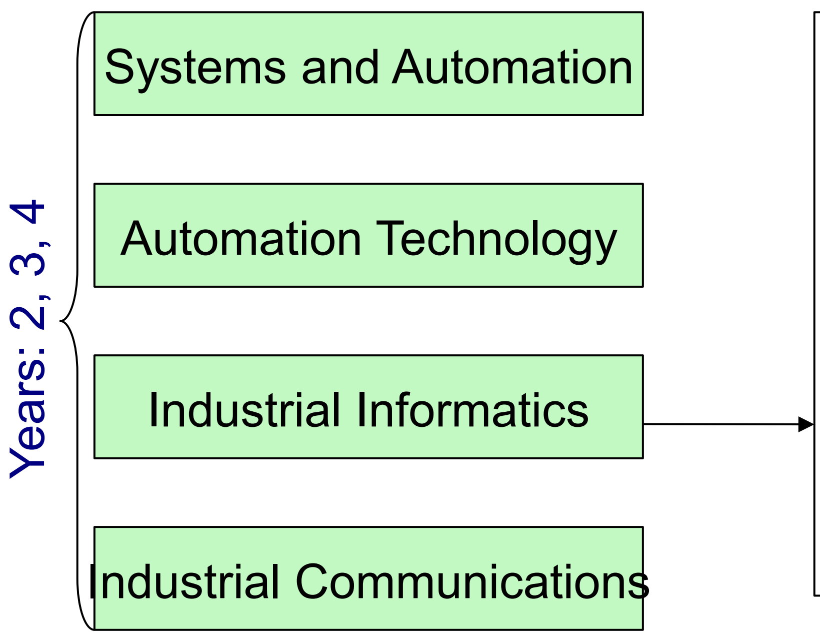
Years: 2, 3, 4

- Systems and Automation
- Automation Technology
- Industrial Informatics
- Industrial Communications

Course outline:

- PLCs programming (IEC 61131-3)
- SoftPLCs
- SCADAS
- Sensors, actuators

MIEEC – The Automation branch



Course outline:

- Object Oriented Design and UML
- Design Patterns
- ISA 88, ISA 95 architectures (e.g. MES, ...)
- Introduction to Distributed Systems
- Introduction to IEC 61499
- OPC overview

MIEEC – The Automation branch

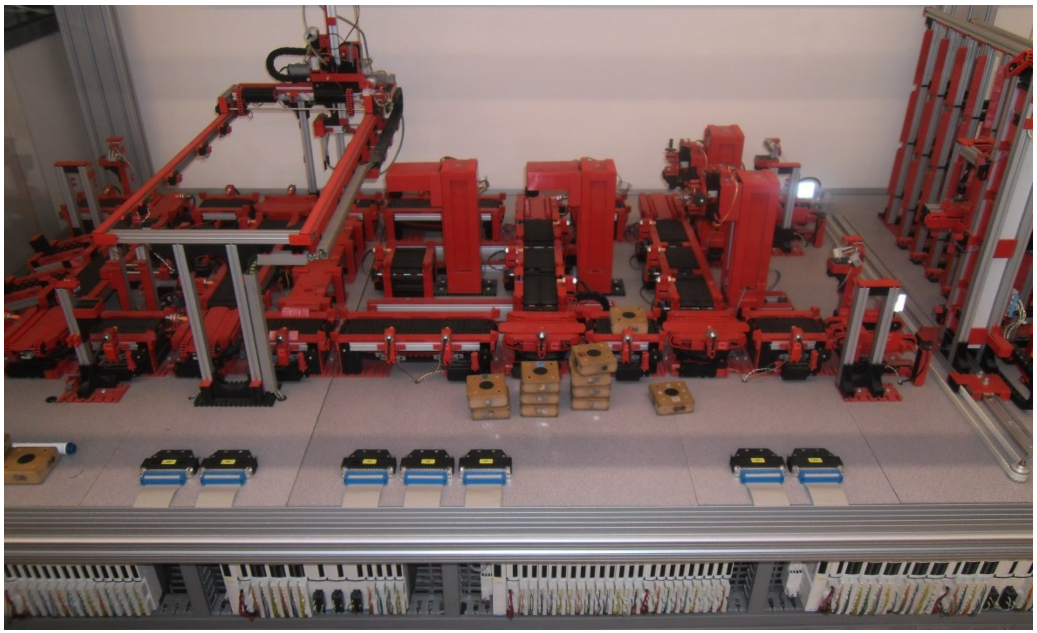
Years: 2, 3, 4

- Systems and Automation
- Automation Technology
- Industrial Informatics
- Industrial Communications

Course outline:

- Industrial Network Requirements
- Layered Stack Model
  (examples for physical, data link, and application layers)
- Distributed Control System Architecture
- Serial Communications (RS232, 485, 422, …)
- Fieldbuses (CAN, CANOpen, Profibus, DeviceNet, …)
- Industrial Ethernet
- Wireless Communications
  (IEEE 802.11 e IEEE 802.15.4)
- TCP/IP Protocol Family
  (IP, ARP, RARP, TCP, UDP, ICMP, IGMP, DNS, FTP, HTTP, ...)

MIEEC – The Automation branch



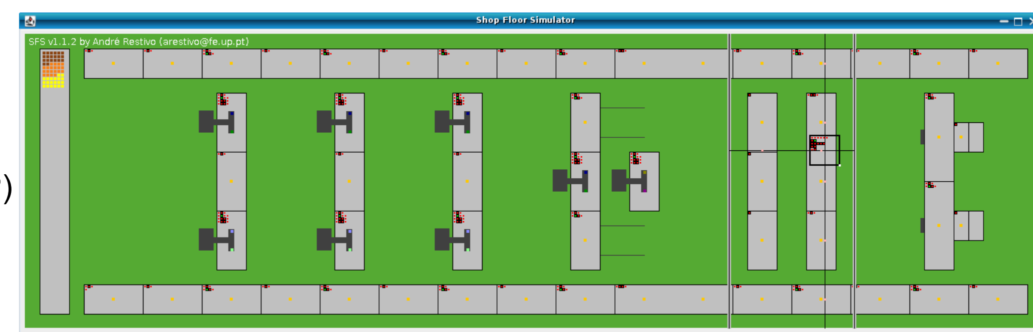
Physical model (kit) for Discrete Control Applications

A manufacturing plant

Java (SW) model of the kit

(both models controlled over **Modbus/TCP**)

(I/O are bit-by-bit **compatible**)



MIEEC – The Automation branch

Physical model (kit) for Discrete Control Applications

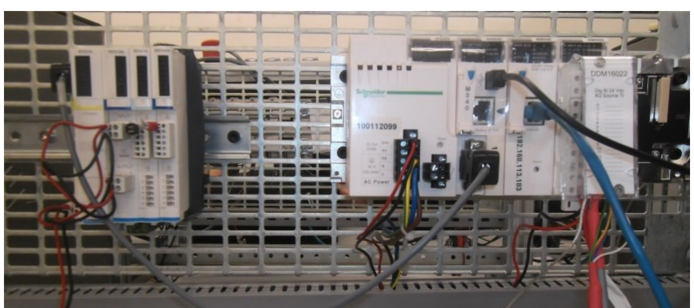
A production line

Physical model (kit) for Continuous Control Applications

Sequence of tanks

MIEEC – The Automation branch

Students’ workbenches



Communication Networks:

- Profibus
- CAN / CANOpen
- DeviceNet
- Ethernet
- Modbus/TCP
- Modbus/RTU + Modbus/ASCII

PLCs:

- Schneider
- Omron
- Siemens

HMI:

- SCADA
- Textual Terminals + Graphical Displays

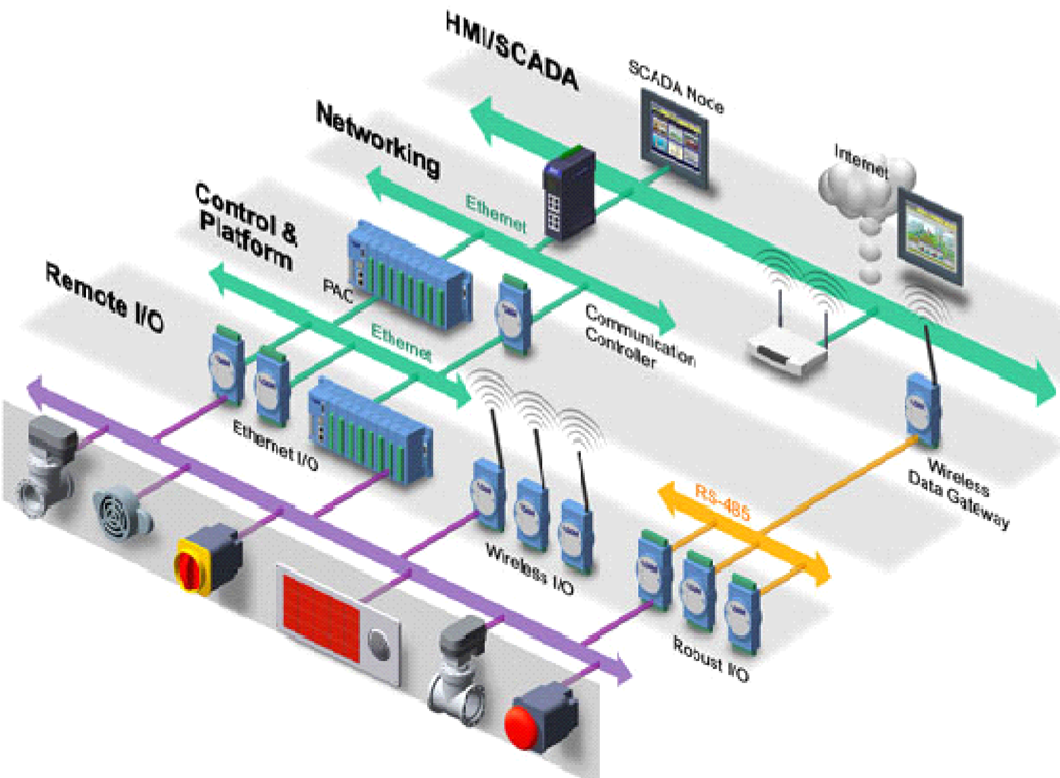
MEDIS – Industrial Networks & Fieldbuses

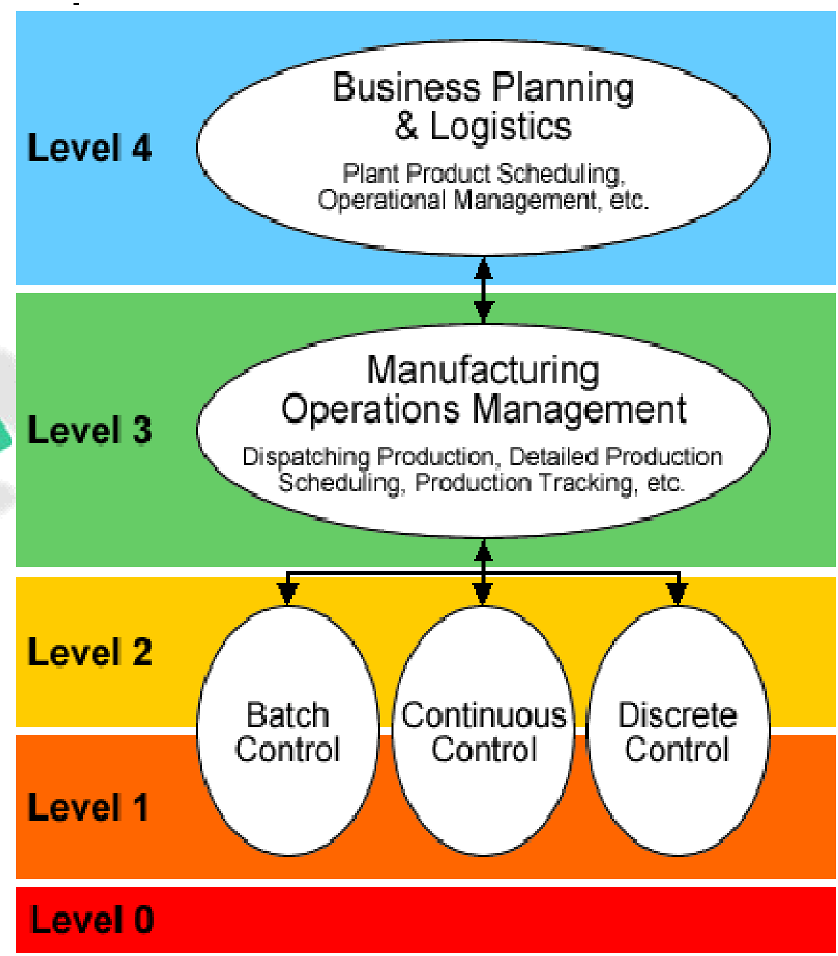
Syllabus

Particular concern in using standard **widely available technologies**

MEDIS – Industrial Networks & Fieldbuses

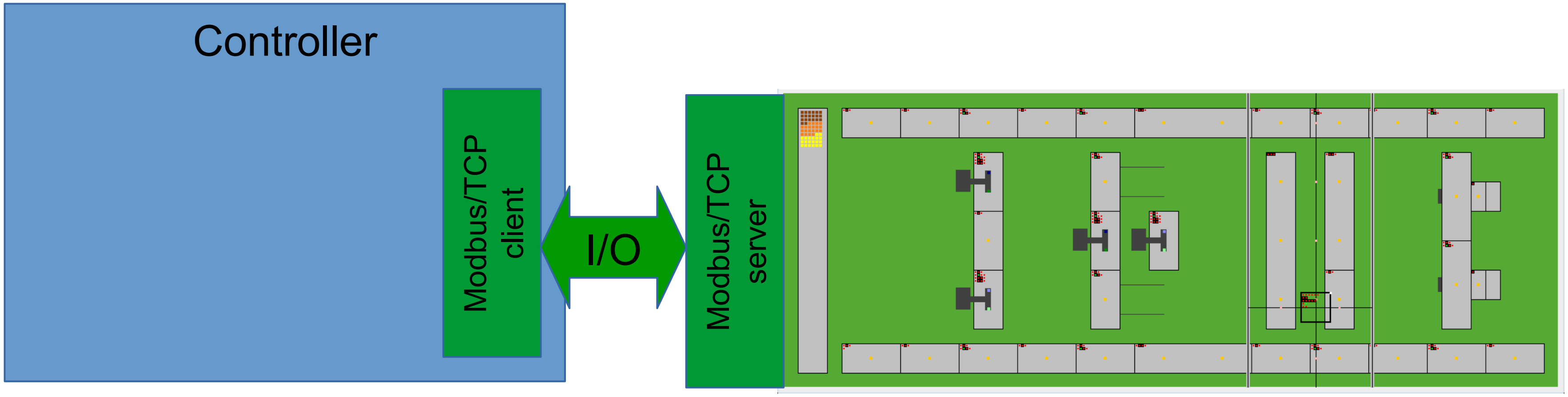
Chapter 1: Introduction





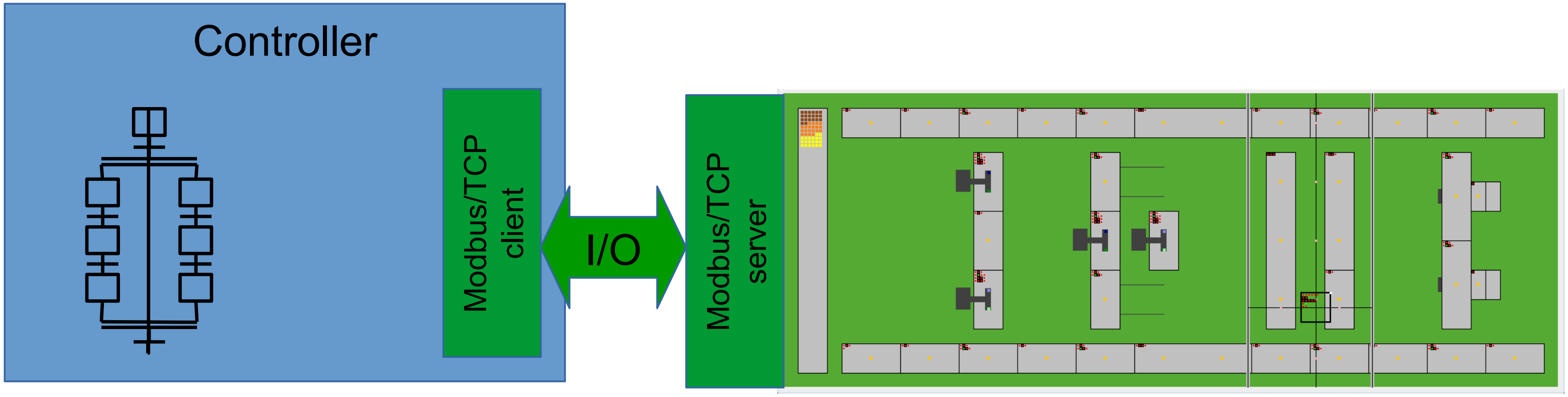
MEDIS – Industrial Networks & Fieldbuses

