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: 3.2

Title of the Deliverable: Adaptation AIISM - AIISM Integration in curricula

Task/WP related to the Deliverable:

Type (Internal or Restricted or Public): Internal

Author(s): Yevgeniya Sulema, Liubov Drozdenko

Partner(s) Contributing:

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 :	

Contractual Date of Delivery to the CEC:

Actual Date of Delivery to the CEC:

Context

WP 3	Adaptation of AIISM to specific curricula in PC
WPLeader	NTUU-KPI
Task 3.2	Adaptation AIISM - AIISM Integration in curricula
Task Leader	NTUU-KPI
Dependencies	
Starting date	
Release date	

Author(s)	Yevgeniya Sulema
Contributor(s)	Natalia Dychka, Liubov Drozdenko, Yurii Bukhtiirov, Anatolii Orlov, Mykola Onai, Olena Temnikova, Michael Seyfarth, Juan V. Capella, Radu Dobrin
Reviewers	

History

Version	**Date**	**Author**	**Comments**
0.8	2014/03/10	Yevgeniya Sulema	Preliminary version
0.85	2014/04/01	Liubov Drozdenko	Improved version
0.9	2014/04/13	Nataliia Dychka	Pre-final version
0.92	2014/05/21	Juan V. Capella	Improved version
0.93	2014/06/15	Michael Seyfarth	First complete version
0.95	2014/09/11	Yevgeniya Sulema, Yurii Bukhtiirov, Mykola Onai, Anatolii Orlov, Olena Temnikova	Improved pre-final version
1.0	2014/09/25	Radu Dobrin	Preliminary version

Table of Contents

1 Executive summary

This deliverable presents the report on the integration of AIISM in the curricula of Master Programs at the Faculty of Applied Mathematics that is responsible for MEDIS Project in the National Technical University of Ukraine “Kyiv Polytechnic Institute”.

2 Possibilities for the Integration of AIISM Courses

According to the analysis of both the proposed AIISM courses (fulfilled on the basis of available Deliverables of WP1, but not WP2) and the curricula of three programmes (Master Programme “Systems Software”, Master Programme “Software Engineering”, Bachelor Programme “Software Engineering”) fulfilled in WP3.1 the following possibilities and obstacles for the Integration of AIISM Courses have been discovered:

1. The duration of the proposed courses is 15 weeks and the duration of each semester in the National Technical University of Ukraine “Kyiv Polytechnic Institute” is 18 weeks. Thus, 3 weeks can be used for additional lectures if necessary.
2. Both Master Programs (“Systems Software” and “Software Engineering”) can be used as the basis for AIISM implementation.
3. The proposed AIISM courses can be integrated into curricula as courses of a variable part of a Master Program. According to current regulations a variable part equals to 19 credits of ECTS and may include several alternatives (elective courses). Thus, the proposed courses can be considered as elective courses to be chosen by students.
4. The Bachelor Programme “Software Engineering” doesn’t provide prospective Master students with sufficient knowledge in electronics (in particular in analog electronics) necessary for some AIISM courses. To overcome this knowledge gap the additional 3 weeks mentioned above can be used for introductory lectures.
5. The Bachelor Programme “Software Engineering” provides prospective Master students with good knowledge in computer architecture that will meet requirements of AIISM courses.
6. The Bachelor Programme “Software Engineering” provides prospective Master students with good knowledge in programming that will fully satisfy requirements of AIISM courses.

3 Program Proposal of EU Partners

3.1 Industrial Computers module

The Industrial Computers is an AIISM module structured with different activities. These activities are developed during 3 hours/day (during 15 weeks) through a PBL methodology, using as a case study the example of the control of the liquids tank. To develop the course, 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 (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.

Based in the previous proposals, the set of chapters to group different topics is the following:

1. Introduction to industrial informatics
2. Computer architecture
3. Project management
4. Software development
5. Process interface
6. Graphical user interface
7. Task scheduling
8. Regulation strategies
9. Integration and validation

Chapters 1 and 2 introduce basics about computer architecture and the applicability of computers to industry.

Another basic of an engineer is the correct management of a project. This is the objective of the chapter 3 that is spread along the course. This is also a horizontal content of the module, so it is spread along the course and in a position where student understands its implications.

Chapter 4 develops skills on C programming to be applied on the application creation. This is a horizontal requirement of the module.

Chapter 5 deals with the connection of the computer to the real world, the so called “process interface”. This is set tends to motivate the student because he/she sees the interaction with physical reality.

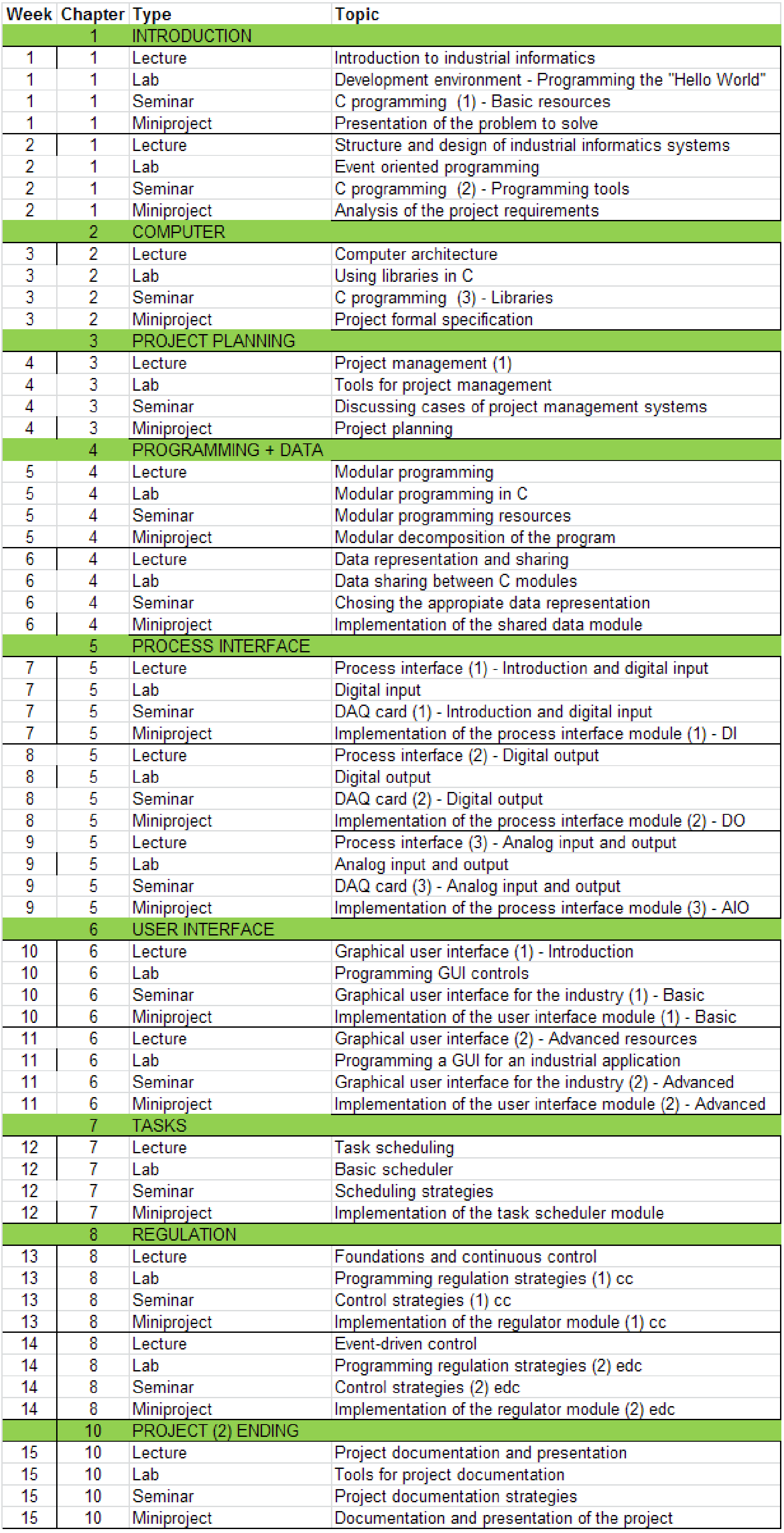
Taking into consideration that the actual student’s generation is accustomed to stunning visual user interfaces. Chapter 6 is in place for introducing another motivating set of activities related to this aspect.

At this point, it is necessary to start coordinating actions inside the application. So chapter 7 introduces the very basics around task coordination/scheduling.

And, finally, the student needs to see that your development works. From the point of view of the teacher, it is adequate to introduce here the regulation problem according to chapter 8.

A serious project of industrial informatics needs an investing on testing of each piece and integration. This important task is in chapter 9.

The scheduling distribution in weeks for this module is shown in Figure 1.



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

Figure 1. Scheduling of the Industrial Computers AIISM module

About the module assessment [WP1.4 UPV], we will collect all grades earned along the continuous assessment developed along the course and proceed to obtain the final grade for the module.

To do this we establish the following 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 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.

With all the information of ratings and percentages described will get a single grade for each student.

In order to integrate the Industrial Computer module in the curricula of National Technical University of Ukraine "Kyiv Polytechnic Institute", the programs of the Faculty of Applied Mathematics has been analyzed and jointly with the study realized in the deliverable WP3.1 [WP31 NTUU-KPI] we can conclude that the master programs that fits better are the "Systems Software" and "Software Engineering".

From the previous integration analyses the following possibilities and obstacles for the integration of the Industrial Computer module have been discovered:

- The Industrial Computer module can be integrated into the cited programs as a course of a variable part of a Master Program, conforming an intensification. According to current regulations a variable part equals to 19 credits of ECTS and may include several alternatives (elective courses).
- The duration of the Industrial Computer module is 15 weeks and the duration of each semester in the NTUU-KPI is 18 weeks. Thus, 3 weeks can be used for additional lectures if necessary.
- The students that course the defined program will obtain the "Information control systems and technologies" master degree.
- The Bachelor Programme "Software Engineering" doesn't provide prospective Master students with sufficient knowledge in electronics (in particular in analog electronics) necessary for the Industrial Computer module. To overcome this knowledge gap the additional 3 weeks mentioned above can be used for introductory lectures to guarantee the appropriate background of all the students.
- The Bachelor Programme "Software Engineering" provides prospective Master students with good knowledge in computer architecture that will meet requirements of the Industrial Computer module.
- The Bachelor Programme "Software Engineering" provides prospective Master students with good knowledge in programming that will fully satisfy requirements of Industrial Computer module.
- The proposed evaluation method for Industrial Computer AISSM module fits with the SPBSPU regulations.

In this line, the Industrial Computer module can be integrated into the “Systems Software” program (of 120 ECTS which structure is shown in Table 1) in the 1st year (autumn or spring semesters) or the 2nd year (autumn semester) covering an elective course slot.

Table 1. NTUU-KPI “Systems Software” master structure

First Year			
First Semester		**Second Semester**	
Course Title	*Credit*	*Course Title*	*Credit*
Corporate Information Systems Design 1	3	Corporate Information Systems Design 2	5,5
Technology of Software Systems Engineering	4	Mathematical Programming	3
Multimedia Technology	6	Modern Operating Systems	3
Applied Tasks of Software Engineering 1	1,5	Applied Tasks of Software Engineering 2	1,5
Information Security Means	3	Digital Signals and Images Processing	3
Applied Programming Technology 1	3	Applied Programming Technology 2	4
Fundamentals of Scientific Researches	2	Scientific Researches in Software Engineering 1	4
Models and Algorithms of Artificial Intelligence	5	Pedagogy of Higher Educational Institutions	1,5
Foreign Language for Profession (Advanced) 1	1,5	Foreign Language for Profession (Advanced) 2	1,5
Civil Defence	1	Factors of Successful Professional Career	1
		Labour Protection in the Branch	1
		Intellectual Property	1
		Marketing	1,5
Total	***30***	***Total***	***31,5***
Second Year			
Third Semester		**Fourth Semester**	
Course Title	*Credit*	*Course Title*	*Credit*
Object-oriented Analysis, Design and Development of Software Systems	4	Scientific Research Practical	6
High-Performance Computing	5	Preparation of Master's Thesis	22,5

Theory of Formal Languages and Compilation	3		
Operations Research	4		
Mathematical Modelling of Systems and Processes	5		
Scientific Researches in Software Engineering 2	2		
Fundamentals of the Society Sustainability	2		
Philosophic Problems of Scientific Knowledge	1,5		
Science of Patent and Copyright	2		
Foreign Language for Profession (Advanced) 3	1,5		
Total	***30***	***Total***	***28,5***
Total for the Programme	***120***		

Alternatively, the Industrial Computer module can be integrated into the 1st or 2nd year of the “Software Engineering” program of 120 ECTS shown in Table 2 covering an elective course slot.

Table 2. NTUU-KPI “Software Engineering” master structure

First Year			
First Semester		**Second Semester**	
Course Title	*Credit*	*Course Title*	*Credit*
Modelling and Design of Information Systems 1	3	Modelling and Design of Information Systems 2	5,5
Technology of Software Systems Engineering	4	Mathematical Programming	3
3D Modelling and Visualisation	6	Modern Operating Systems	3
Applied Tasks of Software Engineering 1	1,5	Applied Tasks of Software Engineering 2	1,5
Information Security Means	3	Digital Signals and Images Processing	3
Web-Programming and Web-Services 1	3	Web-Programming and Web-Services 2	4
Fundamentals of Scientific Researches	2	Scientific Researches in Software Engineering 1	4

Methods of Cryptographic Protection	5	Pedagogy of Higher Educational Institutions	1,5
Foreign Language for Profession (Advanced) 1	1,5	Foreign Language for Profession (Advanced) 2	1,5
Civil Defence	1	Factors of Successful Professional Career	1
		Labour Protection in the Branch	1
		Intellectual Property	1
		Marketing	1,5
Total	***30***	***Total***	***31,5***
Second Year			
Third Semester		**Fourth Semester**	
Course Title	*Credit*	*Course Title*	*Credit*
Object-oriented Analysis, Design and Development of Software Systems	4	Scientific Research Practical	6
Supercomputer Calculations	5	Preparation of Master's Thesis	22,5
Theory of Formal Languages and Compilation	3		
Operations Research	4		
Mathematical Modelling of Systems and Processes	5		
Scientific Researches in Software Engineering 2	2		
Fundamentals of the Society Sustainability	2		
Philosophic Problems of Scientific Knowledge	1,5		
Science of Patent and Copyright	2		
Foreign Language for Profession (Advanced) 3	1,5		
Total	***30***	***Total***	***28,5***
Total for the Programme	***120***		

3.2 Microcontroller module

The above mentioned module is an AIISM module containing different learning activities. These activities are taught during 5 hours/day, one day of the week (during 15 weeks) through a PBL methodology, using as a case study the example of the control of a liquids tank. The total supervised contact time is 75 hours during the semester. The total workload for the students is approximately 150 hours, resulting in 5 credit points (ECTS).

To successfully run through the course, students have to apply the knowledge acquired from the lectures and the laboratory practices. The proposed learning activities are the following:

- Lecture: lecturer presents main ideas of lecture contents and proposes some application problems which student solves individually (1 h contact time).
- Laboratory session: To implement (1 h 45'') a practical problem previously presented during lecture. Students work by teams of two students. Contact time is given by a technician and the lecturer.
- Seminars: a panel discussion with student teams (4 students) lasting 45 minutes is proposed, consisting generally of presenting the solution for a problem, which previously was analysed by the student team. The lecturer leads the discussion and summarizes the main results.
- Mini-project: dedicated to planning, design and development of the control system of an industrial production and transportation process. Teams of 4 students work on the mini-project during 2 hours/week (supervised by a technician and partly by a lectruer). Independent work of about another 2-3 hours/week advance the mini-project progressively.

Based on the previous proposals, the set of chapters to group different topics is the following:

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

Chapter 1 focuses on an introduction to microcontrollers, sample Applications, definition of Basic concepts and important terms.

Chapter 2 deals with project-management. There is no lecture for this topic, as it is not the main focus of this course. The contents of this chapter are worked out by seminars about project management methods and project documentation strategies.

Chapter 3 focuses on the I/O-system of microcontrollers. The interfaces of the microcontroller interact with the process directly. The chapter introduces the different kinds of input- and output-signals and their programming in the microcontroller.

Chapter 4 introduces timer and interrupts. In addition this chapter deals with the concepts of programming timer and interrupts using the microcontroller Arduino Due.