Chapter 2: Modbus/TCP



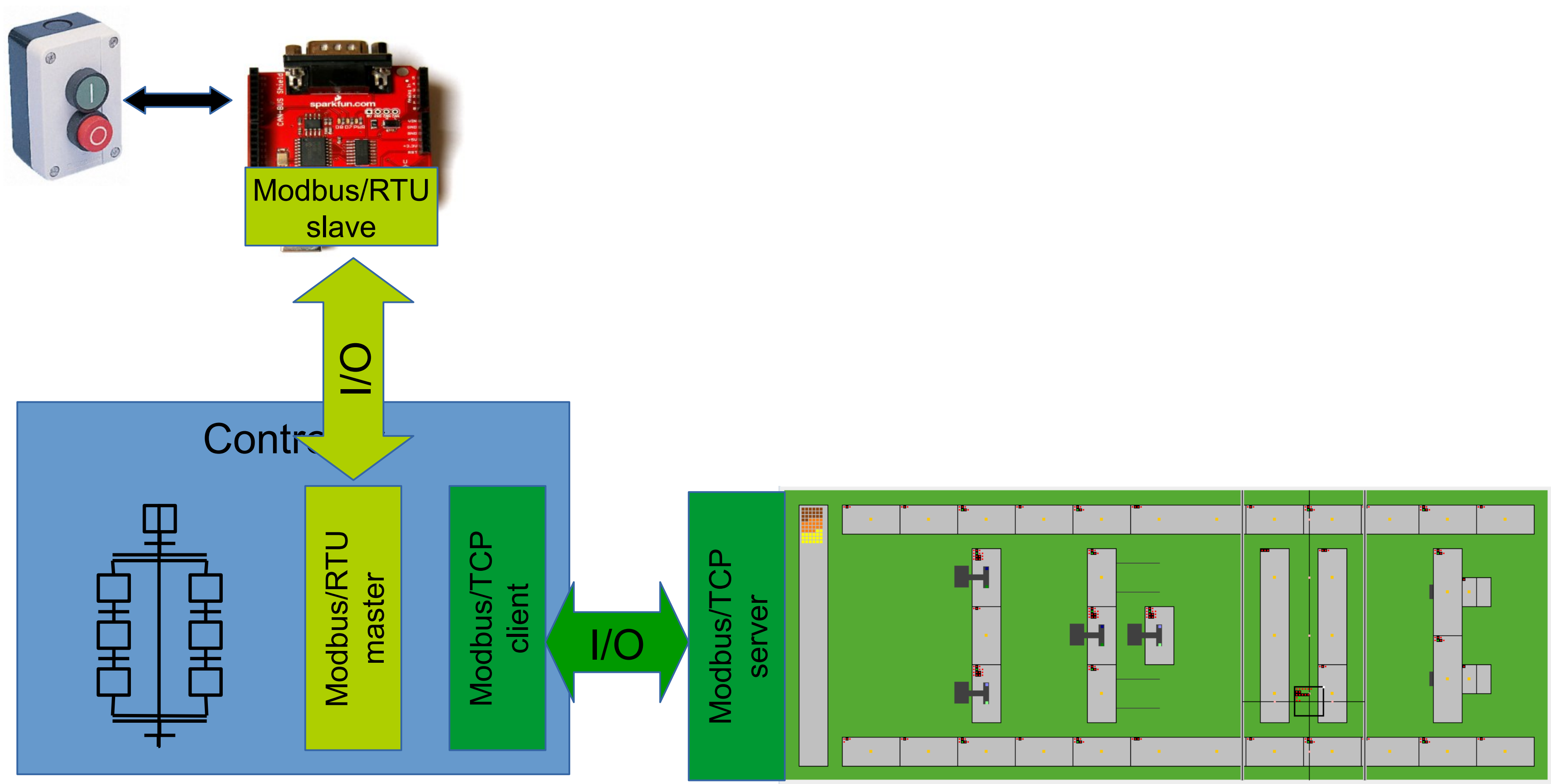
MEDIS – Industrial Networks & Fieldbuses

Chapter 3: Discrete Event Controllers



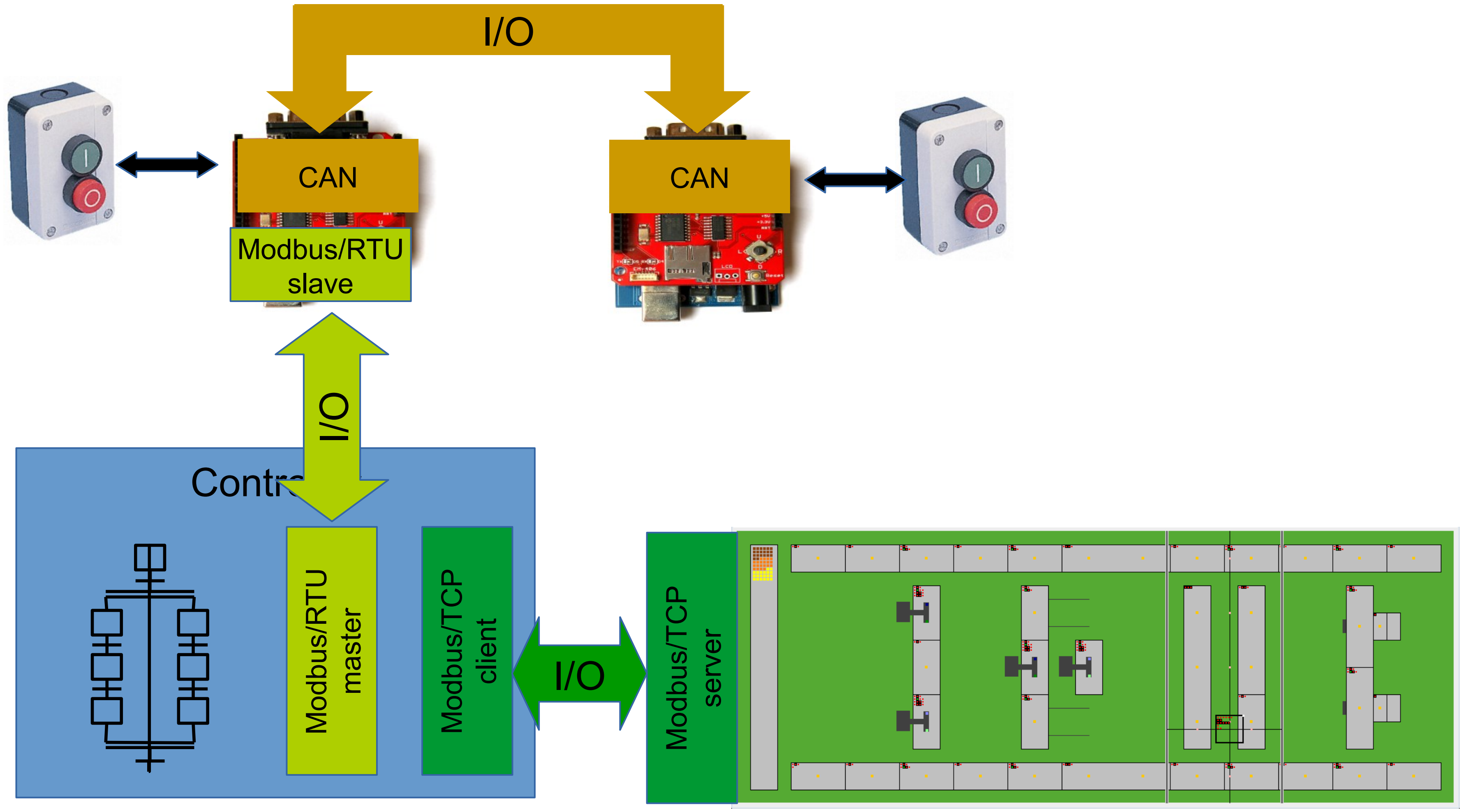
MEDIS – Industrial Networks & Fieldbuses

Chapter 4: Modbus/serial



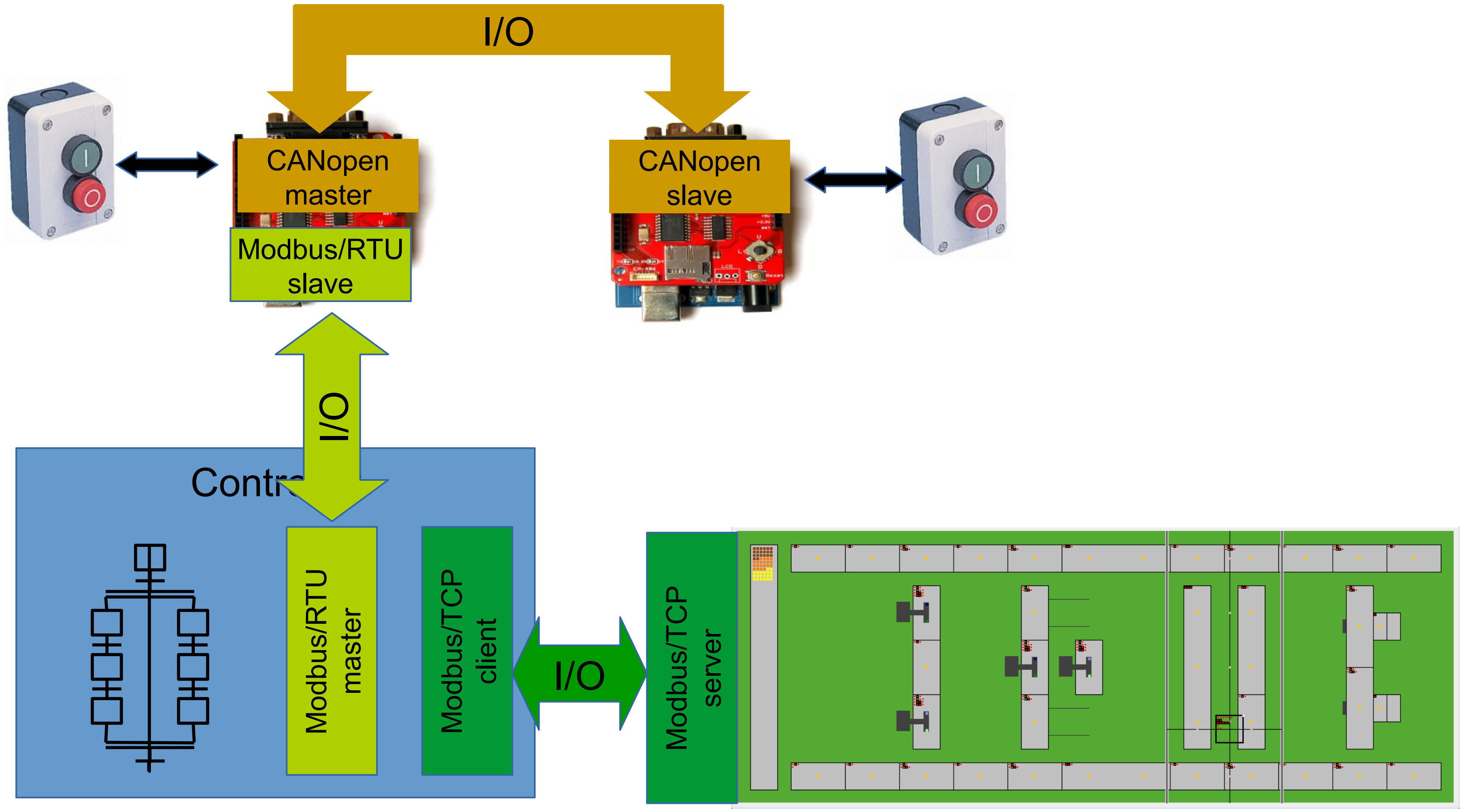
MEDIS – Industrial Networks & Fieldbuses

Chapter 5: CAN



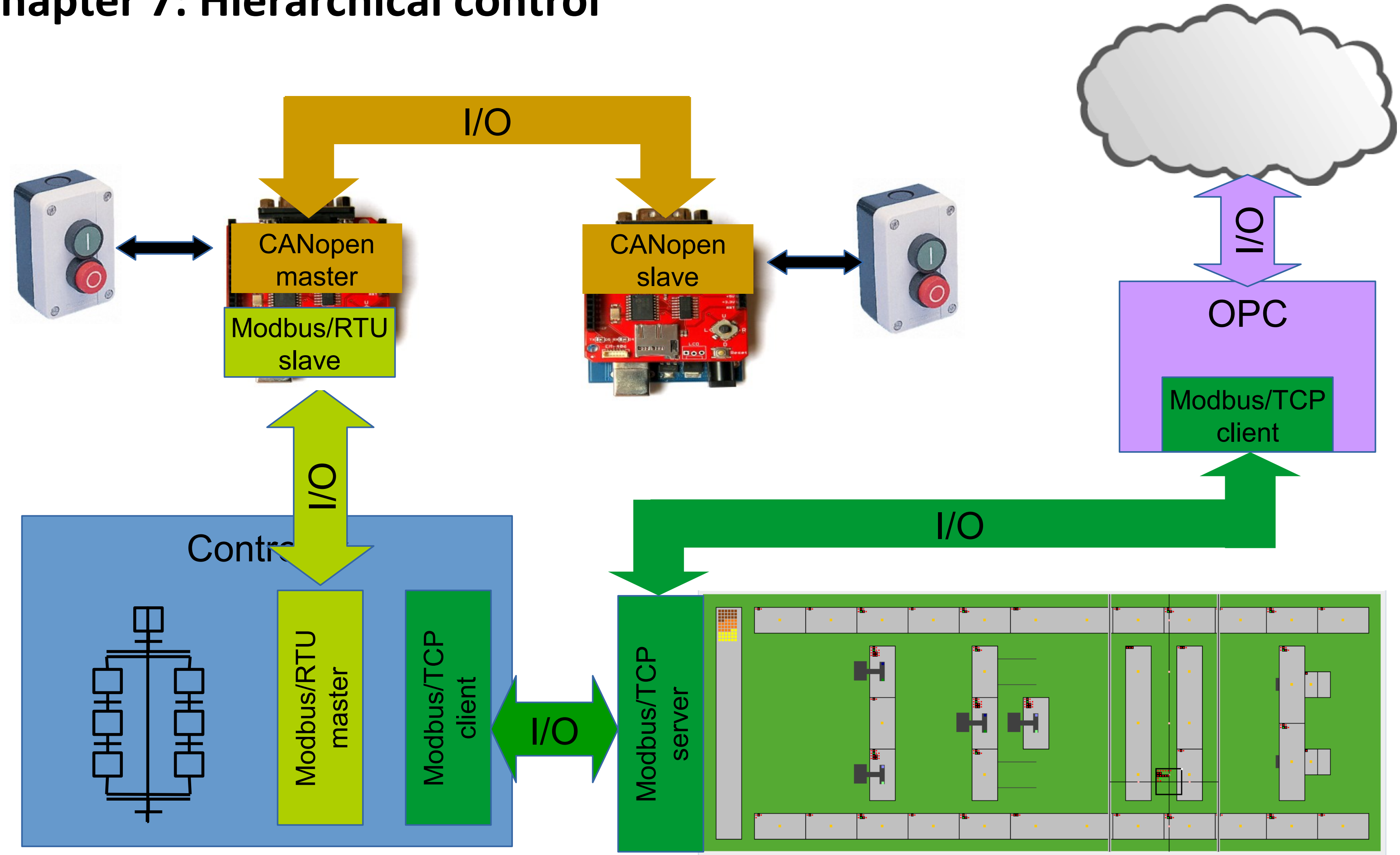
MEDIS – Industrial Networks & Fieldbuses

Chapter 6: CANopen



MEDIS – Industrial Networks & Fieldbuses

Chapter 7: Hierarchical control



MEDIS – the UP (FE) team

Mário de Sousa
- Industrial Informatics
- PLC programming (IEC 61131-3)
- Industrial Communication Networks

(msousa@fe.up.pt)

Luís Almeida
- Industrial Communication Networks
- Real-Time Embedded Systems

(lda@fe.up.pt, fe.up.pt/~lda)

Paulo Portugal
- Industrial Communication Networks
- PLC, SCADA
- Sensors and Actuators

(pportugal@fe.up.pt)

Armando Sousa
- PLC programming
- Robotics
- Artificial Vision Systems

(asousa@fe.up.pt, fe.up.pt/~asousa)

Universität Stuttgart
Institut für Steuerungstechnik
der Werkzeugmaschinen
und Fertigungseinrichtungen

medis

Microcontroller based systems for controlling industrial processes

Teaching

M. Seyfarth, Version 0.1

Contents

Aims of training

Universität Stuttgart

- Detail the structure of the Advanced Industrial Informatics Specialization Modules (AIISM)
- Learn how to organize the AIISM methodology.
- Know the basics of the Problem Based Learning (PBL) methodology to instruct the design and implementation of Industrial Informatic (II) systems to control industrial processes
- The AIISM methodology is designed based on previous experiences of the EU universities on PBL and active learning techniques

Aims of training - PBL

- The purpose is to create a working environment for the students similar to the real environment in companies:
  - Guarantee fundamental knowledge of AIISM as basis for the development of further projects
  - Accustom students to work in teams when solving industrial problems
  - Encourage students to use practical skills to improve their problem solving abilities
  - Develop the ability/skill to adapt to any new computer based systems, due to rapid advances in this area.

Aims of training - PBL

- Other interdisciplinary key qualifications are gained during the course:
  - Teamwork
  - Oral presentation
  - Budget management
  - Project management and documentation
  - Integration and Testing
- The AIISM is structured with different activities of progressive complexity to facilitate the teams to develop their projects along the course.

Contents

Organisational Structure

- Duration: 5 hours time in class, one day per week, 15 weeks; twice the time for preparation, follow-up and mini-project (independent work)
- Learning activities:
  - Lectures
  - Laboratory session
  - Seminars
  - Mini-project
  - Final discussion

Description of learning activities

- Lecture: lecturer presents main ideas of lecture contents and proposes some application problems which students solve individually (~ 1 h).
- Laboratory session: To implement a practical problem previously presented during lecture. Students work in teams of two students (~ 2 h).
- Seminars: a panel discussion with student teams (4 students), students prepare a special topic on their own, present and discuss it (~ 0.75 h).
- Mini-project: dedicated to planning, design and development of the control system of an industrial process. The mini-project is performed by teams of 4 students during ~2 hours and is advanced progressively.

Contents

Lectures - Introduction

- Lecture is the first step in the learning process for each of the topics in each course's module.
- The lecturer presents the main topics of the contents.
- Lectures include some application examples.
- Lectures are interrupted by students interaction:
  - Whispering groups
  - Little questionaires

  → Interrupts should be approx. every 20 minutes, because ability of concentration gets lower.

Lectures - Structure

Universität Stuttgart

During lecture, the teacher:

- PREVIOUSLY: plans the lecture session based on his professional experience.
- Presents the learning objectives of the subject.
- Contextualizes the subject within the module, the course and the career, based on the problems to be solved and the resources that will be proposed to solve these problems.
- Motivates the subject based on the importance of the problems coped by the subject.
- Lists and discusses the bibliographic resources that support the concepts that will be presented during the lecture and that students may use to deepen in their study.

Lectures - Structure

Universität Stuttgart

During lecture, the teacher:

- Presents the key concepts related to the subject, providing the needed details to properly understand them, and specifying the extra available resources and the learning process that students should follow to complete the knowledge's acquisition.
- Follows a logical order for the argumentation, so that students can acquire the knowledge progressively and uses illustrative examples to clarify the presented concepts.
- Verifies that students correctly understand the presented concepts and adapts the speech if needed, allowing students to ask questions about the concepts that are not clear enough and observing the students' answers to the control questions.
- Informs about the skills related to the lecture that will be evaluated.

Lectures - Structure

Universität Stuttgart

During lecture, the teacher:

- Tells students about the practical application of the presented concepts that will be performed later during the seminars, laboratories and mini-project, and which are related to the lecture, encouraging students to study the issue with sufficient interest.
- After the lecture, keeps open communication channels (e.g. consultation hour, forum, email, …) with students.
- LATER: should analyze the lecture session to improve his professional skills (e.g. gather feedback of students)

Lectures - Structure

During lecture, the student:

- PREVIOUSLY: has studied the recommended previous readings.
- Is receptive to the teacher's presentation and is proactive.
- Takes notes to conceptualize what is being exposed to facilitate its further study.
- Asks for clarification of concepts if necessary.
- Answers the questions the teacher addresses to the audience.
- LATER: should follow the learning method proposed by the teacher

Lectures – Aims and Prerequisites

- The students will know:
  - what is the problem and its importance,
  - how the problem is described,
  - which available alternatives would solve it and the criteria to select an appropriate solving alternative even though they will not have yet the skills to apply the proposed methods.
- They will have a studying plan that will be organized in the form of a set of labs, seminars and mini-project.
- Previously to the lecture the student must be informed about the prerequisites to successfully follow the lecture.
- If the prerequisites aren't satisfied before the lecture a list of additional courses to visit or literature lists should be supplied.

Lectures – Recommendations

- During the lecture all the conversations in the classroom should be public, addressed to all of the members.
- The sequence of the lecture should be planned and predictable.
- The dynamics of the lecture should be attractive to the audience with short and direct explanations.
- The lecture shouldn't explain all the details of the concepts of the subject, it should give enough hints to let students autonomously complete the knowledge acquisition.
- The students must have the chance to formulate questions.
- Behavior rules must be defined to let the lecture to be productive and the corrective mechanisms should be effective and proportional.
- Potential distractors of the audience attention should be avoided or minimized.

Contents

Laboratory Session – Introduction

The laboratory session

- Is the first practical exercise that students take to acquire a basic set of skills related to the topic presented in the lecture.
- adresses and solves specific and well-defined problems
- is guided, fully documented, and in progressively increasing complexity
- provides students with a set of tools and skills that can be used to solve more open problems during the mini-project.

Laboratory Session - Structure

During laboratory session, the teacher:

- PREVIOUSLY: plans the lab session based on his professional experience.
- Presents the learning goals of the lab.
- Contextualizes the lab within the subject.
- Motivates the practical exercise to be performed during the lab based on the importance of the problem it addresses.
- Lists and comments the equipment, the material and documentation resources needed to perform the lab.
- Describes the correct utilization of the lab equipment and warns about potential material and personal damage due to inappropriate use.
- Answers students questions during the practical exercise.
- LATER: should analyze the lab session to improve his professional skills.

Laboratory Session - Structure

Universität Stuttgart

During laboratory session, the student:

- PREVIOUSLY: has studied the lab documentation, and has attended the related lectures.
- Is receptive to the teacher's indications and is proactive.
- Takes notes to remember the indications.
- Asks for clarification of concepts if necessary.
- Works in teams of two students on the practical exercises of the lab.
- Answers the questions of the teacher related to the exercise.
- LATER: should review and document the results of the practical exercises and eventually perform the extra optional exercises.

Laboratory Session - Structure

During laboratory session, the technical assistant:

- PREVIOUSLY: Prepares the necessary equipment (software and hardware) for the lab in each of the workbenches based on the teacher's requests and his professional experience.
- Helps solving problems that could arise related to the equipment, power supply, communications and software
- Makes diagnosis about the safe and correct operation of the equipment
- Replaces damaged or missing components.
- LATER: should analyze the lab session to improve his professional skills.

Laboratory Session – Aims and Prerequisites

- The students will:
  - have acquired the skills to solve basic problems.
  - transfer their knowledge to new problems, that will be useful in the next related seminars and the mini-project.
- The students should have attended the related lecture and have read the recommended further lectures and the lab guide.
- Students should have grouped to teams of 2 students, working together.

Laboratory Session – Recommendations

- The lab should be fully documented.
- The lab guide should start with an introduction that remarks to the concepts of the lecture that are going to be applied in the practical exercise, the goals definition and a list of the material and documental resources that will be needed.
- The guide should continue with a definition of the practical activities in the following phases of progressive and increasing of complexity: introductory, reinforcement, advanced and optional.
- The first exercise in the introductory phase should be described step by step.
- The second exercise in the reinforcement should practice the same concepts and method than before but on a different set of problem data and without the help of the guide

Laboratory Session – Recommendations

- The third exercise in the advanced phase should practice the application of the previously acquired tools, protocols and skills to solve a small sized and small complexity application problem.
- A fourth exercise should be defined in an optional final phase to let advanced students to consider further technical questions related to the topic of the lab.
- The teacher should supervise the practical exercises of the students, answering their questions, guiding them and providing enough hints to let the students find solutions by them shelf.
- Repeated errors and problems, and interesting software designs of the students during the lab should be shared with the whole group. Remote desktop sessions of the workbench computer screen could be presented on the beamer.

Contents

Seminars – Introduction

The seminars support

- Problem solving on the topic of the lecture. Students have already basic knowledge on related tools and procedures.
- Acquisition of additional knowledge. Preparation and investigation on new, well defined topics.
- Supplementation of lecture topics.
- Teamwork as students work in groups of 4 students, but they must explain and share their experiences with the whole group during the seminar meetings.

Seminars - Structure

- The group of students, responsible for the seminar, gets detailed information by the teacher
- The group of students prepares a structured presentation and a handout on the result of their investigations
- The seminar starts with an introduction of the teacher, remembering about the key concepts presented during the lecture, and the proposition of several related design problems.
- The group of students presents their results
- Some of the problems are proposed to the whole group to start a discussion
- The seminar ends with a resume by the teacher.

Seminars – Aims and Prerequisites

- The students will:
  - practice the concepts presented in the related lectures and verify that these concepts have been assimilated correctly.
  - relate the previous concepts with other technical concepts that usually are studied in different subjects, in a contextualized way.
  - acquire team-working skills.
  - acquire documentation and presentation skills.
  - acquire critical searching of information skills.
- The students should have attended the related lecture and have read the further recommended readings.
- Working teams of four students should have been set. A balanced team is recommended with a similar level of initial knowledge for each of its members. One of the students will act as a speaker of the team.

Contents

Mini-Project – Introduction

The mini-project

- Students apply the knowledge and skills that they have acquired in the lectures, labs and seminars to develop the controller system of an industrial process in an integral, interdisciplinary way
- The problem of the mini-project is the highest complexity problem in the module.
- The students work in teams of 4 students (same teams as in the seminars are proposed, but not mandatory).
- The teams manage their project by their own (planning of time and resources).
- The teams can use components of other teams or results from the labs in their own mini-project, if other teams share them.

Mini-Project – Aims and Prerequisites

- The students will
  - be able to integrate the tools and protocols practiced during the course to develop simple complexity and medium sized applications to control industrial processes.
  - be able to document and present the mini-project process and results.
  - Be able to plan a project of higher complexity
  - Be able to work in a team and share workload and resources
- The students should have attended the lectures and completed the practical exercises of the labs and seminars that are related to the module.

Contents

Aims of the module

- Understand Architecture and Programming of Microcontrollers
- Understand Basics of Industrial Processes
  - Sensors and Actuators
  - Measurement and Control
- Deepen Knowledge of Functions and Programming of Microcontrollers
- Use Project Management Methods

Contents of the module

Universität Stuttgart

1. Introduction to microcontrollers and process control
2. Project management and project planning
3. Input-/Output system of microcontrollers
4. Timer and interrupt functions on microcontroller systems
5. Graphic systems for microcontrollers
6. Communication systems on microcontrollers
7. Implementation of Control methods on microcontrollers
8. Integration and validation

Examples for lectures

- Lecture
  - Power-Point-Presentations for lectures available (see example)

Examples for laboratory session

- Laboratory Session
  - Work orders for different topics of the module available as PPT-presentations (see example)
  - Based on Microcontroller Arduino DUE and the corresponding Integrated Development Environment (open source)
  - Programmed in programming language „C“ (C++ object orientation is possible)
  - Use of a Hardware-in-the-loop Simulation-system, integrated on a compact test rig.