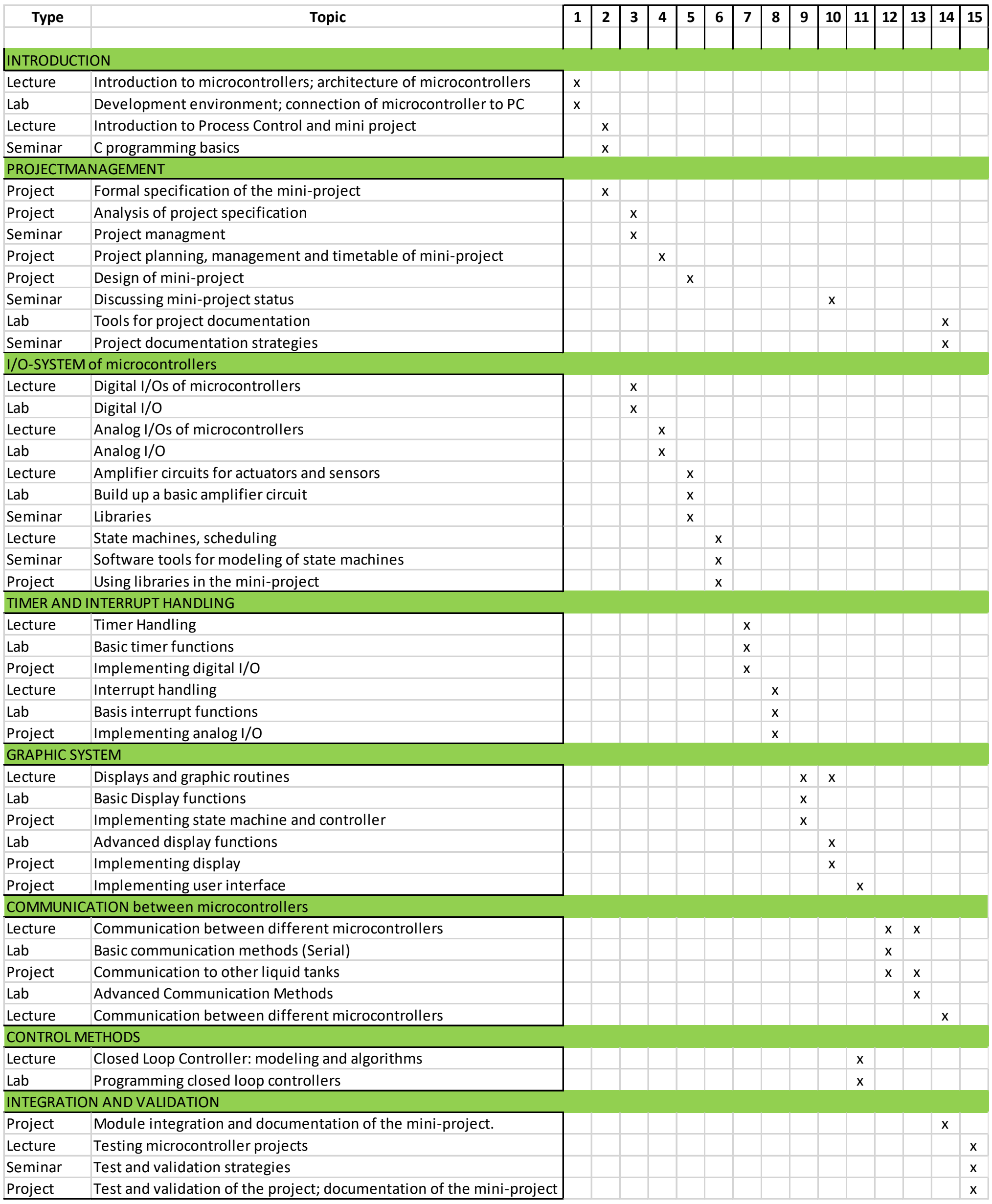
Chapter 5 deals with graphical user interfaces for microcontrollers. The main part of this chapter focuses on a graphical TFT-display wired to the microcontroller Arduino Due. In detail the necessary libraries and functions are explained.

Chapter 6 gives a short introduction to concepts of communication between microcontrollers. This lecture focuses on special communication mechanisms used with microcontrollers.

Chapter 7 introduces algorithms of closed loop control. Key feature of closed loop control is the recirculation of a current value and comparison with a desired value. There are different types of controllers – their mathematical models will be explained.

Chapter 8 deals with the integration and validation of the mini-project. There is no lecture for this topic. The contents of this chapter are worked out by a seminar about test and validation strategies. In addition the students learn directly by developing the mini-project and integrating and documenting their own mini-project.

Figure 2 shows the scheduling of this module spread to 15 weeks.



Type	Topic	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
INTRODUCTION																
Lecture	Introduction to microcontrollers; architecture of microcontrollers	x														
Lab	Development environment; connection of microcontroller to PC	x														
Lecture	Introduction to Process Control and mini project		x													
Seminar	C programming basics		x													
PROJECTMANAGEMENT																
Project	Formal specification of the mini-project		x													
Project	Analysis of project specification			x												
Seminar	Project managment			x												
Project	Project planning, management and timetable of mini-project				x											
Project	Design of mini-project					x										
Seminar	Discussing mini-project status										x					
Lab	Tools for project documentation														x	
Seminar	Project documentation strategies														x	
I/O-SYSTEM of microcontrollers																
Lecture	Digital I/Os of microcontrollers			x												
Lab	Digital I/O			x												
Lecture	Analog I/Os of microcontrollers				x											
Lab	Analog I/O				x											
Lecture	Amplifier circuits for actuators and sensors					x										
Lab	Build up a basic amplifier circuit					x										
Seminar	Libraries					x										
Lecture	State machines, scheduling						x									
Seminar	Software tools for modeling of state machines						x									
Project	Using libraries in the mini-project						x									
TIMER AND INTERRUPT HANDLING																
Lecture	Timer Handling							x								
Lab	Basic timer functions							x								
Project	Implementing digital I/O							x								
Lecture	Interrupt handling								x							
Lab	Basis interrupt functions								x							
Project	Implementing analog I/O								x							
GRAPHIC SYSTEM																
Lecture	Displays and graphic routines									x	x					
Lab	Basic Display functions									x						
Project	Implementing state machine and controller									x						
Lab	Advanced display functions										x					
Project	Implementing display										x					
Project	Implementing user interface											x				
COMMUNICATION between microcontrollers																
Lecture	Communication between different microcontrollers												x	x		
Lab	Basic communication methods (Serial)												x			
Project	Communication to other liquid tanks												x	x		
Lab	Advanced Communication Methods													x		
Lecture	Communication between different microcontrollers														x	
CONTROL METHODS																
Lecture	Closed Loop Controller: modeling and algorithms											x				
Lab	Programming closed loop controllers											x				
INTEGRATION AND VALIDATION																
Project	Module integration and documentation of the mini-project.														x	
Lecture	Testing microcontroller projects															x
Seminar	Test and validation strategies															x
Project	Test and validation of the project; documentation of the mini-project															x

Figure 2. Scheduling of the Microcontroller module

Deliverable [WP1.4 USTUTT] describes the details of grading the students in the module. The overall grade is calculated by different grades, earned along the continuous assessment of the students during the whole module.

For the microcontroller module the following proportion of grading between the different learning activities ensures a fair rating for differentiating the individual acquisition of knowledge and skills of the students:

- The evaluation of the student attitude (A): 10% of the final score.
- The evaluation of the miniproject (MP) represents 40% of the final score.
- 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.

In order to integrate the Industrial Computer module in the curricula of National Technical University of Ukraine “Kyiv Polytechnic Institute”, the programs of the Faculty of Applied Mathematics has been analyzed and jointly with the study realized in the deliverable WP3.1 [WP31 NTUU-KPI] it can be concluded that the master programs that fit best are the “Systems Software” and “Software Engineering” programs.

From the previous integration analyses the following possibilities and obstacles for the integration of the Microcontroller module have been discovered:

- The Microcontroller module can be integrated into the cited programs as a course of a variable part of a Master Program, conforming an intensification. According to current regulations a variable part equals to 19 credits of ECTS and may include several alternatives (elective courses).
- The duration of the Microcontroller module is 15 weeks and the duration of each semester in the NTUU-KPI is 18 weeks. Thus, 3 weeks can be used for additional training skills. For example the results of the miniprojects of different student teams can be combined to an integrated shop floor. This would additionally strengthen the PBL approach.
- The students that choose the defined program will obtain the “Information control systems and technologies” master degree.
- The Bachelor Programme “Software Engineering” doesn’t provide prospective Master students with sufficient knowledge in electronics (in particular in analog electronics) and industrial processes necessary for the Microcontroller module. To overcome this knowledge gap the additional 3 weeks mentioned above can be used for introductory lectures to guarantee the appropriate background of all the students.
- The Bachelor Programme “Software Engineering” provides prospective Master students with good knowledge in computer architecture that will meet requirements of the Microcontroller module.
- The Bachelor Programme “Software Engineering” provides prospective Master students with good knowledge in programming that will fully satisfy requirements of the Microcontroller module.