Examples for laboratory session: test rig

Adapter board for converting and filtering signals

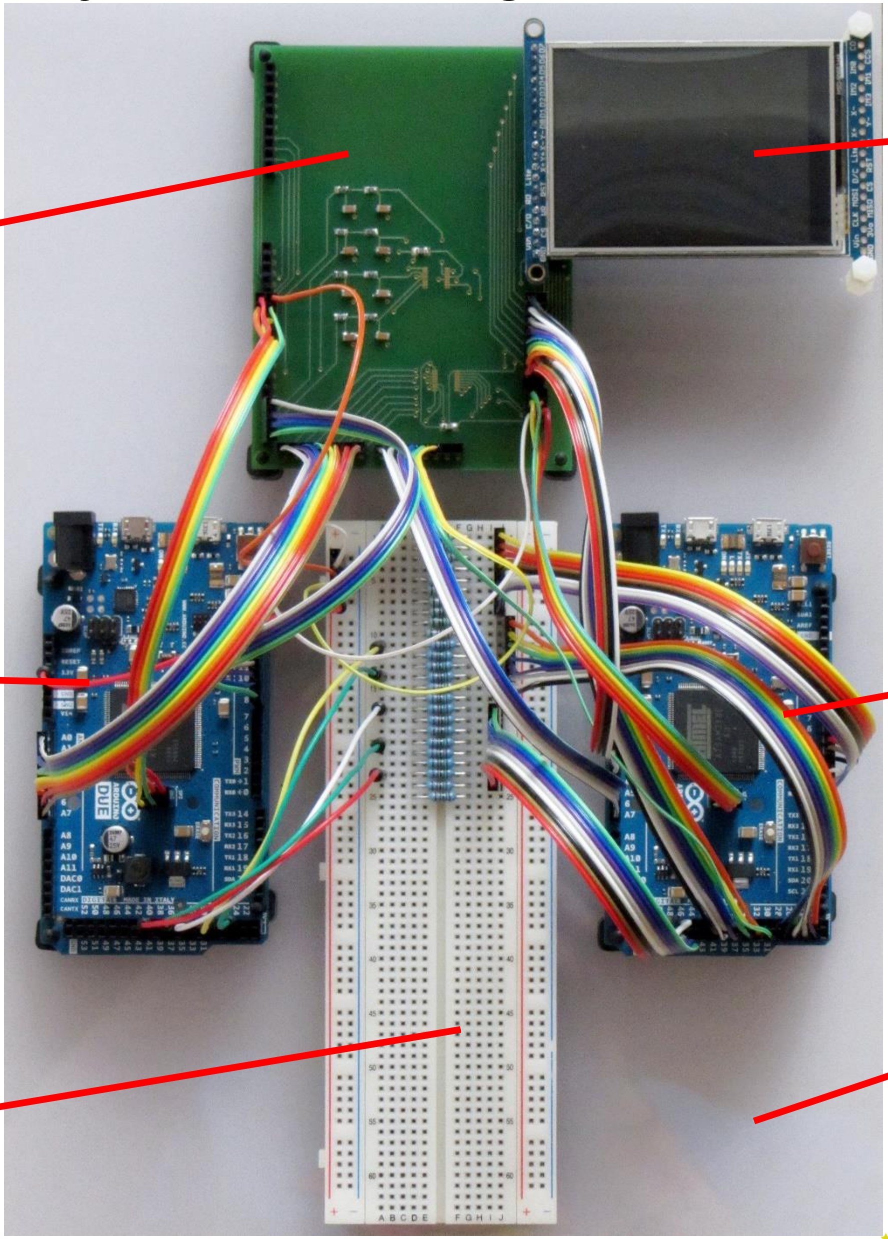
LCD display

Microcontroller for Control Application

Microcontroller for Simulation System

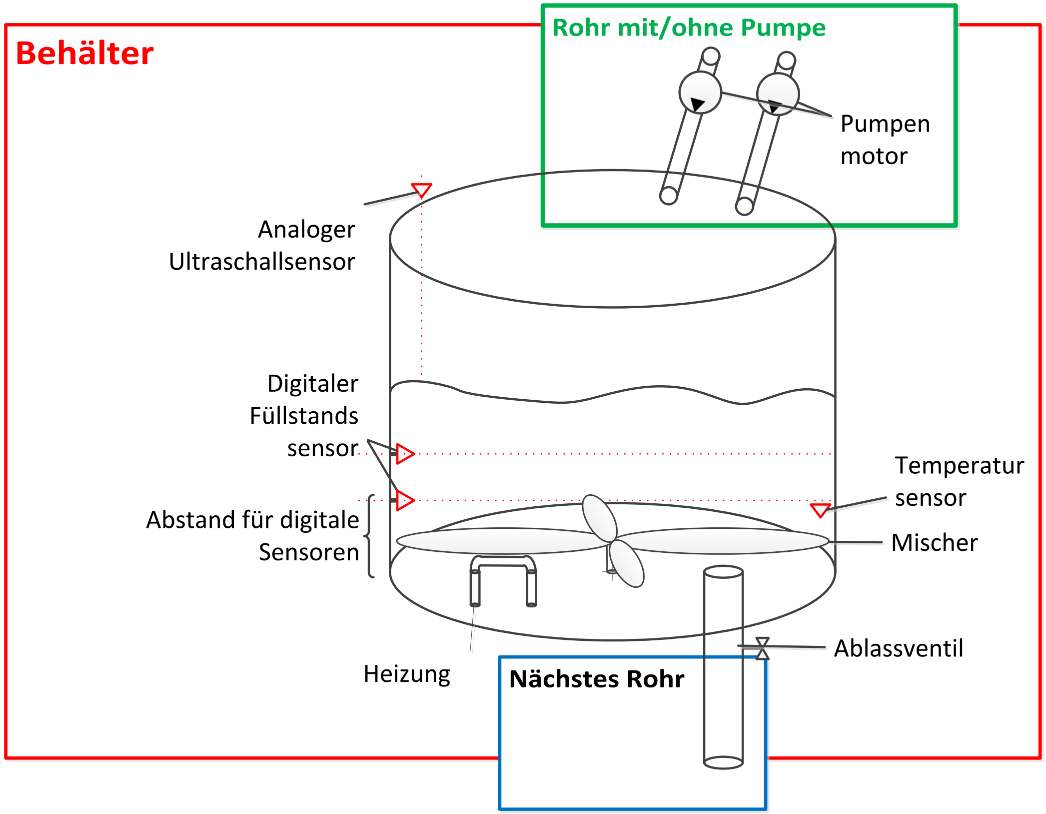
Breadboard for individual wiring and additional components

Plexi glass disk (200x300 mm)



Examples for laboratory session: simulation of fluid tank

- Simulation of a fluid tank system
  - Configurable scenarios: different number of tanks, different sensors and actuators
  - Pumps and valves and branches between tanks
  - Heater and Mixer
  - Sensors for measuring filling level and temperature



Examples for seminars

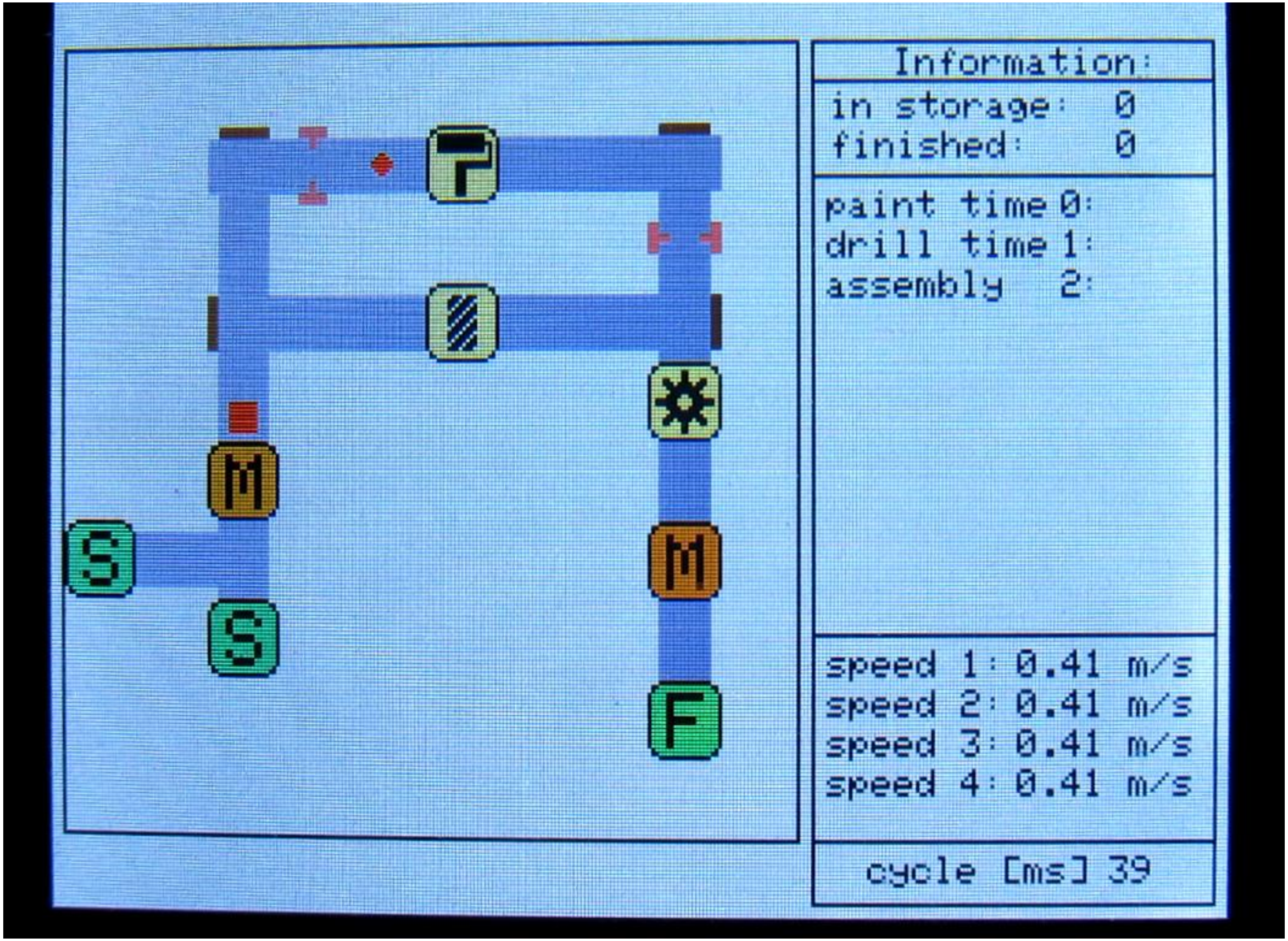
- Seminars
  - Different seminar topics for different phases of the module:
    - Overview on microcontrollers
    - Basics in C programming
    - Methods of project management, project documentation, testing and validation strategies
    - Use of libraries within microcontrollers
  - Student group (4 students) gets topic in advance and prepares a presentation and handout (approx. 1 week lead time)

Example for mini-project

Universität Stuttgart

- Mini-Project
  - Based on same simulation system as laboratory session
  - Student groups (4 students) get a complex control problem for an industrial process
  - Students do project management, scheduling, conception, design, implementation, testing, documentation, …
  - Scenario: production system, to produce letterpresses. System of stocks, machines, conveyors, sensors must be configured and programmed
  - Specification document available
  - Progress of work during the whole course.

Example for mini-project: possible result



Contents

Introduction - Overview

- Required resources
  - Classroom-laboratory
  - Personal computers (one computer each two students)
  - Microcontroller boards: control microcontroller and simulation microcontroller (one combination each two students)
  - Software tools (mostly open source, except operating system for PC)
  - Staff

Classroom-laboratory

- Best: a fully equipped classroom with laboratory equipment
  - No change in room from lecture to laboratory session
    → Avoid differentiating between theory and laboratory practive
    → Provides flexibility in session development

Classroom-laboratory: Required equipment

- Lectures and Seminars
  - Board
  - Laptop/Personal Computer for teacher
  - Digital projector (Beamer)
  - Office Applications
  - Development Software
  - Internet connection
  - Optional: samples of real devices (sensors, actuators, …)

Classroom-laboratory: Required equipment

- Laboratory Session and Mini-Project
  - Laptop/Personal Computer for teacher
  - Digital projector (Beamer)
  - Personal Computer each two students
  - Test rig: Microcontroller Boards (2x), Displays (2x), Breadboard each two students
  - DC power supply
  - Wires, Push buttons, resistors, LEDs, USB cables
  - Oscilloscope, Multimeter
  - Development Environmet for implementing applications programming microcontrollers
  - Simulation software for tank model and production system
  - Internet connection

Human resources

- Teachers
  - Give the lectures
  - Guide through the course
  - Answer questions regarding teaching structure, synchronization of learning sessions (lectures, laboratory session, seminars)
  - Answer questions to specific contents of the course and seminars
  - Evaluate the students

Human resources

- Technicians
  - Support laboratory session
  - Set-up all hardware and software tools
  - Answer questions regarding installation and configuration of software and hardware
  - Repair or replace damaged or missing material

Human resources

- Administrative
  - Help to translate EU grading system to PC grading system
  - Help to transfer the ECTS credits into the PC credit system
  - Support organisational integration of course in module structure at PC

Contents

Universität Stuttgart

Aim: determine the level of assimilation of knowledge and skills by the students

- Evaluation criteria
  - Technical knowledge
  - Cognitive skills (analysis, synthesis, application, evaluation, critique)
  - Soft skills (organizing time and resources, coordination, negotiating, tolerating, teamwork)

  → Students must acquire a certain appropriate level, must proove their capabilities
- Large amount of information can be collected during the whole course

Evaluation: Formation of Teams

- Teams of 2 or 3 students.
- Groups should remain invariant throughout the course.
- Students who form a team have a similar level of knowledge.

→ Classification of different levels of students by a short test (about 25 multiple choice questions about e.g. microprocessor- and/or programming knowledge)

→ Questions should collect general aspects of the concepts required as prerequisites, e.g. Basic Engineering concepts, Programming (Software)

Level One: Attitude / Engagement

- Motivation that students have within the course.
- Teachers keep ongoing dialogue with different student groups throughout the entire course.
- Special attention to delivery of activities in time (deadlines) and manner
- This part can be 10% of the final overall grade.

- Identify those students with a special motivation for the subject because their attitude is above average
- It is important that students meet requirements of work deadlines in the subject. This will facilitate the professional future of the graduates in order to work in environments with strict deadlines.

Student Evaluation Methodolgy

Level Two: Learning

- Level of knowledge and skills that students have acquired throughout the course
- Lecturer presents main ideas of lecture contents and proposes some application problems which students solve individually.
- The teachers throughout the course individually evaluate how the student has solved different problems.
- Watch students during laboratory session: can they solve the work orders
- Ask questions about the solution the students propose
- Rate the quality and completeness of the students solution
  - Introduction phase: will reflect 20% of the grade.
  - Reinforcement Phase: will reflect 40% of the grade.
  - Advanced stage: Will represent a 40% of the grade.

Level Two: Learning (in Seminars)

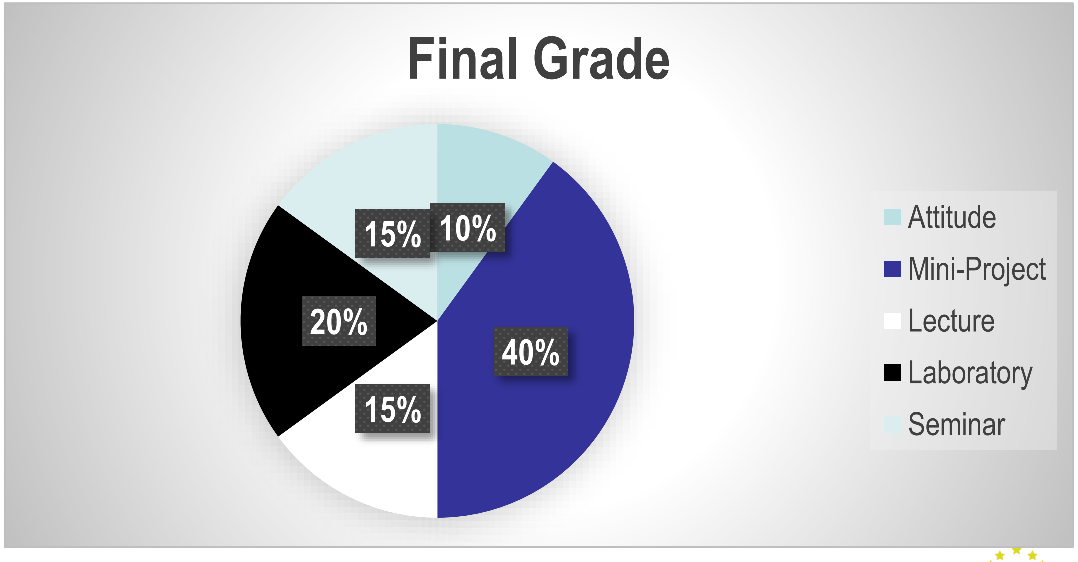
- It is important that the teacher talks with and asks all members of the group to identify how well each student attended and has acquired the relevant knowledge.
- Some aspects to be taken into account for the assessment:
  - Level of responsibility among group members.
  - Number of studied and analysed possibilities for a solution in terms of advantages and disadvantages
  - Quality of technical report writing
  - Defense of ideas and how to defend against constructive criticism
  - Interaction of knowledge of this area with other
  - Management of bibliographic sources
  - Extra work done with respect to the requested
  - Robustness of the proposed solution adopted
  - …

Level Two: Learning (in Mini-Project)

- Quality of planning, design and development of the control system for the real world problem.
- The developed project will be presented publicly to a jury composed of three lecturers.
- The jury evaluates the following aspects including some interdisciplinary skills:
  - Report: Maximum score of this part is 25% of the total mark.
  - Oral presentation: The score of this part is 10% maximum. The team presents the work during maximum of 20 minutes.
  - Implementation: The score assigned to this part is 65% of total mark. After the oral presentation, the team shows the project application.

Level Three: Grading (Final result)

- Collect all grades earned during the course.
- Proposed calculation of final grade:
  - The evaluation of the students attitude (A): 10% of the final grade.
  - The evaluation of the Mini-Project (MP): 40% of the final grade.
  - The evaluation of the Lecture (L): 15% of the final score.
  - The evaluation of the Laboratory (P): 20% of the final score.
  - The evaluation of the Seminars (S): 15% of the final score.



Evaluation of the course design

- Evaluation of the methodology used for teaching this course.
- Important to know the opinion of students and teachers involved to discover what has been done well and what parts could be improved
- Two points of view:
  - Students point of view
  - Lecturers point of view

Evaluation of the course design

Students point of view

- Survey among students who attended the course to have information about the acceptance of the course.
- Students can give their opinion at the end of the course and before obtaining their qualifications
- Design a survey in a way that can be simple and easy to answer. E.g. based on few questions with 5 possible answers (“A”: Strongly Agree; “B”: Agree; “C”: Unsure; “D”: Disagree; “E”: Strongly Disagree).
- Possible questions:
  - I had enough time to ask questions
  - I think the lecturer was well prepared
  - I think the work orders of the laboratory sessions were well balanced
  - I think the overall organization of the course was very good
  - ...

Evaluation of the course design

Students point of view

- Possible questions (continued):
  - The methodology supported my learning process
  - Every important concept of the subject has been addressed in the miniproject
  - The complexity level of every part of the subject has been reasonable
  - I have felt motivated during the learning process
  - I can recommend this course to other students?
  - …

Evaluation of the course design

Lecturers point of view

- The opinion of teachers is important to make an overall evaluation of how the course has worked and what aspects should be improved.
- Teachers should maintain an open dialogue throughout the course and continuously discuss their experience.
- Possible aspects to be evaluated:
  - Number of approved students
  - Quality of ratings
  - Amount of work done by teachers
  - Problems that have arisen and how they have been resolved
  - Possible updating of content
  - Duplication and overlap with other subjects
  - …

Many thanks for your attention

Dipl.-Ing. Michael Seyfarth
Institute for Control Engineering of Machine Tools
michael.seyfarth@isw.uni-Stuttgart.de

70 YEARS Technical University of Sofia
We succeed!

medis

Introduction

Introduction:

- To detail the structure of the Advanced Industrial Informatics Specialization Modules (AIISM)
- Learn how to organize the AIISM methodology.
- The AIISM course uses a PBL (Problem Based Learning) methodology to instruct the design and implementation of II systems to control industrial processes
- The AIISM methodology is designed based on previous experiences of the EU universities on PBL and active learning techniques
- The purpose is to create a working environment for the students similar to the real environment in companies:
  - To guarantee fundamental knowledge of AIISM as basis for the development of further objectives
  - To accustom students work in teams when solving industrial problems
  - To encourage students to use practical skills to improve their problem solving abilities
  - To develop the capacity to adapt to any new computer based systems, due to rapid advances in this area.

- Other engineering transversal skills are gained during the course:
  - Teamwork
  - Technical competencies
  - Oral presentation
  - Budget management
  - Report redaction
  - Etc.

- The AIISM is structured with different activities of progressive complexity to facilitate the teams to develop their projects along the course.

- The activities are developed during 5 hours/day, one day of the week through a PBL methodology, using as case study the examples of several objects, their HIL simulators and corresponding control systems.

Learning activities

Learning activities description:

- Students have to apply the knowledge acquired from the lectures and the laboratory practices.
- The proposed learning activities are the following:
  - **Lecture and problems**: lecturer presents main ideas of lecture contents and proposes some application problems which student solves individually (e. g., 1 h).
  - **Laboratory session**: To implement (1 h 15”) a practical problem previously presented during lecture. Students work by teams of two students.
  - **Seminars**: a panel discussion with student teams (4 students) lasting 45 minutes is proposed, consisting generally of solving a problem by means of PBL.
  - **Mini-project**: dedicated to planning, design and development of the control system of the educational liquids tank. The mini-project is performed by teams of 4 students during 2 hours. Weekly, the mini-project is advanced progressively.

Lectures

Lectures: Introduction

- Lecture is the first step in the learning process for each of the topics in each course’s module.
- The lecturer presents the main topics of the theme contents.
- It includes some application examples.

Lectures: Structure

During lecture, **the teacher**:

- PREVIOUSLY: plans the lecture session based on their professional experience.
- Presents the learning objectives of the subject.
- Contextualizes the subject within the module, the course and the career, based on the problems to be solved and the resources that will be proposed to solve these problems.
- Motivates the subject based on the importance of the problems coped by the subject.
- Lists and discusses the bibliographic resources that support the concepts that will be presented during the lecture and that students may use to deepen in their study.
- Presents the key concepts related to the subject, providing the needed details to properly understand them, and specifying the extra available resources and the learning process that students should follow to complete the knowledge's acquisition. Follows a logical order for the argumentation, so that students can acquire the knowledge progressively and uses illustrative examples to clarify the presented concepts.

Lectures: Structure (Cont.)

- Verifies that students correctly understand the presented concepts and adapts the speech if needed, allowing students to ask questions about the concepts that are not clear enough and observing the students' answers to the control questions.
- Informs about the skills related to the lecture that will be evaluated.
- Tells students about the practical application of the presented concepts that will be performed later during the seminars, laboratories and mini-project, and which are related to the lecture, encouraging students to study the issue with sufficient interest.
- After the lecture, keeps open communication channels with students, so they can make consultations before next lecture if necessary.
- LATER: should analyze the lecture session to improve their professional skills.

Lectures: Structure (Cont.)

During lecture, **the student**:

- PREVIOUSLY: has studied the recommended previous readings.
- Is receptive to the teacher's presentation and is proactive.
- Takes notes to conceptualize what is being exposed to facilitate its further study.
- Asks for concepts' clarification if necessary.
- Answers the questions the teacher addresses to the audience.
- LATER: should follow the learning method proposed by the teacher

Lectures: Goals

- After the lecture the students will know what is the problem and its importance, how the problem is described, which available alternatives would solve it and the criteria to select an appropriate solving alternative even though they will not have yet the skills to apply the proposed methods.

- They will have a studying plan that will be organized in the form of a set of labs, seminars and mini-project.

Lectures: Prerequisites

- Previously to the lecture the student must be informed about the prerequisites to successfully follow the lecture.

- If the prerequisites aren't satisfied a previous to the lecture formation plan should be supplied.

Lectures: Example

"Process Interface"

In this lecture the signal exchange between the controller computer and the simulator one will be discussed and explained.

- Goal:
  To learn methods to observe the current situation of the physical process under control and manipulate it, in order to make control decisions and apply them to influence the dynamic evolution of the process.

- Contextualization:
  The general concept of the process interface is particularized for the specific characteristics of the embedded control system platform. Hardware devices involved in the process interface are presented. The architecture and electronic technology aspects of these devices will be studied in other courses, their programming aspects will be discussed in this course.

Lectures: Example (Cont.)

- Motivation

  The process interface is presented as an essential module in the control program.

  The correct observation of the current situation is necessary to all simulation activities.

  Students already know how to communicate their computer programs with the real world; they have defined user interfaces in previous courses, where users were able to pass input data to the program and get the results using interfaces like local industrial interfaces and communication to the operator workplace. The process interface they are going to develop in this course has to enable communication between the controller and the object simulator.

- Bibliography
  - References to analog-to-digital and digital-to-analog converters.
  - References to discrete and timer I/O.
  - References to data acquisition industrial I/O interfaces.
  - References to APIs of data acquisition libraries.

Lectures: Example (Cont.)

- Concepts
  - The necessity of observation and manipulation of the physical process is presented.
  - The different types of signals: input and output, digital and analog, their codification and their interpretation as physical magnitudes are presented.
  - The available methods of signal acquisition are presented and their programming aspects and the response time requirements are discussed.
  - C-based Board-Support-Package I/O libraries for embedded control systems are presented and discussed.

.

Lectures: Example (Cont.)

- Examples
  - o The signals from a heterogeneous object are used as an example of analog and digital I/O signals
    - The alarm signals from the empty/overload sensors is used as an example of digital input signal.
    - The signal to the mixer motor, which is manipulated on an ON/OFF basis, is used as an example of digital output signal.
    - Signals from the flow scales are used as an example of analog input signal.

- Control Questions and Recommended Further Reading
  - o Question Example:
    - How many types of analog-to-digital converters do you know?
    - How is influenced the accuracy of the ADC by the environment temperature?

Lectures: Recommendations

- During the lecture all the conversations in the classroom should be public, addressed to all of the members.
- The sequence of the lecture should be planned and predictable.
- The dynamics of the lecture should be attractive to the audience with short and direct explanations.
- The lecture shouldn't explain all the details of the concepts of the subject, it should give enough hints to let students autonomously complete the knowledge acquisition.
- The students must have the chance to formulate questions.
- Behavior rules must be defined to let the lecture to be productive and the corrective mechanisms should be effective and proportional.
- Potential distractors of the audience attention should be avoided or minimized.

Labs

Labs

The lab is the first practical exercise that students take to acquire a basic set of skills related to the topic presented in the lecture. The exercises in the labs solve specific and well-defined problems; they are guided, fully documented, and in progressively increasing complexity. The labs provide students with a set of tools and skills that can be used to solve more open problems during the seminars.

Labs: Structure

During lab, **the teacher**:

- PREVIOUSLY: plans the lab session based on his/her professional experience.
- Presents the learning goals of the lab.
- Contextualizes the lab within the subject.
- Motivates the practical exercise to be performed during the lab based on the importance of the problem it addresses.
- Lists and comments the equipment, the material and documentation resources needed to perform the lab.
- Describes the correct utilization of the lab equipment and warns about potential material and personal damage due to inappropriate use.
- Answers students questions during the practical exercise.
- LATER: should analyze the lab session to improve their professional skills.

Labs: Structure (Cont.)

During lab, **the student**:

- PREVIOUSLY: has studied the lab documentation, and has attended the related lectures.
- Is receptive to the teacher's indications and is proactive.
- Takes notes to remember the indications.
- Asks for concepts' clarification if necessary.
- Works in teams of two students on the practical exercises of the lab.
- Answers the questions of the teacher related to the exercise.
- LATER: should review and document the results of the practical exercises and eventually performs the extra optional exercises.

Labs: Structure (Cont.)

During lab, **the technical assistant**:

- PREVIOUSLY: Sets the necessary equipment for the lab in each of the workbenches based on the teacher's requests and his/her professional experience.
- Helps solving problems that could arise related to the equipment, power supply, communications and software, making diagnosis about the safeness and correctness operation of the equipment and replacing damaged components.
- LATER: should analyze the lab session to improve their professional skills.

Labs: Goal

- After the lab the student should have acquired the skills to develop a basic data acquisition system. These skills will be useful in the next related seminars and mini-project exercises.

Labs: Prerequisites

- The student should have attended the related lecture and have read the recommended further lectures and the lab guide.
- Working teams of two people should have been set.

Labs: Lab Example

"Process Interface" Subject - "Analog and Digital I/O" Lab

In this lab analog input and output signals are processed using the internal converters of the industrial embedded controller.

- Goal

  To learn how to acquire analog input signals, how to generate analog output signal and how to read/generate digital I/O signals using the Board-Support-Package library of the embedded controller.

- Contextualization

  The general concept of the process interface is particularized for the specific characteristics of the personal computer control platform studied in this module of the course.

  Hardware devices involved in the process interface are presented. The architecture and electronic technology aspects of these devices will be studied in other courses, their programming aspects will be studied in this course though.

Labs: Lab Example (Cont.)

- Motivation

  The importance of the accurate reading and generating both analog and digital signals in the control of physical processes is presented.

- Contents

  The lab documentation starts with and introduction, the general problem specification, the necessary hardware equipment – ADC/DAC and Digital I/O, and a description of the Board-Support-Package supporting the interface functions.

Labs: Lab Example (Cont.)

- List of exercises:

1. Introductory phase - basic and completely guided exercise
   - The process to use the ADC/DAC and digital I/O to acquire/generate signals. The transfer functions of the analog signals are presented. Dual-controller environment is user to enable mirror connections.
2. Reinforcement phase
   - The student is requested to acquire a different analog signal, coming it from a different analog input port, and to interpret this new signal with a different transfer function. The student must develop the exercise without the previous guide. The physical signal are simulated using pre-programed signal generator in this case. The frequency of the signal acquisition is set constant.
3. Advanced phase
   - The student is requested to combine the code that was developed in the previous exercises with the developed code in a previous labs. The goal is to acquire and save as log file all input signal from the one controller and all generated signal from the other controller.
4. Optional phase
   - The student is requested to analyze both input and output signal streams logged.

Labs: Recommendations

- The lab should be fully documented.
- The lab guide should start with an introduction that remarks to the concepts of the lecture that are going to be applied in the practical exercise, the goals definition and a list of the material and documental resources that will be needed.
- The guide should continue with a definition of the practical activities in the following phases of progressive and increasing of complexity: introductory, reinforcement, advanced and optional.
- The first exercise in the introductory phase should be described step by step.
- The second exercise in the reinforcement should practice the same concepts and method than before but on a different set of problem data and without the help of the guide
- The third exercise in the advanced phase should practice the application of the previously acquired tools, protocols and skills to solve a small sized and small complexity application problem.
- A fourth exercise should be defined in an optional final phase to let advanced students to consider further technical questions related to the topic of the lab.
- The teacher should supervise the practical exercises of the students, answering their questions, guiding them and providing enough hints to let the students find solutions by them shelf.
- Recurrent errors and problems, and interesting student's designs during the lab should be shared with the whole group. Remote desktop sessions of the workbench computer screen could be presented on the slide projector.

Seminars

Seminars

During the seminars the students must solve problems on the topic of the lecture. They have already exercised on related tools and procedures in the previous laboratories.

The type of the activities in the seminar has to do with the application of those tools and procedures to design solutions to specific problems. These problems are manageable parts of more complex problems. The activities involve information searching over the Internet, designs and calculations.

Students work in groups of 4 people, but they must explain and share their experiences with the whole group during the seminar meetings.

In the seminars of the course different problems and sub-problems related to the design and programming of physical processes controllers are analyzed from the perspective of the personal computer control platform studied in this modules.

Seminars: Structure

- The seminar starts with a teacher speech remembering about the key concepts presented during the lecture, and the proposition of several related design problems.
- The problems are decomposed in smaller parts, and the connections between the different sub-problems are discussed.
- Some of the problems are proposed to the whole group to start a debate, meanwhile other problems are proposed to be selected and solved by the different working teams during the seminar.
- After the investigation and design activities developed by each of the working teams the spokespeople share the obtained conclusions with the whole group and a second debate starts.
- The seminar ends with a resume by the teacher and the homework definition for each of the working teams that will prepare the next seminar session.

Seminars: Goals

- To practice the concepts presented in the related lectures and verify that these concepts have been assimilated correctly.
- To relate the previous concepts with other technical concepts that usually are studied in different subjects, in a contextualized way.
- Acquire team-working skills.
- Acquire documentation and presentation skills.
- Acquire critical searching of information skills.

Seminars: Prerequisites

- The student should have attended the related lecture and have read the further recommended readings.

- Working teams of four people should have been set. A balanced team is recommended with a similar level of initial knowledge for each of its members. One of the members' team will act as spokesperson.

Seminars: Example

- Problem: Integration of the following C mini-project modules for the implementation of the concrete production centre : Process Interface Module and Control Module.

- Sub-problems:
  - Simulation of the signal dynamics to validate the simulator.
  - Simulation of the behavior of the physical process under control due to expected perturbations (aperture of the output valve, change of the ambient temperature and change of the input liquid temperature).
  - Observations of the continuous signals.
  - Digital signal handling.
  - Analog signals generation according to a predefined laws.
  - Programming of the analog input and output modules using BSP libraries.
  - Programming of the digital input and output modules using BSP libraries.
  - Log file generation and analysis.

Mini-project

Mini-project

During the mini-project students apply the knowledge and skills that they have acquired in the lectures, labs and seminars to develop in an integral way the controller of a physical process.

The problem of the mini-project is the highest complexity problem in the course.

The working teams in the mini-project are the same as in the seminars.