- The proposed evaluation method for the Microcontroller module fits with the SPBSPU regulations.

Consequently the Microcontroller module can be integrated into the “Systems Software” program (of 120 ECTS which structure is shown in Table 3) in the 1st year (autumn or spring semesters) or the 2nd year (autumn semester) covering an elective course slot.

Table 3. NTUU-KPI “Systems Software” master structure

First Year			
First Semester		**Second Semester**	
Course Title	*Credit*	*Course Title*	*Credit*
Corporate Information Systems Design 1	3	Corporate Information Systems Design 2	5,5
Technology of Software Systems Engineering	4	Mathematical Programming	3
Multimedia Technology	6	Modern Operating Systems	3
Applied Tasks of Software Engineering 1	1,5	Applied Tasks of Software Engineering 2	1,5
Information Security Means	3	Digital Signals and Images Processing	3
Applied Programming Technology 1	3	Applied Programming Technology 2	4
Fundamentals of Scientific Researches	2	Scientific Researches in Software Engineering 1	4
Models and Algorithms of Artificial Intelligence	5	Pedagogy of Higher Educational Institutions	1,5
Foreign Language for Profession (Advanced) 1	1,5	Foreign Language for Profession (Advanced) 2	1,5
Civil Defence	1	Factors of Successful Professional Career	1
		AIISM courses (elective)	4-5
		Labour Protection in the Branch	1
		Intellectual Property	1
		Marketing	1,5
Total	***30***	***Total***	***31,5***
Second Year			
Third Semester		**Fourth Semester**	
Course Title	*Credit*	*Course Title*	*Credit*
Object-oriented Analysis, Design	4	Scientific Research Practical	6

and Development of Software Systems			
High-Performance Computing	5	Preparation of Master's Thesis	22,5
Theory of Formal Languages and Compilation	3		
Operations Research	4		
Mathematical Modelling of Systems and Processes	5		
AIISM courses (elective)	4-5		
Scientific Researches in Software Engineering 2	2		
Fundamentals of the Society Sustainability	2		
Philosophic Problems of Scientific Knowledge	1,5		
Science of Patent and Copyright	2		
Foreign Language for Profession (Advanced) 3	1,5		
Total	***30***	***Total***	***28,5***
Total for the Programme	***120***		

Alternatively, the Industrial Computer module can be integrated into the 1st or 2nd year of the "Software Engineering" program of 120 ECTS shown in Table 4 covering an elective course slot.

Table 4. NTUU-KPI "Software Engineering" master structure

First Year			
First Semester		**Second Semester**	
Course Title	*Credit*	*Course Title*	*Credit*
Modelling and Design of Information Systems 1	3	Modelling and Design of Information Systems 2	5,5
Technology of Software Systems Engineering	4	Mathematical Programming	3
3D Modelling and Visualisation	6	Modern Operating Systems	3
Applied Tasks of Software Engineering 1	1,5	Applied Tasks of Software Engineering 2	1,5
Information Security Means	3	Digital Signals and Images Processing	3

Web-Programming and Web-Services 1	3	Web-Programming and Web-Services 2	4
Fundamentals of Scientific Researches	2	Scientific Researches in Software Engineering 1	4
Methods of Cryptographic Protection	5	Pedagogy of Higher Educational Institutions	1,5
Foreign Language for Profession (Advanced) 1	1,5	Foreign Language for Profession (Advanced) 2	1,5
Civil Defence	1	Factors of Successful Professional Career	1
		AIISM courses (elective)	4-5
		Labour Protection in the Branch	1
		Intellectual Property	1
		Marketing	1,5
Total	***30***	***Total***	***31,5***
Second Year			
Third Semester		**Fourth Semester**	
Course Title	*Credit*	*Course Title*	*Credit*
Object-oriented Analysis, Design and Development of Software Systems	4	Scientific Research Practical	6
Supercomputer Calculations	5	Preparation of Master's Thesis	22,5
Theory of Formal Languages and Compilation	3		
Operations Research	4		
Mathematical Modelling of Systems and Processes	5		
AIISM courses (elective)	4-5		
Scientific Researches in Software Engineering 2	2		
Fundamentals of the Society Sustainability	2		
Philosophic Problems of Scientific Knowledge	1,5		
Science of Patent and Copyright	2		
Foreign Language for Profession (Advanced) 3	1,5		
Total	***30***	***Total***	***28,5***
Total for the Programme	***120***		

3.3 Mobile and Cloud Computing module

This module is an AIISM module structured with different activities. Similar to the other modules, the activities are developed during 3 hours/day (during 15 weeks) through a PBL methodology, using as a case study the example of the remote control of the liquids tank. To develop the course, 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 (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.

Based in the previous proposals, the set of chapters to group different topics is the following:

1. Fundamentals of Remote Monitoring and Control
2. Integrated Development Environment
3. Basic App Development
4. Graphical user interface (GUI)
5. Security
6. Reliability

The first 2 lectures will give an introduction to the fundamentals of remote monitoring and control of embedded systems as well as the liquid tank system which will be used throughout the course. Lectures 3 and 4 will give an overview of a number of integrated development environments (IDEs) for developing apps that will be deployed on the most common operating systems, i.e. IOS and Android. Lectures 5 and 6 lecture will cover practical aspects of developing a basic app in one of the platforms using an example. The main feature of the app will be inter-device communication using Bluetooth and WiFi. The basics for designing a functional and intuitive graphical user interface will be covered in lectures 7 and 8. It will provide knowledge on the programming of GUI controls as well as addressing the limited bandwidth issues that can occur while dealing with media streaming. Security and Reliability will be addressed in lectures 9-12 that will cover the security related issues that a system connected to the internet might experience, suggest solutions and discuss the role based access control approach as well as 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. The last lecture will be a seminar for discussions of the research finding in the selected topics.

Additionally, the labs are the practical exercises that follow the corresponding lectures that help the students to acquire basic set of skills related to the topic of the lecture. Each lecture will be followed by a seminar where the student teams present the outcomes of their course work, as well as submit a written report, as well as miniprojects 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.

The schedule distribution in weeks for this module is shown in Table 5.

Table 5. Schedule of the Remote Monitoring and Control Module

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 (Anita)
3	Seminar	IDE for mobile devices
3	Laboratory	IDE introduction - installation and usability
3	Miniproject	Design and structuring of control application
4	Lecture	Cross-platform development 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 (Anita)
5	Seminar	Research inter-device communication
5	Lab	Basic app development II
5	Miniproject	Implement basic control logic
6	Lecture	Inter-device communication 2 (Anita)
6	Seminar	Research synchronisation in distributed systems.
6	Lab	GUI development and implementation
6	Miniproject	Implement complex control logic
4 – Graphical User Interface (GUI)		
7	Lecture	Graphical User Interface (GUI) 1 (Anita)
7	Seminar	Structure of code for GUIl
7	Lab	GUI development and implementation II
7	Miniproject	Add control logic to GUI
8	Lecture	Graphical User Interface (GUI) 2 (Anita)
8	Seminar	Propose structure of code implementing wireless communication on micro-controller
8	Lab	Implement wireless communication with the PC
8	Miniproject	Mapping of physical I/O to mobile devices
5 – Security		

9	Lecture	Security in mobile communication
9	Seminar	Research application areas of secure wireless communication
9	Lab	Access control 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 control and synchronization mechanisms II
10	Miniproject	Secure sending and receiving of messages.
6 – Reliability		
11	Lecture	Reliability in mobile 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 communication
13	Laboratory	Testing approaches for mobile communication
13	Miniproject	Determine latency of traffic in miniproject.
7 – Research findings		
14	Lecture	Research on dependable mobile communication 1 (All)
14	Seminar	Research on app controlled ES
14	Laboratory	App control framework
14	Miniproject	Add hierarchical supervisory control of distributed control application.
15	Lecture	Research on dependable mobile communication 2 (All)
15	Seminar	Research on reliable communication for embedded control systems
15	Laboratory	Final project demonstration
15	Miniproject	Presentation of the project(s)

About the module assessment [WP1.4 MDH], we will collect all grades earned along the continuous assessment developed along the course and proceed to obtain the final grade for the module.

To do this we establish the following 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 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.

With all the information of ratings and percentages described will get a single grade for each student.

3.4 Industrial Networks and Fieldbuses module

The Industrial Networks and Fieldbuses module has been prepared as a 15 week long 3 ECTS course, corresponding to a total workload of 81 hours. This workload is distributed between classes 3 hour long (totalling 45 hours), and individual study (comprising 36 hours total). Class hours include lectures, seminars and assisted laboratory, and individual study includes independent laboratory work (mini-project). The division of effort between these 4 activity types will differ from week to week depending on the subject matter and the expected output for the mini-project work for that week.