The designs developed by the teams during the seminars are used as components of the mini-project's problem's solution.

The teams can use in their own mini-projects, seminar designs that other teams have shared.

Mini-project: Goals

- After the mini-project, the students should be able to integrate the tools and protocols practiced during the course to develop simple complexity and medium sized applications to simulate physical process.

- They should be able to document and present the mini-project process and outcome.

Mini-project: Prerequisites

- The students should have attended the lectures and completed the practical exercises of the labs and seminars that are related to the module of the mini-project that is going to be developed.

Mini-project: Example

To develop of a complete simulator of the inputs and outputs of one subsystem of a concrete production centre developing the following modules:

- Analog signals generators according to given equations.
- Generators-to-DAC Multichannel Interface Module.
- Timer activation module.
- Both analog and digital acquisition modules.
- Log Module.

Scheduling

Scheduling

Chapter	Type	Topic	Week														
			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	**Introduction**																
1	Lecture		x														
1	Seminar		x														
1	Lab		x														
1	Mini-project		x														
2	**Architecture of Computer Control Systems**																
2	Lecture			x													
2	Seminar			x													
2	Lab			x													
2	Mini-project			x													
3	**Organization and structure of computers for control purposes**																
3	Lecture				x												
3	Seminar				x												
3	Lab				x												
3	Mini-project				x												
4	**Basic control algorithms**																
4	Lecture					x											
4	Seminar					x											
4	Lab					x											
4	Mini-project					x											
5	**Real-Time software environment**																
5	Lecture						x										
5	Seminar						x										
5	Lab						x										
5	Mini-project						x										
5	**Real-Time software environment**																
5	Lecture							x									
5	Seminar							x									
5	Lab							x									
5	Mini-project							x									

Scheduling

Chapter	Type	Topic	Week														
			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
6	**SCADA**																
6	Lecture								x								
6	Seminar								x								
6	Lab								x								
6	Mini-project								x								
6	**SCADA**																
6	Lecture									x							
6	Seminar									x							
6	Lab									x							
6	Mini-project									x							
7	**Simulators – general theory**																
7	Lecture										x						
7	Seminar										x						
7	Lab										x						
7	Mini-project										x						
7	**Simulators – general theory**																
7	Lecture											x					
7	Seminar											x					
7	Lab											x					
7	Mini-project											x					
8	**Simulators – practical aspects**																
8	Lecture												x				
8	Seminar												x				
8	Lab												x				
8	Mini-project												x				
8	**Simulators – practical aspects**																
8	Lecture													x			
8	Seminar													x			
8	Lab													x			
8	Mini-project													x			

Scheduling

Chapter	Type	Topic	Week														
			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
9	**Simulation of distributed objects and control systems**																
9	Lecture														x		
9	Seminar														x		
9	Lab														x		
9	Mini-project														x		
10	**Simulators validation**																
10	Lecture															x	
10	Seminar															x	
10	Lab															x	
10	Mini-project															x	
11	**Real-Time system improvement using simulation environment**																
11	Lecture																x
11	Seminar																x
11	Lab																x
11	Mini-project																x

Human
and
Material Resources

Human and Material Resources: Introduction

This methodology requires classroom-laboratories with general and pertinent equipment: personal computers, bundle of two embedded controllers.

Open source software is used for programming the applications, which will allow reducing the cost of the project.

This deliverable describes the resources required for learning activities and the staff required for supporting this methodology.

Human and Material Resources: The classroom-laboratory

The methodology proposed for this project can be enhanced using a single fully equipped classroom-laboratory.

These classroom-laboratories are interesting in order to avoid differentiating between theory, problems, and the laboratory practice and providing flexibility in the session development.

Each type of learning activity has different requirements as explained in the following subsections.

Human and Material Resources: The classroom-laboratory

Lectures

To present the main ideas of the contents we require the following typical classroom equipment:

- Bundle of two board properly connected.
- Professor personal computer.
- Digital projector.
- Office applications.
- Internet connection.

To enhance the learning experience, the following extra equipment is recommended

- e-learning materials

Human and Material Resources: The classroom-laboratory

Seminars and problems

Seminars require the same equipment than the lectures because the main purpose is to solve problems and discuss possible solutions:

- Student personal computer or mobile device.
- Collaborative software environments.
- Office applications.
- Internet connection.
- Development software.

To enhance the learning experience, it is recommended the following extra equipment

- Samples of real devices (sensors, actuators, etc.).

Human and Material Resources: The classroom-laboratory

Laboratory

The resources required for the “Industrial computers” module are:

- Student personal computer.
- Power source.
- Oscilloscope.
- Multimeter.
- Scale models, for example a “liquids tank” scale model.
- Bundle of two embedded controllers.
- Development environment software.
- Internet access.

Miniprojects

This activity is dedicated to the planning, design and development of an industrial control system.

Besides the equipment already mentioned in the previous section, for the miniproject, the following additional equipment will be required:

- Student personal computer or mobile device.
- Collaborative software environments.
- Office applications.
- Internet connection.
- Development software.

To enhance the hands on experience, it is recommended the following extra equipment

- Samples of real devices (sensors, actuators, etc.).

Human and Material Resources: Human resources

Teachers

Regarding teachers, the assistance consists of resolving questions related to the teaching structure and the synchronization of learning sessions (lectures, labs, seminars) as well as questions of the specific contents of the course.

Teachers will follow a course about the implementation of the PBL methodology, the organization of the different learning units, the evaluation system for the students.

Part of the training course will deal with how to teach a class applying the proposed methodology to a small group of students. This is a pilot course for testing the proposals.

Human and Material Resources: Human resources

Technicians

Laboratory technician is required to set-up all the hardware and software tools. Technicians will obtain support on aspects related to the installation and configuration of the software and hardware used to develop the laboratories and miniprojects.

Regarding the technicians, the training course shows how to use the necessary development tools (hardware and software) and their installation and configuration for the laboratory and mini-project activities.

Human and Material Resources: Human resources

Administrative

The administrative staff will have support to help them translate the EU evaluation marks to the PC evaluation system as well as regarding the transference credits system.

The training of the administrative staff explains the use of the ECTS credits and its transference to the PC credit system as well as the grades equivalence among EU and the different PC systems.

Evaluation

Evaluation: Introduction

- A very important aspect of the learning process as it will allow us to determine the level of assimilation of knowledge and skills by students.
- Not only be focused on the technical knowledge of the subject but should also include assessment of those skills and competencies that students must acquire.
- Pay attention to how they have developed cognitive skills (analysis, synthesis, application, evaluation, and critique)
- Action skills (organizing time, resources, coordination, negotiating, tolerating)
- The assessment must take into account how students are acquiring the knowledge, skills and competencies and ensure that those who pass the course have appropriate capabilities
- Problem solving related to real world problems is motivating for students as they see direct application and better assimilate concepts
- Students identify the problem, research on how to solve it applying concepts and principles. If they work in teams, develop communication skills and collaborative work, developing analytic skills
- During the evaluation process large amount of information will be collected

Evaluation: Formation of Teams

- Teams of 2 or 3 students.
- Groups should remain invariant throughout the course.
- Students who form a team have a similar level of knowledge.
- To set the level of students, the first day of class an objective type test will be performed, with around 25 questions, each one with 3 possible answers and only one answer will be correct. In this way we can know the real starting level of the students. This test should contain questions on microprocessor-based hardware and programming system.
- Questions should be aimed to collect general aspects of the concepts required as prerequisites to begin to pursue the matter. For example:
    - Basic Engineering concepts
    - Programming (Software)

Evaluation: Student evaluation methodology

Level One: Attitude (Student engagement)

- Motivation that students have within the course.
- Keep an ongoing dialogue with different student groups of the course, this should be maintained by the lecturer throughout the entire course.
- Special attention to students deliver activities in the time and manner agreed as it is a clear indicator of student motivation
- Deadline for delivery of the different activities that students set and meet deadlines. This should be something to evaluate.
- Tis level of evaluation should be a part of the final mark. This part can be 10% of the final overall rating.
- Identify those students with a special motivation for the subject because their attitude is above average
- It is important that students meet requirements of work deadlines in the subject. This will facilitate the professional future of the graduates in order to work in environments with strict deadlines.

Evaluation: Student evaluation methodology

Level Two: Learning

We determine the acquisition of knowledge and skills that students have acquired throughout the course.

Problems:

- Lecturer presents main ideas of lecture contents and proposes some application problems which student solves individually.
- The teachers throughout the course individually assess how the student has solved different problems.
- For example, through the problems we can assess how students can generate analog output using the analytical representation of the modification law. We can see if the student can program the equation and to understand how to link the generator to the DAC module and to run all of this under timer control, etc.
- For the evaluation of the problem must take into account: Approach resolution procedures, steps followed in the resolution, final result, method, clarity of presentation and approach, inclusion of units of measure, focus on the important issues facing superfluous. In the event that the final result is not correct,

Evaluation: Student evaluation methodology

Level Two: Learning

Laboratory:

- A practical problem previously presented during lecture.
- Students work by teams of two/three students.
- During the lab sessions students will show the teacher how they are solving the proposed activities
- Teacher will make questions about how is the resolution of the activity.
- At the end of each lab session the teacher will rate each group based on the work done and the objectives achieved.
- For evaluation can take into account:
    - Introduction phase: will reflect 20% of the grade.
    - Reinforcement Phase: will reflect 40% of the grade.
    - Advanced stage: Will represent a 40% of the grade.

Evaluation: Student evaluation methodology

Level Two: Learning

Seminars:

- A panel discussion with student teams (around six students) is proposed, consisting generally of solving a problem by means of PBL.
- The teachers will meet with each of the groups who will present how they have raised the issue, what options for the resolution are viable and which ones have been taken.
- It is important that the teacher dialogue with all members of the group to identify how well attended and have acquired the relevant knowledge.
- Some aspects to be taken into account for the assessment:
    - Level of responsibility among group members.
    - Number of studied and analysis in terms of advantages and disadvantages of each possible solution solutions.
    - Quality technical report writing as to the work done
    - Defense of ideas and how to defend against constructive criticism
    - Interaction of knowledge of this area with other
    - Management of bibliographic sources
    - Extra work done with respect to the requested
    - Robustness of the proposed solution adopted
    - Etc.

Evaluation: Student evaluation methodology

Level Two: Learning

Mini-project:

- Dedicated to planning, design and development of the control system of a real problem design.
- The developed project will presented publicly to a jury composed of three lecturers.
- The jury assesses the following aspects including some transversal skills:
    - Report: Maximum score of this part is 25% of the total mark.
    - Oral presentation: The score of this part is 10% maximum. The team presents the work during maximum of 20 minutes.
    - Implementation: The score assigned to this part is 65% of total mark. After the oral presentation, the team shows the project application.

Evaluation: Student evaluation methodology

Level Two: Learning

Mini-project (Cont.):

In each of these sections in addition to the assessment of knowledge, the teacher should take into consideration and evaluate all the important skills and transversal skills for engineers:

- Cognitive skills: Analysis, synthesis, application, evaluation, critique, etc.
- Action skills: Organizing time, resources, coordination, negotiating, tolerating, etc.

The advantage of the methodology is that it allows including other skills that assessment allow an integrated formation of the student: *competition, working in teams, cooperation, oral presentations, budget management, report redaction, etc*.

The rating of these skills should be included in the appropriate rating to each of the evaluation issues associated at this level.

Evaluation: Student evaluation methodology

Level Three: Grading (outcome)

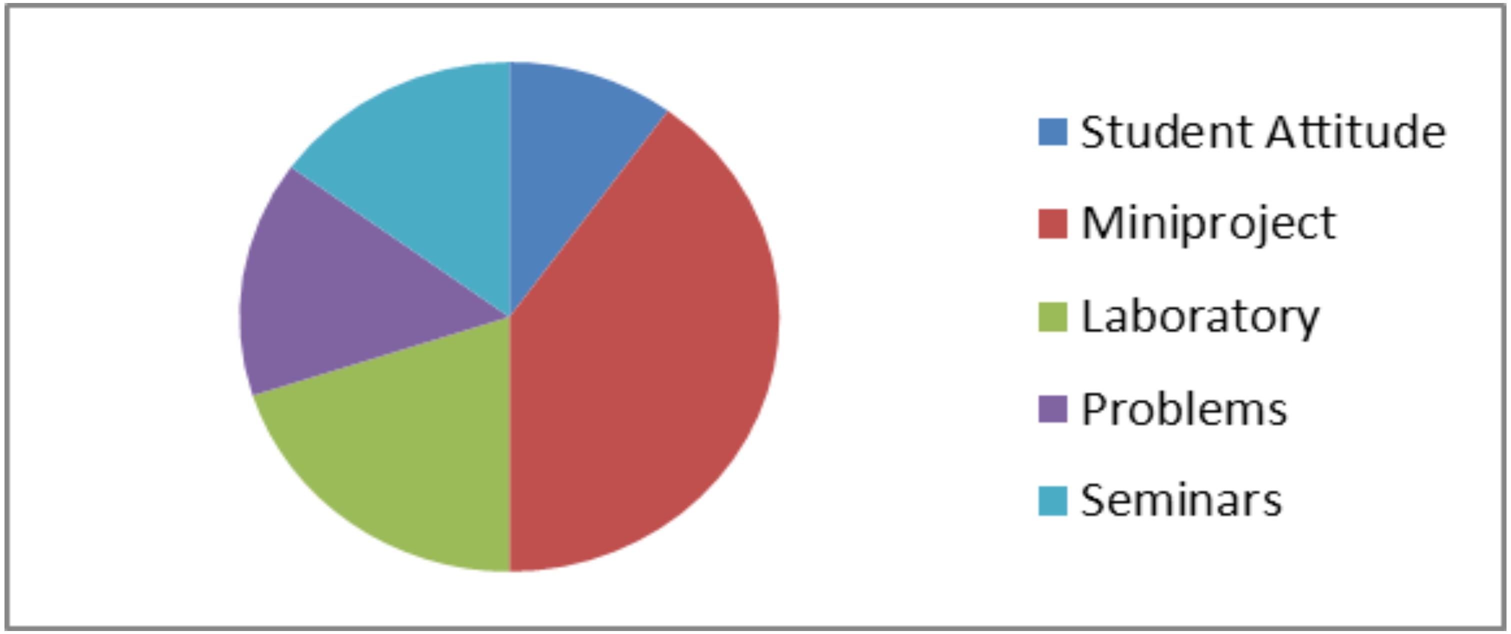
- We will collect all grades earned along the continuous assessment developed along the course
- Proceed to obtain the final grade for the course.
- We establish a just proportion between the different sections to ensure a fair rating for differentiating their individual acquisition of knowledge and skills against the student group work.
- The proposal will apply as follows:
    - The evaluation of the student attitude (A) a 10% of the final score.
    - The evaluation of the miniproject represents (MP) a 40% of the final score.
    - The evaluation of the Laboratory (L): 20% of the final score.
    - The evaluation of the Problems (P): 15% of the final score.
    - The evaluation of the Seminars (S): 15% of the final score.

Evaluation: Student evaluation methodology

Level Three: Grading (outcome) (Cont.)

For the calculation of the final grade (FG) can be followed as the following equation:

$$FG = A*0.1+MP*0.4+L*0.2+P*0.15+S*0.15$$



Evaluation: Assessment of the methodology system

- Evaluation of the system used for teaching this subject.
- Important to know the opinion of students and teachers involved to find out what has been done well and what parts could be improved
- In this sense the evaluation board system from two points of view:

  - Student point of view:

    - A survey among students to have information about the acceptance of the course.
    - Students can give their opinion at the end of the course and before obtaining their qualifications
    - Design a survey in a way that can be simple and easy to answer. For instance, can be made based on 6 questions with 5 possible answers ("A": Strongly Agree; "B": Agree; "C": Unsure; "D": Disagree; "E": Strongly Disagree) for each, rated from A to E marks.

Evaluation: Assessment of the methodology system (Cont.)

- The questions can be of the type:
  - Has the subject methodology facilitated your learning process?
  - Has every important concept of the subject been addressed in the miniproject?
  - Has the complexity level of every part of the subject been reasonable?
  - Has the activities promoted cooperation skills as in real work environments?
  - Have you felt motivated during the learning process?
  - Would you recommend taking this course to other students?

- Lecturer viewpoint:
  - The opinion of teachers is important to make an overall assessment of how the course has worked and what aspects should be improved.
  - Teachers should maintain an open dialogue throughout the course and at the end make the balance.
  - The aspects to be evaluated are for example:
    - Ratio of approved students
    - Quality of ratings
    - Amount of work done by teachers
    - Problems that have arisen and how they have been resolved
    - Possible updating of content,
    - Duplication and overlap with other subjects, etc.

Thanks!

Technical University of Sofia

We succeed!

www.tu-sofia.bg

70 YEARS

Technical University of Sofia

We succeed!

medis

Introduction

Introduction:

- To detail the structure of the Advanced Industrial Informatics Specialization Modules (AIISM)
- Learn how to organize the AIISM methodology.
- The AIISM course uses a PBL (Problem Based Learning) methodology to instruct the design and implementation of II systems to control industrial processes
- The AIISM methodology is designed based on previous experiences of the EU universities on PBL and active learning techniques
- The purpose is to create a working environment for the students similar to the real environment in companies:
  - To guarantee fundamental knowledge of AIISM as basis for the development of further objectives
  - To accustom students work in teams when solving industrial problems
  - To encourage students to use practical skills to improve their problem solving abilities
  - To develop the capacity to adapt to any new computer based systems, due to rapid advances in this area.

- Other engineering transversal skills are gained during the course:
  - Teamwork
  - Technical competencies
  - Oral presentation
  - Budget management
  - Report redaction
  - Etc.

- The AIISM is structured with different activities of progressive complexity to facilitate the teams to develop their projects along the course.

- The activities are developed during 5 hours/day, one day of the week through a PBL methodology, using as case study the examples of several objects, their HIL simulators and corresponding control systems.

Learning activities

Learning activities description:

- Students have to apply the knowledge acquired from the lectures and the laboratory practices.
- The proposed learning activities are the following:
  - **Lecture and problems**: lecturer presents main ideas of lecture contents and proposes some application problems which student solves individually (e. g., 1 h).
  - **Laboratory session**: To implement (1 h 15") a practical problem previously presented during lecture. Students work by teams of two students.
  - **Seminars**: a panel discussion with student teams (4 students) lasting 45 minutes is proposed, consisting generally of solving a problem by means of PBL.
  - **Mini-project**: dedicated to planning, design and development of the control system of the educational liquids tank. The mini-project is performed by teams of 4 students during 2 hours. Weekly, the mini-project is advanced progressively.

Lectures

Lectures: Introduction

- Lecture is the first step in the learning process for each of the topics in each course's module.
- The lecturer presents the main topics of the theme contents.
- It includes some application examples.

Lectures: Structure

During lecture, **the teacher**:

- PREVIOUSLY: plans the lecture session based on their professional experience.
- Presents the learning objectives of the subject.
- Contextualizes the subject within the module, the course and the career, based on the problems to be solved and the resources that will be proposed to solve these problems.
- Motivates the subject based on the importance of the problems coped by the subject.
- Lists and discusses the bibliographic resources that support the concepts that will be presented during the lecture and that students may use to deepen in their study.
- Presents the key concepts related to the subject, providing the needed details to properly understand them, and specifying the extra available resources and the learning process that students should follow to complete the knowledge's acquisition. Follows a logical order for the argumentation, so that students can acquire the knowledge progressively and uses illustrative examples to clarify the presented concepts.

Lectures: Structure (Cont.)

- Verifies that students correctly understand the presented concepts and adapts the speech if needed, allowing students to ask questions about the concepts that are not clear enough and observing the students' answers to the control questions.
- Informs about the skills related to the lecture that will be evaluated.
- Tells students about the practical application of the presented concepts that will be performed later during the seminars, laboratories and mini-project, and which are related to the lecture, encouraging students to study the issue with sufficient interest.
- After the lecture, keeps open communication channels with students, so they can make consultations before next lecture if necessary.
- LATER: should analyze the lecture session to improve their professional skills.

Lectures: Structure (Cont.)

During lecture, **the student**:

- PREVIOUSLY: has studied the recommended previous readings.
- Is receptive to the teacher's presentation and is proactive.
- Takes notes to conceptualize what is being exposed to facilitate its further study.
- Asks for concepts' clarification if necessary.
- Answers the questions the teacher addresses to the audience.
- LATER: should follow the learning method proposed by the teacher

Lectures: Goals

- After the lecture the students will know what is the problem and its importance, how the problem is described, which available alternatives would solve it and the criteria to select an appropriate solving alternative even though they will not have yet the skills to apply the proposed methods.

- They will have a studying plan that will be organized in the form of a set of labs, seminars and mini-project.

Lectures: Prerequisites

- Previously to the lecture the student must be informed about the prerequisites to successfully follow the lecture.

- If the prerequisites aren’t satisfied a previous to the lecture formation plan should be supplied.

Lectures: Example

"Process Interface"

In this lecture the signal exchange between the controller computer and the simulator one will be discussed and explained.

- Goal:
  To learn methods to observe the current situation of the physical process under control and manipulate it, in order to make control decisions and apply them to influence the dynamic evolution of the process.

- Contextualization:
  The general concept of the process interface is particularized for the specific characteristics of the embedded control system platform. Hardware devices involved in the process interface are presented. The architecture and electronic technology aspects of these devices will be studied in other courses, their programming aspects will be discussed in this course.

Lectures: Example (Cont.)

- Motivation

  The process interface is presented as an essential module in the control program.

  The correct observation of the current situation is necessary to all simulation activities.

  Students already know how to communicate their computer programs with the real world; they have defined user interfaces in previous courses, where users were able to pass input data to the program and get the results using interfaces like local industrial interfaces and communication to the operator workplace. The process interface they are going to develop in this course has to enable communication between the controller and the object simulator.

- Bibliography
  - References to analog-to-digital and digital-to-analog converters.
  - References to discrete and timer I/O.
  - References to data acquisition industrial I/O interfaces.
  - References to APIs of data acquisition libraries.

Lectures: Example (Cont.)

- Concepts
  - The necessity of observation and manipulation of the physical process is presented.
  - The different types of signals: input and output, digital and analog, their codification and their interpretation as physical magnitudes are presented.
  - The available methods of signal acquisition are presented and their programming aspects and the response time requirements are discussed.
  - C-based Board-Support-Package I/O libraries for embedded control systems are presented and discussed.

.

Lectures: Example (Cont.)

- Examples
  - The signals from a heterogeneous object are used as an example of analog and digital I/O signals
    - The alarm signals from the empty/overload sensors is used as an example of digital input signal.
    - The signal to the mixer motor, which is manipulated on an ON/OFF basis, is used as an example of digital output signal.
    - Signals from the flow scales are used as an example of analog input signal.

- Control Questions and Recommended Further Reading
  - Question Example:
    - How many types of analog-to-digital converters do you know?
    - How is influenced the accuracy of the ADC by the environment temperature?

Lectures: Recommendations

- During the lecture all the conversations in the classroom should be public, addressed to all of the members.
- The sequence of the lecture should be planned and predictable.
- The dynamics of the lecture should be attractive to the audience with short and direct explanations.
- The lecture shouldn't explain all the details of the concepts of the subject, it should give enough hints to let students autonomously complete the knowledge acquisition.
- The students must have the chance to formulate questions.
- Behavior rules must be defined to let the lecture to be productive and the corrective mechanisms should be effective and proportional.
- Potential distractors of the audience attention should be avoided or minimized.

Labs

Labs

The lab is the first practical exercise that students take to acquire a basic set of skills related to the topic presented in the lecture. The exercises in the labs solve specific and well-defined problems; they are guided, fully documented, and in progressively increasing complexity. The labs provide students with a set of tools and skills that can be used to solve more open problems during the seminars.

Labs: Structure

During lab, **the teacher**:

- PREVIOUSLY: plans the lab session based on his/her professional experience.
- Presents the learning goals of the lab.
- Contextualizes the lab within the subject.
- Motivates the practical exercise to be performed during the lab based on the importance of the problem it addresses.
- Lists and comments the equipment, the material and documentation resources needed to perform the lab.
- Describes the correct utilization of the lab equipment and warns about potential material and personal damage due to inappropriate use.
- Answers students questions during the practical exercise.
- LATER: should analyze the lab session to improve their professional skills.

Labs: Structure (Cont.)

During lab, **the student**:

- PREVIOUSLY: has studied the lab documentation, and has attended the related lectures.
- Is receptive to the teacher's indications and is proactive.
- Takes notes to remember the indications.
- Asks for concepts' clarification if necessary.
- Works in teams of two students on the practical exercises of the lab.
- Answers the questions of the teacher related to the exercise.
- LATER: should review and document the results of the practical exercises and eventually performs the extra optional exercises.

Labs: Structure (Cont.)

During lab, **the technical assistant**:

- PREVIOUSLY: Sets the necessary equipment for the lab in each of the workbenches based on the teacher's requests and his/her professional experience.
- Helps solving problems that could arise related to the equipment, power supply, communications and software, making diagnosis about the safeness and correctness operation of the equipment and replacing damaged components.
- LATER: should analyze the lab session to improve their professional skills.

Labs: Goal

- After the lab the student should have acquired the skills to develop a basic data acquisition system. These skills will be useful in the next related seminars and mini-project exercises.

Labs: Prerequisites

- The student should have attended the related lecture and have read the recommended further lectures and the lab guide.
- Working teams of two people should have been set.

Labs: Lab Example

“Process Interface” Subject - “Analog and Digital I/O” Lab

In this lab analog input and output signals are processed using the internal converters of the industrial embedded controller.

- Goal

  To learn how to acquire analog input signals, how to generate analog output signal and how to read/generate digital I/O signals using the Board-Support-Package library of the embedded controller.

- Contextualization

  The general concept of the process interface is particularized for the specific characteristics of the personal computer control platform studied in this module of the course.

  Hardware devices involved in the process interface are presented. The architecture and electronic technology aspects of these devices will be studied in other courses, their programming aspects will be studied in this course though.

Labs: Lab Example (Cont.)

- Motivation

  The importance of the accurate reading and generating both analog and digital signals in the control of physical processes is presented.

- Contents

  The lab documentation starts with and introduction, the general problem specification, the necessary hardware equipment – ADC/DAC and Digital I/O, and a description of the Board-Support-Package supporting the interface functions.

Labs: Lab Example (Cont.)

- List of exercises:

1. Introductory phase - basic and completely guided exercise
   - The process to use the ADC/DAC and digital I/O to acquire/generate signals. The transfer functions of the analog signals are presented. Dual-controller environment is user to enable mirror connections.
2. Reinforcement phase
   - The student is requested to acquire a different analog signal, coming it from a different analog input port, and to interpret this new signal with a different transfer function. The student must develop the exercise without the previous guide. The physical signal are simulated using pre-programed signal generator in this case. The frequency of the signal acquisition is set constant.
3. Advanced phase
   - The student is requested to combine the code that was developed in the previous exercises with the developed code in a previous labs. The goal is to acquire and save as log file all input signal from the one controller and all generated signal from the other controller.
4. Optional phase
   - The student is requested to analyze both input and output signal streams logged.

Labs: Recommendations

- The lab should be fully documented.
- The lab guide should start with an introduction that remarks to the concepts of the lecture that are going to be applied in the practical exercise, the goals definition and a list of the material and documental resources that will be needed.
- The guide should continue with a definition of the practical activities in the following phases of progressive and increasing of complexity: introductory, reinforcement, advanced and optional.
- The first exercise in the introductory phase should be described step by step.
- The second exercise in the reinforcement should practice the same concepts and method than before but on a different set of problem data and without the help of the guide
- The third exercise in the advanced phase should practice the application of the previously acquired tools, protocols and skills to solve a small sized and small complexity application problem.
- A fourth exercise should be defined in an optional final phase to let advanced students to consider further technical questions related to the topic of the lab.
- The teacher should supervise the practical exercises of the students, answering their questions, guiding them and providing enough hints to let the students find solutions by them shelf.
- Recurrent errors and problems, and interesting student's designs during the lab should be shared with the whole group. Remote desktop sessions of the workbench computer screen could be presented on the slide projector.

Seminars

Seminars

During the seminars the students must solve problems on the topic of the lecture. They have already exercised on related tools and procedures in the previous laboratories.

The type of the activities in the seminar has to do with the application of those tools and procedures to design solutions to specific problems. These problems are manageable parts of more complex problems. The activities involve information searching over the Internet, designs and calculations.

Students work in groups of 4 people, but they must explain and share their experiences with the whole group during the seminar meetings.

In the seminars of the course different problems and sub-problems related to the design and programming of physical processes controllers are analyzed from the perspective of the personal computer control platform studied in this modules.

Seminars: Structure

- The seminar starts with a teacher speech remembering about the key concepts presented during the lecture, and the proposition of several related design problems.
- The problems are decomposed in smaller parts, and the connections between the different sub-problems are discussed.
- Some of the problems are proposed to the whole group to start a debate, meanwhile other problems are proposed to be selected and solved by the different working teams during the seminar.
- After the investigation and design activities developed by each of the working teams the spokespeople share the obtained conclusions with the whole group and a second debate starts.
- The seminar ends with a resume by the teacher and the homework definition for each of the working teams that will prepare the next seminar session.

Seminars: Goals

- To practice the concepts presented in the related lectures and verify that these concepts have been assimilated correctly.
- To relate the previous concepts with other technical concepts that usually are studied in different subjects, in a contextualized way.
- Acquire team-working skills.
- Acquire documentation and presentation skills.
- Acquire critical searching of information skills.

Seminars: Prerequisites

- The student should have attended the related lecture and have read the further recommended readings.

- Working teams of four people should have been set. A balanced team is recommended with a similar level of initial knowledge for each of its members. One of the members' team will act as spokesperson.

Seminars: Example

- Problem: Integration of the following C mini-project modules for the implementation of the concrete production centre : Process Interface Module and Control Module.

- Sub-problems:
  - Simulation of the signal dynamics to validate the simulator.
  - Simulation of the behavior of the physical process under control due to expected perturbations (aperture of the output valve, change of the ambient temperature and change of the input liquid temperature).
  - Observations of the continuous signals.
  - Digital signal handling.
  - Analog signals generation according to a predefined laws.
  - Programming of the analog input and output modules using BSP libraries.
  - Programming of the digital input and output modules using BSP libraries.
  - Log file generation and analysis.

Mini-project

Mini-project

During the mini-project students apply the knowledge and skills that they have acquired in the lectures, labs and seminars to develop in an integral way the controller of a physical process.

The problem of the mini-project is the highest complexity problem in the course.

The working teams in the mini-project are the same as in the seminars.

The designs developed by the teams during the seminars are used as components of the mini-project's problem's solution.

The teams can use in their own mini-projects, seminar designs that other teams have shared.

Mini-project: Goals

- After the mini-project, the students should be able to integrate the tools and protocols practiced during the course to develop simple complexity and medium sized applications to simulate physical process.
- They should be able to document and present the mini-project process and outcome.

Mini-project: Prerequisites

- The students should have attended the lectures and completed the practical exercises of the labs and seminars that are related to the module of the mini-project that is going to be developed.

Mini-project: Example

To develop of a complete simulator of the inputs and outputs of one subsystem of a concrete production centre developing the following modules:

- Analog signals generators according to given equations.
- Generators-to-DAC Multichannel Interface Module.
- Timer activation module.
- Both analog and digital acquisition modules.
- Log Module.