1. Lecture and problems: lecturer presents main ideas of lecture contents and proposes some application problems which students solve individually (0h45 to 1h30).
2. Seminars: a panel discussion by teams of 4 students, 0h45 long, consisting research related to communication protocols, or to the problem to be solved later during laboratory and/or mini-project session.
3. Laboratory session: Lasting from 0h45 to 1h30, the students (in teams of 2) implement a practical problem previously presented during the lecture, following the guidance provided by the lecturer.
4. Mini-project: dedicated to planning, designing, development and configuration of several industrial communication protocols, and using these protocols in a distributed automated control system. The mini-project is performed by teams of 4 students over an average of 2h00 each week.

As mentioned in previous documents, this module is organized in the following chapters:

1. Introduction
2. Modbus/TCP
3. Discrete Event Control
4. Modbus/RTU
5. CAN
6. CAN Open
7. Hierarchical Control

Chapter 1 (Introduction to Industrial Filedbuses and Networks) is taught over the first 2 weeks (weeks 1 and 2), and introduces the field of industrial communication networks, providing an overview of how these are organized.

Chapter 2 focuses on the Modbus protocol in general, and its implementation over the TCP/IP stack in particular. During these 2 weeks the students are expected to implement a very simple control algorithm for 3 conveyors that only use *binary* sensors and actuators.

Chapter 3 also takes 2 weeks (weeks 5 and 6), and during this period the students become acquainted with methodologies for modeling complex discrete event control algorithms, and how to implement these in software.

In chapter 4 the serial version of the Modbus protocol is introduced over weeks 7 and 8. This serial Modbus protocol is used for the communication between an Arduino and the computer running both the simulated plant floor and the discrete event control program. This involves implementing the Modbus serial protocol on both the PC as well as the Arduino in the laboratory sessions. The buttons and the lights on the Arduino are then used during the mini-project sessions as a physical interface with an operator, and the control algorithm is extended to take into account a RUN and STOP state.

In chapter 5, corresponding to weeks 9 and 10, the students come into contact with the CAN fieldbus and use it to establish a network of Arduino devices. This network is used to simulate remote discrete Input/Outputs, and the students are expected to integrate these remote I/Os as an extra physical interface to the operator (RUN, STOP and PAUSE buttons, and GREEN and RED lights).

In chapter 6 the students are asked to implement the CAN-Open protocol over the CAN network, and to use it in the mini-project. This is done during weeks 11, 12 and 13.

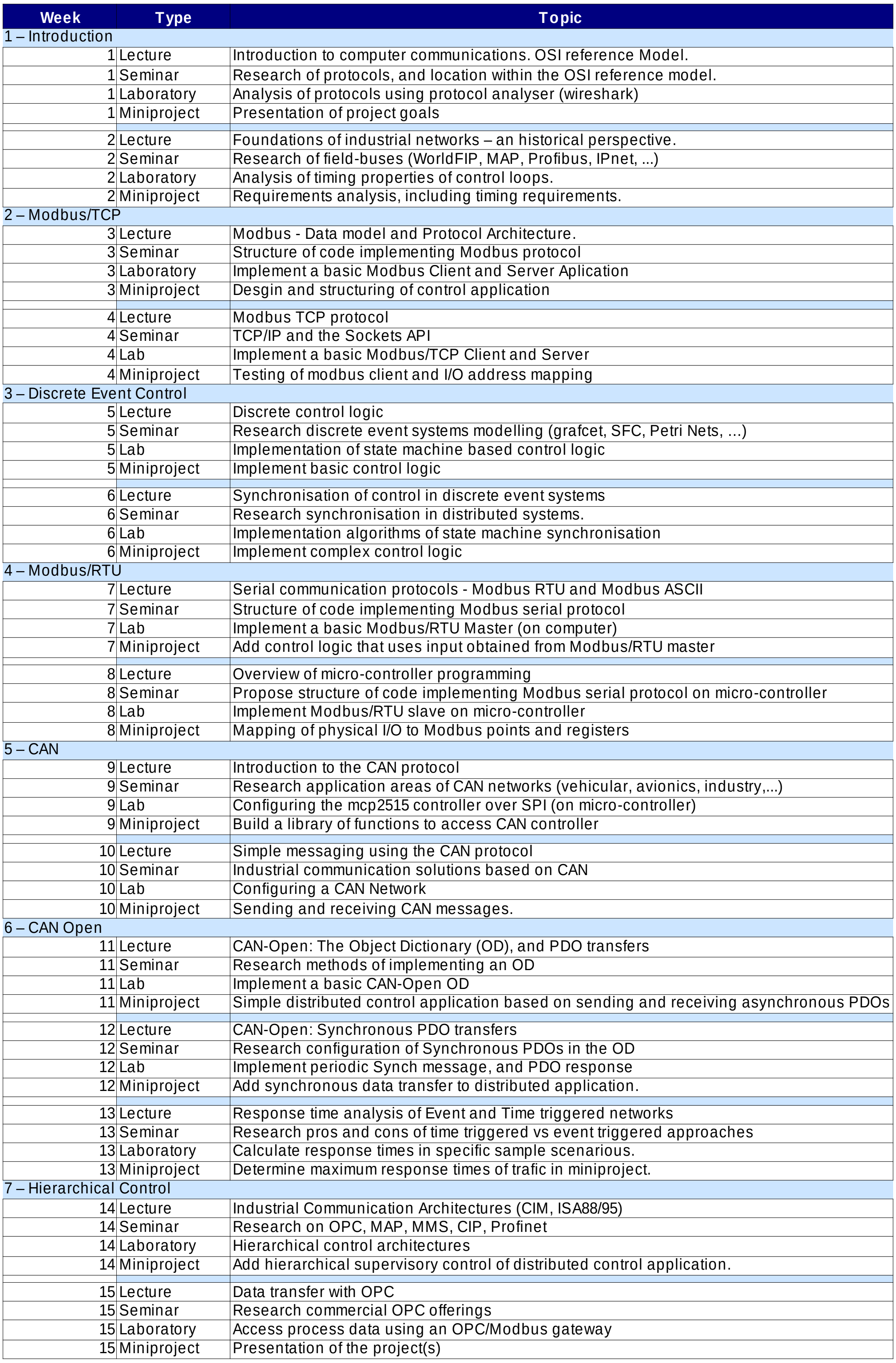
In chapter 7 the industrial networks module focuses on hierarchical control architectures, and the communication protocols used in this capacity. In the first of two weeks (week 14) the students are asked to build a small SCADA based (Supervisory Control And Data Acquisition) graphical user interface (GUI) for a plant floor supervisor and/or operator. The mini-project session of week 15 (the last week of the module) is used for the presentations of the mini-project work.

The scheduling of the classes for this module are summarised Figure 3.

Grading and assessing the individual work of each student is done on a continuous bases, throughout the 15 weeks of the module.

The final grade of each student is obtained from a weighted average of the following evaluation criteria: :

- evaluation of the student attitude (A), 10% of the final grade.
- evaluation of the miniproject (MP), 45% of the final grade.
- evaluation of the Laboratory work (L): 30% of the final grade.
- evaluation of the Seminars (S): 15% of the final score.