Scheduling

Scheduling

Chapter	Type	Topic	Week														
			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	**Introduction**																
1	Lecture		x														
1	Seminar		x														
1	Lab		x														
1	Mini-project		x														
2	**Architecture of Computer Control Systems**																
2	Lecture			x													
2	Seminar			x													
2	Lab			x													
2	Mini-project			x													
3	**Organization and structure of computers for control purposes**																
3	Lecture				x												
3	Seminar				x												
3	Lab				x												
3	Mini-project				x												
4	**Basic control algorithms**																
4	Lecture					x											
4	Seminar					x											
4	Lab					x											
4	Mini-project					x											
5	**Real-Time software environment**																
5	Lecture						x										
5	Seminar						x										
5	Lab						x										
5	Mini-project						x										
5	**Real-Time software environment**																
5	Lecture							x									
5	Seminar							x									
5	Lab							x									
5	Mini-project							x									

Scheduling

Chapter	Type	Topic	Week 1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
6	**SCADA**																
6	Lecture								x								
6	Seminar								x								
6	Lab								x								
6	Mini-project								x								
6	**SCADA**																
6	Lecture									x							
6	Seminar									x							
6	Lab									x							
6	Mini-project									x							
7	**Simulators – general theory**																
7	Lecture										x						
7	Seminar										x						
7	Lab										x						
7	Mini-project										x						
7	**Simulators – general theory**																
7	Lecture											x					
7	Seminar											x					
7	Lab											x					
7	Mini-project											x					
8	**Simulators – practical aspects**																
8	Lecture												x				
8	Seminar												x				
8	Lab												x				
8	Mini-project												x				
8	**Simulators – practical aspects**																
8	Lecture													x			
8	Seminar													x			
8	Lab													x			
8	Mini-project													x			

Scheduling

Chapter	Type	Topic	Week														
			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
9	**Simulation of distributed objects and control systems**																
9	Lecture														x		
9	Seminar														x		
9	Lab														x		
9	Mini-project														x		
10	**Simulators validation**																
10	Lecture															x	
10	Seminar															x	
10	Lab															x	
10	Mini-project															x	
11	**Real-Time system improvement using simulation environment**																
11	Lecture																x
11	Seminar																x
11	Lab																x
11	Mini-project																x

Human
and
Material Resources

Human and Material Resources: Introduction

This methodology requires classroom-laboratories with general and pertinent equipment: personal computers, bundle of two embedded controllers.

Open source software is used for programming the applications, which will allow reducing the cost of the project.

This deliverable describes the resources required for learning activities and the staff required for supporting this methodology.

Human and Material Resources: The classroom-laboratory

The methodology proposed for this project can be enhanced using a single fully equipped classroom-laboratory.

These classroom-laboratories are interesting in order to avoid differentiating between theory, problems, and the laboratory practice and providing flexibility in the session development.

Each type of learning activity has different requirements as explained in the following subsections.

Human and Material Resources: The classroom-laboratory

Lectures

To present the main ideas of the contents we require the following typical classroom equipment:

- Bundle of two board properly connected.
- Professor personal computer.
- Digital projector.
- Office applications.
- Internet connection.

To enhance the learning experience, the following extra equipment is recommended

- e-learning materials

Human and Material Resources: The classroom-laboratory

Seminars and problems

Seminars require the same equipment than the lectures because the main purpose is to solve problems and discuss possible solutions:

- Student personal computer or mobile device.
- Collaborative software environments.
- Office applications.
- Internet connection.
- Development software.

To enhance the learning experience, it is recommended the following extra equipment

- Samples of real devices (sensors, actuators, etc.).

Human and Material Resources: The classroom-laboratory

Laboratory

The resources required for the “Industrial computers” module are:

- Student personal computer.
- Power source.
- Oscilloscope.
- Multimeter.
- Scale models, for example a “liquids tank” scale model.
- Bundle of two embedded controllers.
- Development environment software.
- Internet access.

Human and Material Resources: The classroom-laboratory

Miniprojects

This activity is dedicated to the planning, design and development of an industrial control system.

Besides the equipment already mentioned in the previous section, for the miniproject, the following additional equipment will be required:

- Student personal computer or mobile device.
- Collaborative software environments.
- Office applications.
- Internet connection.
- Development software.

To enhance the hands on experience, it is recommended the following extra equipment

- Samples of real devices (sensors, actuators, etc.).

Human and Material Resources: Human resources

Teachers

Regarding teachers, the assistance consists of resolving questions related to the teaching structure and the synchronization of learning sessions (lectures, labs, seminars) as well as questions of the specific contents of the course.

Teachers will follow a course about the implementation of the PBL methodology, the organization of the different learning units, the evaluation system for the students.

Part of the training course will deal with how to teach a class applying the proposed methodology to a small group of students. This is a pilot course for testing the proposals.

Human and Material Resources: Human resources

Technicians

Laboratory technician is required to set-up all the hardware and software tools. Technicians will obtain support on aspects related to the installation and configuration of the software and hardware used to develop the laboratories and miniprojects.

Regarding the technicians, the training course shows how to use the necessary development tools (hardware and software) and their installation and configuration for the laboratory and mini-project activities.

Human and Material Resources: Human resources

Administrative

The administrative staff will have support to help them translate the EU evaluation marks to the PC evaluation system as well as regarding the transference credits system.

The training of the administrative staff explains the use of the ECTS credits and its transference to the PC credit system as well as the grades equivalence among EU and the different PC systems.

Evaluation

Evaluation: Introduction

- A very important aspect of the learning process as it will allow us to determine the level of assimilation of knowledge and skills by students.
- Not only be focused on the technical knowledge of the subject but should also include assessment of those skills and competencies that students must acquire.
- Pay attention to how they have developed cognitive skills (analysis, synthesis, application, evaluation, and critique)
- Action skills (organizing time, resources, coordination, negotiating, tolerating)
- The assessment must take into account how students are acquiring the knowledge, skills and competencies and ensure that those who pass the course have appropriate capabilities
- Problem solving related to real world problems is motivating for students as they see direct application and better assimilate concepts
- Students identify the problem, research on how to solve it applying concepts and principles. If they work in teams, develop communication skills and collaborative work, developing analytic skills
- During the evaluation process large amount of information will be collected

Evaluation: Formation of Teams

- Teams of 2 or 3 students.
- Groups should remain invariant throughout the course.
- Students who form a team have a similar level of knowledge.
- To set the level of students, the first day of class an objective type test will be performed, with around 25 questions, each one with 3 possible answers and only one answer will be correct. In this way we can know the real starting level of the students. This test should contain questions on microprocessor-based hardware and programming system.
- Questions should be aimed to collect general aspects of the concepts required as prerequisites to begin to pursue the matter. For example:
    - Basic Engineering concepts
    - Programming (Software)

Evaluation: Student evaluation methodology

Level One: Attitude (Student engagement)

- Motivation that students have within the course.
- Keep an ongoing dialogue with different student groups of the course, this should be maintained by the lecturer throughout the entire course.
- Special attention to students deliver activities in the time and manner agreed as it is a clear indicator of student motivation
- Deadline for delivery of the different activities that students set and meet deadlines. This should be something to evaluate.
- Tis level of evaluation should be a part of the final mark. This part can be 10% of the final overall rating.
- Identify those students with a special motivation for the subject because their attitude is above average
- It is important that students meet requirements of work deadlines in the subject. This will facilitate the professional future of the graduates in order to work in environments with strict deadlines.

Evaluation: Student evaluation methodology

Level Two: Learning

We determine the acquisition of knowledge and skills that students have acquired throughout the course.

Problems:

- Lecturer presents main ideas of lecture contents and proposes some application problems which student solves individually.
- The teachers throughout the course individually assess how the student has solved different problems.
- For example, through the problems we can assess how students can generate analog output using the analytical representation of the modification law. We can see if the student can program the equation and to understand how to link the generator to the DAC module and to run all of this under timer control, etc.
- For the evaluation of the problem must take into account: Approach resolution procedures, steps followed in the resolution, final result, method, clarity of presentation and approach, inclusion of units of measure, focus on the important issues facing superfluous. In the event that the final result is not correct,

Evaluation: Student evaluation methodology

Level Two: Learning

Laboratory:

- A practical problem previously presented during lecture.
- Students work by teams of two/three students.
- During the lab sessions students will show the teacher how they are solving the proposed activities
- Teacher will make questions about how is the resolution of the activity.
- At the end of each lab session the teacher will rate each group based on the work done and the objectives achieved.
- For evaluation can take into account:
    - Introduction phase: will reflect 20% of the grade.
    - Reinforcement Phase: will reflect 40% of the grade.
    - Advanced stage: Will represent a 40% of the grade.

Evaluation: Student evaluation methodology

Level Two: Learning

Seminars:

- A panel discussion with student teams (around six students) is proposed, consisting generally of solving a problem by means of PBL.
- The teachers will meet with each of the groups who will present how they have raised the issue, what options for the resolution are viable and which ones have been taken.
- It is important that the teacher dialogue with all members of the group to identify how well attended and have acquired the relevant knowledge.
- Some aspects to be taken into account for the assessment:
  - Level of responsibility among group members.
  - Number of studied and analysis in terms of advantages and disadvantages of each possible solution solutions.
  - Quality technical report writing as to the work done
  - Defense of ideas and how to defend against constructive criticism
  - Interaction of knowledge of this area with other
  - Management of bibliographic sources
  - Extra work done with respect to the requested
  - Robustness of the proposed solution adopted
  - Etc.

Evaluation: Student evaluation methodology

Level Two: Learning

Mini-project:

- Dedicated to planning, design and development of the control system of a real problem design.
- The developed project will presented publicly to a jury composed of three lecturers.
- The jury assesses the following aspects including some transversal skills:
    - Report: Maximum score of this part is 25% of the total mark.
    - Oral presentation: The score of this part is 10% maximum. The team presents the work during maximum of 20 minutes.
    - Implementation: The score assigned to this part is 65% of total mark. After the oral presentation, the team shows the project application.

Evaluation: Student evaluation methodology

Level Two: Learning

Mini-project (Cont.):

In each of these sections in addition to the assessment of knowledge, the teacher should take into consideration and evaluate all the important skills and transversal skills for engineers:

- Cognitive skills: Analysis, synthesis, application, evaluation, critique, etc.
- Action skills: Organizing time, resources, coordination, negotiating, tolerating, etc.

The advantage of the methodology is that it allows including other skills that assessment allow an integrated formation of the student: *competition, working in teams, cooperation, oral presentations, budget management, report redaction, etc.*

The rating of these skills should be included in the appropriate rating to each of the evaluation issues associated at this level.

Evaluation: Student evaluation methodology

Level Three: Grading (outcome)

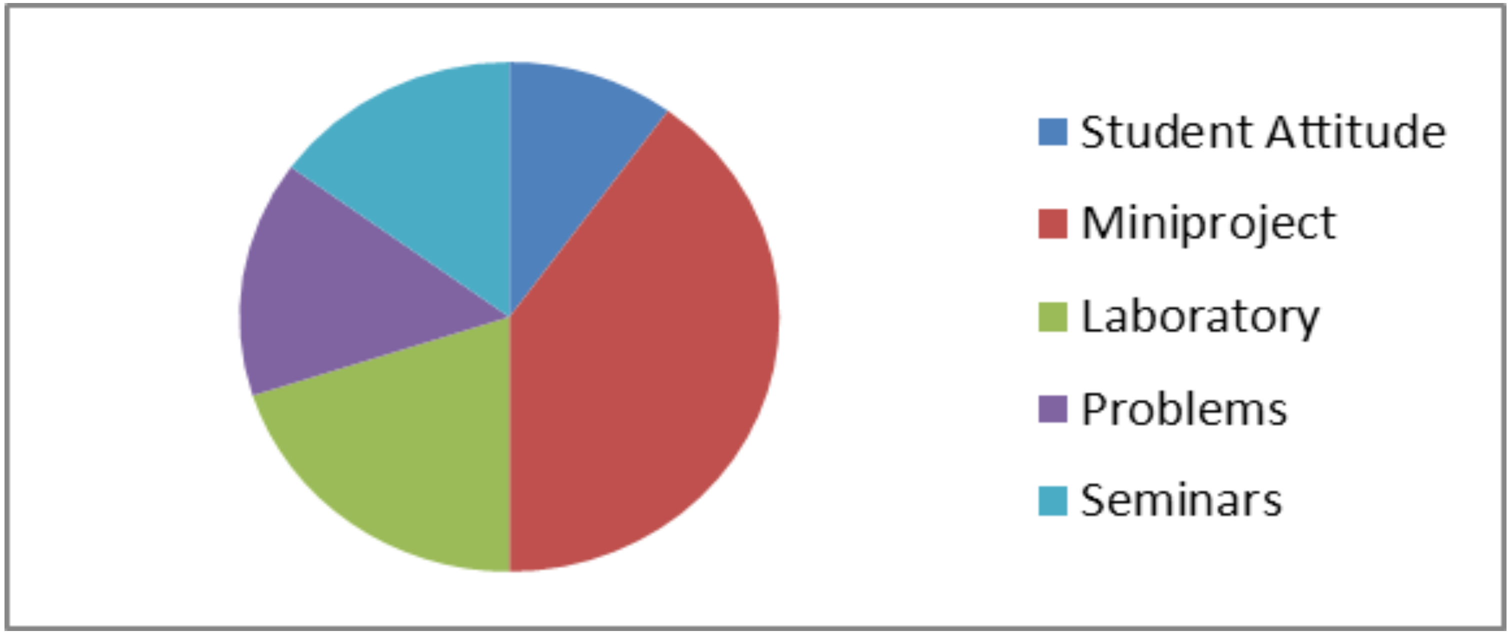
- We will collect all grades earned along the continuous assessment developed along the course
- Proceed to obtain the final grade for the course.
- We establish a just proportion between the different sections to ensure a fair rating for differentiating their individual acquisition of knowledge and skills against the student group work.
- The proposal will apply as follows:
    - The evaluation of the student attitude (A) a 10% of the final score.
    - The evaluation of the miniproject represents (MP) a 40% of the final score.
    - The evaluation of the Laboratory (L): 20% of the final score.
    - The evaluation of the Problems (P): 15% of the final score.
    - The evaluation of the Seminars (S): 15% of the final score.

Evaluation: Student evaluation methodology

Level Three: Grading (outcome) (Cont.)

For the calculation of the final grade (FG) can be followed as the following equation:

$$FG = A*0.1 + MP*0.4 + L*0.2 + P*0.15 + S*0.15$$



Evaluation: Assessment of the methodology system

- Evaluation of the system used for teaching this subject.
- Important to know the opinion of students and teachers involved to find out what has been done well and what parts could be improved
- In this sense the evaluation board system from two points of view:
    - Student point of view:
        - A survey among students to have information about the acceptance of the course.
        - Students can give their opinion at the end of the course and before obtaining their qualifications
        - Design a survey in a way that can be simple and easy to answer. For instance, can be made based on 6 questions with 5 possible answers ("A": Strongly Agree; "B": Agree; "C": Unsure; "D": Disagree; "E": Strongly Disagree) for each, rated from A to E marks.

Evaluation: Assessment of the methodology system (Cont.)

- The questions can be of the type:
  - Has the subject methodology facilitated your learning process?
  - Has every important concept of the subject been addressed in the miniproject?
  - Has the complexity level of every part of the subject been reasonable?
  - Has the activities promoted cooperation skills as in real work environments?
  - Have you felt motivated during the learning process?
  - Would you recommend taking this course to other students?

- Lecturer viewpoint:
  - The opinion of teachers is important to make an overall assessment of how the course has worked and what aspects should be improved.
  - Teachers should maintain an open dialogue throughout the course and at the end make the balance.
  - The aspects to be evaluated are for example:
    - Ratio of approved students
    - Quality of ratings
    - Amount of work done by teachers
    - Problems that have arisen and how they have been resolved
    - Possible updating of content,
    - Duplication and overlap with other subjects, etc.

Thanks!

70 YEARS

Technical University of Sofia

We succeed!

www.tu-sofia.bg