Week	Type	Topic
1 – Introduction		
1	Lecture	Introduction to computer communications. OSI reference Model.
1	Seminar	Research of protocols, and location within the OSI reference model.
1	Laboratory	Analysis of protocols using protocol analyser (wireshark)
1	Miniproject	Presentation of project goals
2	Lecture	Foundations of industrial networks – an historical perspective.
2	Seminar	Research of field-buses (WorldFIP, MAP, Profibus, IPnet, ...)
2	Laboratory	Analysis of timing properties of control loops.
2	Miniproject	Requirements analysis, including timing requirements.
2 – Modbus/TCP		
3	Lecture	Modbus - Data model and Protocol Architecture.
3	Seminar	Structure of code implementing Modbus protocol
3	Laboratory	Implement a basic Modbus Client and Server Aplication
3	Miniproject	Desgin and structuring of control application
4	Lecture	Modbus TCP protocol
4	Seminar	TCP/IP and the Sockets API
4	Lab	Implement a basic Modbus/TCP Client and Server
4	Miniproject	Testing of modbus client and I/O address mapping
3 – Discrete Event Control		
5	Lecture	Discrete control logic
5	Seminar	Research discrete event systems modelling (grafcet, SFC, Petri Nets, ...)
5	Lab	Implementation of state machine based control logic
5	Miniproject	Implement basic control logic
6	Lecture	Synchronisation of control in discrete event systems
6	Seminar	Research synchronisation in distributed systems.
6	Lab	Implementation algorithms of state machine synchronisation
6	Miniproject	Implement complex control logic
4 – Modbus/RTU		
7	Lecture	Serial communication protocols - Modbus RTU and Modbus ASCII
7	Seminar	Structure of code implementing Modbus serial protocol
7	Lab	Implement a basic Modbus/RTU Master (on computer)
7	Miniproject	Add control logic that uses input obtained from Modbus/RTU master
8	Lecture	Overview of micro-controller programming
8	Seminar	Propose structure of code implementing Modbus serial protocol on micro-controller
8	Lab	Implement Modbus/RTU slave on micro-controller
8	Miniproject	Mapping of physical I/O to Modbus points and registers
5 – CAN		
9	Lecture	Introduction to the CAN protocol
9	Seminar	Research application areas of CAN networks (vehicular, avionics, industry,...)
9	Lab	Configuring the mcp2515 controller over SPI (on micro-controller)
9	Miniproject	Build a library of functions to access CAN controller
10	Lecture	Simple messaging using the CAN protocol
10	Seminar	Industrial communication solutions based on CAN
10	Lab	Configuring a CAN Network
10	Miniproject	Sending and receiving CAN messages.
6 – CAN Open		
11	Lecture	CAN-Open: The Object Dictionary (OD), and PDO transfers
11	Seminar	Research methods of implementing an OD
11	Lab	Implement a basic CAN-Open OD
11	Miniproject	Simple distributed control application based on sending and receiving asynchronous PDOs
12	Lecture	CAN-Open: Synchronous PDO transfers
12	Seminar	Research configuration of Synchronous PDOs in the OD
12	Lab	Implement periodic Synch message, and PDO response
12	Miniproject	Add synchronous data transfer to distributed application.
13	Lecture	Response time analysis of Event and Time triggered networks
13	Seminar	Research pros and cons of time triggered vs event triggered approaches
13	Laboratory	Calculate response times in specific sample scenarious.
13	Miniproject	Determine maximum response times of trafic in miniproject.
7 – Hierarchical Control		
14	Lecture	Industrial Communication Architectures (CIM, ISA88/95)
14	Seminar	Research on OPC, MAP, MMS, CIP, Profinet
14	Laboratory	Hierarchical control architectures
14	Miniproject	Add hierarchical supervisory control of distributed control application.
15	Lecture	Data transfer with OPC
15	Seminar	Research commercial OPC offerings
15	Laboratory	Access process data using an OPC/Modbus gateway
15	Miniproject	Presentation of the project(s)

Figure 3. Scheduling of the Industrial Networks and Fieldbuses AIISM module

3.5 Process Controllers and Simulators module

The Process Controllers and Simulators module in AISSM course is structured with different activities. These activities are developed during 4 hours/day (during 15 weeks) through a PBL methodology, using as a case study the example of the control of the liquids tank. The learning sessions are organized in these activities: lectures, seminars, laboratories, mini-project and tutorship.

The proposed learning activities are the following:

- **Lectures** - the first step in the learning process for each of the topics in a module. The lecturer presents the main topics, basic knowledge and the structure of the contents. This includes some application examples. Some lectures include elements of general theory not directly included in the exercises and mini-projects but very important for the applications.
- **Laboratory sessions (labs)** - 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 lab solve specific and well-defined problems; they are guided, fully documented, and of progressively increasing complexity. The lab provides students with a set of tools and skills that can be used to solve more open problems during the seminars.
- **Seminars** - During the seminars the students must solve problems on the topic of the lecture. They have already collected experiences on related topics and procedures in the previous laboratories.
- **Mini-projects** - During the mini-project students use the knowledge and skills that they have acquired in the lectures, labs and seminars to develop the couple controller-simulator for a physical process in an integral way. 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 combine seminar designs of different other teams to solve their mini-project.

Based in the previous proposals, the set of chapters to group different topics is the following:

1. **Introduction**

Classification and characteristics of the Computer Control Systems (CCS): embedded (specialized) systems; control systems for industrial applications with standardized functions.

2. **Architecture of Computer Control Systems**

Functional organization of the modern hierarchical industrial control systems.

Types of computer control and data acquisition systems: data collection systems, supervisor control, direct numerical control, logical controllers, etc. Modern, decentralized and distributed control systems.

3. **Organization and structure of computers for control purposes**

- Organization and structure of computers for the industrial controller and for embedded system. Analog and discrete I / O subsystems; analog and discrete control peripherals.
- Organization of computational processes in CCS for continuous control. Concept of static and dynamic process scheduling.

- Organization of computational processes in CCS interacting with discrete objects: implementation of synchronous, asynchronous and synchronous-asynchronous state machines.

4. Basic control algorithms

Controllers for analog objects - standard functions and algorithms; concept of configuration vs coding - specialized languages for continuous control system. Controllers for discrete objects - logical and sequential controllers.

5. **Real-Time software environment**

Real-Time operating systems - functions and subsystems; management processes (tasks). Scheduling in hard real-time constraints.

6. **SCADA**

Basic structure. Functions. Standards. Connection to the controllers. Interfaces.

7. **Simulators – general theory**

Software-in-the-Loop simulators. Hardware-in-the-Loop simulators. Agent-based simulators.

8. **Simulators – practical aspects**

Computer simulators including process periphery. Connecting the controller to the simulator. Setting-up the simulator. Induction of errors and special situations. Keeping the history of the process. Analysis encountered in real operation problems. Training of the personnel.

9. Simulation of distributed objects and control systems

Virtual monomachine approach. Component approach. Communication network influences – simulation.

10. **Simulators validation**

Validation using the "Configure/Reconfigure" approach.

11. **Real-Time system improvement using simulation environment**

Model improvements. Software improvements. Performance optimization.

The course has been scheduled assuming duration of 15 weeks, with **4** hours of direct teacher student interaction per week.

Another basic of an engineer is the correct management and documentation of a project. This is the objective of the following activities that are spread along the course. This is also a horizontal content of the module, so it is spread along the course and in a position where student understands its implications.

The scheduling distribution in weeks for this module is shown in the Table 6.

Table 6. Schedule of the Process Controllers and Simulators module

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									
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			
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

About the module assessment [WP1.4 TUS], at this level we will collect all grades earned along the continuous assessment developed along the course and proceed to obtain the final grade for the course.

To do this 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.

With all the information of ratings and percentages described will get a single grade for each student.

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

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

Distribution of percentages for final grade is shown in Figure 4.

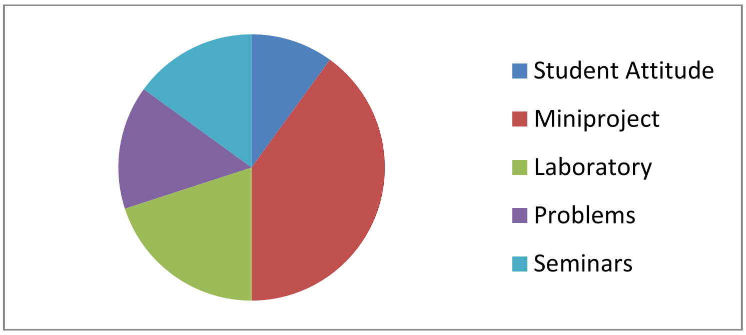


Figure 4. Distribution of percentages for final grade

4 Integration of AIISM Courses into Curricula at NTUU “KPI”

4.1 Option 1

As Option 1 AIISM Courses can be integrated into Curricula instead of existing courses:

- In the curriculum of the Master Programme 8.05010301 “Systems Software” – instead of disciplines “Information Security Means”, “Applied Tasks of Software Engineering 1”, “Applied Tasks of Software Engineering 2” (Table 7).

Table 7

First Year			
First Semester		***Second Semester***	
Course Title	*Credit*	*Course Title*	*Credit*
Corporate Information Systems Design 1	3	Corporate Information Systems Design 2	5,5
Technology of Software Systems Engineering	4	Mathematical Programming	3
Multimedia Technology	6	Modern Operating Systems	3
Programming of Industrial Computers 1	1,5	**Programming of Industrial Computers 2**	1,5
Programming of Microcomputers	3	Digital Signals and Images Processing	3
Applied Programming Technology 1	3	Applied Programming Technology 2	4
Fundamentals of Scientific Researches	2	Scientific Researches in Software Engineering 1	4
Models and Algorithms of Artificial Intelligence	5	Pedagogy of Higher Educational Institutions	1,5
Foreign Language for Profession (Advanced) 1	1,5	Foreign Language for Profession (Advanced) 2	1,5

Civil Defence	1	Factors of Successful Professional Career	1
		Labour Protection in the Branch	1
		Intellectual Property	1
		Marketing	1,5
Total	***30***	***Total***	***31,5***
Second Year			
Third Semester		***Fourth Semester***	
Course Title	*Credit*	*Course Title*	*Credit*
Object-oriented Analysis, Design and Development of Software Systems	4	Scientific Research Practical	6
High-Performance Computing	5	Preparation of Master's Thesis	22,5
Theory of Formal Languages and Compilation	3		

Continuation of Table 7

Operations Research	4		
Mathematical Modelling of Systems and Processes	5		
Scientific Researches in Software Engineering 2	2		
Fundamentals of the Society Sustainability	2		
Philosophic Problems of Scientific Knowledge	1,5		
Science of Patent and Copyright	2		
Foreign Language for Profession (Advanced) 3	1,5		
Total	***30***	***Total***	***28,5***
Total for the Programme	***120***		

- In the curriculum of the Master Programme 8.05010302 “Software Engineering” – instead of disciplines “Information Security Means”, “Applied Tasks of Software Engineering 1”, “Applied Tasks of Software Engineering 2”, “Digital Signals and Images Processing” (Table 8).

Table 8

First Year			
First Semester		***Second Semester***	
Course Title	*Credit*	*Course Title*	*Credit*
Modelling and Design of Information Systems 1	3	Modelling and Design of Information Systems 2	5,5
Technology of Software Systems Engineering	4	Mathematical Programming	3
3D Modelling and Visualisation	6	Modern Operating Systems	3
Programming of Controllers and Simulators 1	1,5	**Programming of Controllers and Simulators 2**	1,5
Industrial Networks and Fieldbuses	3	**Mobile and Cloud Computing**	3
Web-Programming and Web-Services 1	3	Web-Programming and Web-Services 2	4

Continuation of Table 8

Fundamentals of Scientific Researches	2	Scientific Researches in Software Engineering 1	4
Methods of Cryptographic Protection	5	Pedagogy of Higher Educational Institutions	1,5
Foreign Language for Profession (Advanced) 1	1,5	Foreign Language for Profession (Advanced) 2	1,5
Civil Defence	1	Factors of Successful Professional Career	1
		Labour Protection in the Branch	1
		Intellectual Property	1
		Marketing	1,5
Total	***30***	***Total***	***31,5***
Second Year			
Third Semester		***Fourth Semester***	
Course Title	*Credit*	*Course Title*	*Credit*
Object-oriented Analysis, Design and Development of Software Systems	4	Scientific Research Practical	6
Supercomputer Calculations	5	Preparation of Master's Thesis	22,5
Theory of Formal Languages and Compilation	3		
Operations Research	4		
Mathematical Modelling of Systems and Processes	5		
Scientific Researches in Software Engineering 2	2		
Fundamentals of the Society Sustainability	2		
Philosophic Problems of Scientific Knowledge	1,5		
Science of Patent and Copyright	2		
Foreign Language for Profession (Advanced) 3	1,5		
Total	***30***	***Total***	***28,5***
Total for the Programme	***120***		

4.2 Option 2

As Option 2 AIISM Courses can be integrated into Curricula as the alternative Module by student's choice:

- In the curriculum of the Master Programme 8.05010301 "Systems Software" – Table 9.

Table 9

First Year			
First Semester		***Second Semester***	
Course Title	*Credit*	*Course Title*	*Credit*
Module 1 by student's choice (KPI disciplines)			
Applied Tasks of Software Engineering 1	1,5	Applied Tasks of Software Engineering 2	1,5
Applied Programming Technology 1	3	Applied Programming Technology 2	4
Information Security Means	3		
Total	***7,5***	***Total***	***5,5***
Module 2 by student's choice (AIISM disciplines)			
Programming of Industrial Computers	2,5	**Programming of Controllers and Simulators**	2,5
Programming of Microcomputers	2,5	**Mobile and Cloud Computing**	3
Industrial Networks and Fieldbuses	2,5		
Total	***7,5***	***Total***	***5,5***

- In the curriculum of the Master Programme 8.05010302 "Software Engineering" – Table 10.

Table 10

First Year			
First Semester		***Second Semester***	
Course Title	*Credit*	*Course Title*	*Credit*
Module 1 by student's choice (KPI disciplines)			
Applied Tasks of Software Engineering 1	1,5	Applied Tasks of Software Engineering 2	1,5

Continuation of Table 10

Web-Programming and Web-Services 1	3	Web-Programming and Web-Services 2	4
Information Security Means	3		
Total	***7,5***	***Total***	***5,5***
Module 2 by student's choice (AIISM disciplines)			
Programming of Industrial Computers	2,5	**Programming of Controllers and Simulators**	2,5
Programming of Microcomputers	2,5	**Mobile and Cloud Computing**	3
Industrial Networks and Fieldbuses	2,5		
Total	***7,5***	***Total***	***5,5***

5 Adaptation of EU Proposal to NTUU "KPI" Curricula

The analysis of the options presented above enabled to develop the Curricula based on the Option 2. Thus, the proposed AIISM disciplines are grouped as the Module 2 by student's choice and included into existing Curricula with the distribution of academic hours shown at the Table 11.

Table 11

Subject	**Semester**	**ECTS Credits**	**Academic Hours per Semester**					**Academic Hours per Week**		
			Total	**Contact Hours**	**Lectures**	**Laboratory Lessons**	**Unsuper-vised Work**	**1 Semester**	**2 Semester**	**3 Semester**
Programming of Industrial Computers	**1**	**3**	**90**	**54**	**36**	**18**	**36**	**3**		
Programming of Controllers and Simulators	**2**	**3**	**90**	**54**	**36**	**18**	**36**		**3**	
Programming of Microcomputers	**2**	**3**	**90**	**54**	**36**	**18**	**36**		**3**	
Mobile and Cloud Computing	**2**	**3**	**90**	**54**	**36**	**18**	**36**		**3**	
Industrial Networks and Fieldbuses	**3**	**3**	**90**	**54**	**36**	**18**	**36**			**3**

As it is presented in the Table 11, every subject has equal number of ECTS credits, namely 3 ECTS credits that is an equivalent of 90 academic hours. This total number of academic hours is divided to 54 contact hours and 36 academic hours of unsupervised student’s work. Contact hours include 36 academic hours of lectures and 18 academic hours of laboratory lessons.

Contact hours of 54 academic hours per a semester correspond to 3 academic hours per a week (the detailed information on education process at NTUU “KPI” is presented in Section 6 of this document) that include 2 academic hours per a week for lectures and 1 academic hour per a week for laboratory lessons. Usual practice for lessons scheduling in NTUU “KPI” is that laboratory lessons are scheduled as 2 academic hours per 2 weeks, what gives the same total number of laboratory lessons.

The distribution of AIISM disciplines within the developed curricula is that the majority of subjects are scheduled for the 2nd (spring) semester of the 1st year of the Master Program. This distribution has been fulfilled according to the analysis of AIISM presented in the paper “Intellectual Tutoring System for Implementation of MEDIS Project Problem-Based Learning Methodology” by authors Dr. Yevgeniya Sulema, Olena Temnikova that has been published in the proceedings of the 15th International Scientific Conference named after T. A. Taran “Intellectual Analysis of Information” (IAI-2015) [1].

6 Implementation of ECTS

The procedure of implementation of European Credit Transfer and Accumulation System (ECTS) into educational process at NTUU “KPI” has been started in 2006 according to the Rector’s Order № 1-150 dated 09.10.2006. The regulations on ECTS implementation at NTUU “KPI” are available at the university web-site (in Ukrainian) – http://kpi.ua/kmsonp. According to these regulations the scale A, B, C, D, E, FX, and F is used in NTUU “KPI”. At the same time, traditional grading system (“excellent”, “good”, “satisfactory”, and “unsatisfactory” for exams; “passed” and “failed” for tests) is used as well. The correspondence between total rating of a student, traditional grades, and ESCT grades is shown in Table 12. The total rating is a sum of points obtained by a student during the studying of a certain subject. Points can be obtained for a student’s work on mini-projects, laboratory tasks, practical lessons, seminars, etc.

Table 12

Total Rating (TR)	ECTS Grades	Traditional Grades	
		For Exams	For Tests
$95 \leq \text{TR} \leq 100$	A	Excellent	Passed
$85 \leq \text{TR} < 94$	B	Good	
$75 \leq \text{TR} < 84$	C		
$65 \leq \text{TR} < 74$	D	Satisfactory	
$60 \leq \text{TR} < 64$	E		
$0 \leq \text{TR} < 60$	Fx	Unsatisfactory	Failed
Learning activities haven’t been completed	F	Not admitted	

According to the educational reform started in 2015 one academic year corresponds to 60 ECTS credits that are equivalent to 1800 academic hours of study, i.e. 1 ECTS credit corresponds to 30 academic hours (1 academic hour is equal to 45 minutes). Before the reform 1 ECTS credit corresponded to 36 academic hours.

Usually one subject includes from 3 to 5 ECTS credits (90-150 academic hours). However either less or more number of ECTS credits is acceptable as well. An average subject of 3 ECTS credits (90 academic hours) includes typically 36 academic hours of lectures, 18 academic hours of practical lessons (laboratory lessons), and 36 academic hours of student’s unsupervised work.

The duration of a semester in NTUU “KPI” is 18 weeks. An examination session lasts 2 weeks and includes 3 exams. In addition, students pass tests on subjects that don’t include exams. Thus, every subject finishes either with test or with exam.

7 Evaluation system

According to the internal regulations of NTUU “KPI” an evaluation system description is a part of a teacher’s documentation prepared for every subject of the curriculum. This evaluation system description includes detailed information about components of student’s rating, criteria of evaluation, rules of rating calculation, etc. Standard practice in NTUU “KPI” is to use 100-point scale for the evaluation system of every subject. The main objective of the evaluation system is to encourage students to work actively and continuously during a whole semester as well as to ensure fair evaluation of student’s learning results.

This existing document can be adapted for implementation of the evaluation system proposed by EU partners. To achieve this objective as well as to satisfy demands of the internal regulations of NTUU “KPI” the following actions should be fulfilled for every subject of MEDIS section of the curriculum:

1. The set of student’s activities should be defined.
2. The maximum number of points of the final score should be defined.
3. The maximum number of points within the maximum final score should be assigned for every activity.
4. Clear and monosemantic criteria of evaluation should be formulated for every component of the evaluation system.

Since the proposed learning activities within MEDIS methodology are the following:

- Lecture
- Problems for individual work of a student
- Laboratory session
- Seminar
- Mini-project,

they are used as a basic set of student’s activities for every subject of MEDIS section of the curriculum. To obtain a final set of activities, the proposed proportion between the different sections to ensure a fair rating for differentiating individual acquisition of knowledge and skills against the student group work should be taken into consideration:

- 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.

Thus, the student attitude (A) can be applicable to “Lecture” activity in the basic set, what gives us the final activity to be evaluated – “Active attitude of the student on lectures”, what means student’s questions and participation in discussions during a lecture. The maximum number of points for activity is 10, what is 10% of 100 points. The criteria of evaluation for this activity can be the following:

- The student gets from 8 to 10 points if he or she participates actively in every lecture – asks questions, participates in discussions (active attitude).
- The student gets from 5 to 7 points if he or she participates actively in discussions or asks questions in more than a half of lectures during the semester (quite active attitude).
- The student gets from 2 to 4 points if he or she participates actively in discussions or asks questions in less than a half of lectures during the semester (rather active than passive attitude).
- The student gets 1 point if he or she doesn’t participate in discussions or ask questions (passive attitude).
- The student gets 0 point if he or she doesn’t attend lectures.

The evaluation of the mini-project representation (MP) is applicable to “Mini-project” activity in the basic set, what gives us the final activity to be evaluated – “Fulfilment and presentation of mini-project”, what allows to evaluate a student’s work from the very beginning of the work on the mini-project to presentation of this project. The maximum number of points for activity is 40, what is 40% of 100 points. Since this activity is complex, it should be evaluated by several criteria to be defined separately. The final score for this activity is a sum of points according to every criterion. The list of criteria can be the following:

- Quality of the mini-project fulfilment – it gives the student from 1 to 10 points according to the project’s quality level.
- Ability to work in a team – it gives the student from 1 to 10 points according to effectiveness of the student’s work.
- Quality of the report documentation on the mini-project – it gives the student from 1 to 10 points according to the project documentation’s quality level.
- Quality of the presentation of the mini-project – it gives the student from 1 to 10 points according to the project presentation’s quality level.

These criteria should be applied to every student in the project team individually.

The evaluation of the Laboratory (L) can be applicable to “Laboratory session” activity in the basic set, what gives us the final activity to be evaluated – “Quality of student’s work during laboratory sessions”, what includes different aspects of the student’s work during laboratory session. The maximum number of points for activity is 20, what is 20% of 100 points. Since this activity is also complex, it should be evaluated by two criteria to be defined separately. The final score for this activity is a sum of points according to every of two criterion. The criteria are as follows:

- Quality of the laboratory tasks fulfilment – it gives the student from 1 to 10 points according to the quality level.
- Quality of the answers during fulfilled task discussion – it gives the student from 1 to 10 points according to the student’s answers quality level.

The evaluation of the Problems (P) can be applicable to “Problems for individual work of a student” activity in the basic set, what gives us the final activity to be evaluated – “Quality of student’s individual work on given problems”, what means quality student’s solution of given individual tasks. The maximum number of points for activity is 15, what is 15% of 100 points.

The criteria of evaluation for this activity can be the following:

- The student gets from 14 to 15 points if problems solved correctly, the solution is explained and illustrated well, there are no mistakes of typos.
- The student gets from 11 to 13 points if problems solved correctly, but the solution is explained and illustrated too short, but there are no mistakes of typos.
- The student gets from 7 to 10 points if problems solved almost correctly, but there are some insignificant mistakes of typos.
- The student gets from 3 to 6 points if problems solved, but there are significant mistakes of typos.
- The student gets from 1 to 2 points if problems don’t solved.
- The student gets 0 point if solutions are missing.

The evaluation of the Seminars (S) can be applicable to “Seminar” activity in the basic set, what gives us the final activity to be evaluated – “Active attitude of the student on seminars”, what means student’s participation in discussions during a seminar. The maximum number of points for activity is 15, what is 15% of 100 points. The criteria of evaluation for this activity can be the following:

- The student gets from 13 to 15 points if he or she participates actively in every seminar (active attitude).
- The student gets from 8 to 11 points if he or she participates actively in discussions in more than a half of seminars during the semester (quite active attitude).
- The student gets from 4 to 7 points if he or she participates actively in discussions in less than a half of lectures during the semester (rather active than passive attitude).
- The student gets from 2 to 3 points if he or she rarely participates in discussions

(rather passive than active attitude).

- The student gets 1 point if he or she doesn't participate in discussions (passive attitude).
- The student gets 0 point if he or she doesn't attend seminars.

The final score of the student is calculated as a sum of his or her individual score for every of 5 evaluation elements. The final student's grade is defined according to Table 6.

8 Supporting documents

By decision of the Academic Council of the Faculty of Applied Mathematics of NTUU "KPI" the MEDIS section of 5 subjects is recommended for including into Curricula of Master Program "Systems Software" and Master Program "Software Engineering" in 2015/2016 academic year (Annex).

9 Conclusion

There are two options for the integration of AIISM courses into curricula at NTUU "KPI":

1. Replacement of some existing courses by AIISM courses;
2. Including AIISM courses into the curriculum as alternative module of disciplines by student's choice.

The final decision on more appropriate option can be made only after reviewing of teaching materials developed as the result of WP2.

The slight change of some AIISM courses titles is necessary:

- "Programming of Industrial Computers" instead of "Industrial Computers"
- "Programming of Microcomputers" instead of "Microcomputers"
- "Programming of Controllers and Simulators" instead of "Controllers and Simulators"

in order to adapt to the demands of existing Master Programmes ("Systems Software" and "Software Engineering") at NTUU "KPI".

10 References

1. Yevgeniya Sulema, Olena Temnikova, (2015) "INTELLECTUAL TUTORING SYSTEM FOR IMPLEMENTATION OF MEDIS PROJECT PROBLEM-BASED LEARNING METHODOLOGY", Proceedings of the 15th International Scientific Conference "Intellectual Analysis of Information" (IAI-2015), Kiev, Ukraine.
2. Pfeifer N. E., Jarassova G. S., Ispulov N. A., Ospanova N. N., (2014) "MEDIS Structure of Educational Program Industrial Informatics", HERALD of Pavlodar State University, (p. 150-158), Pedagogika, 2, 2014.

3. Galchonkov, O., & Loziienko, N. (2015). “Advanced problem-based learning. The experience of the European universities participants to the TEMPUS MEDIS project”, Odes’kyi Politechnichnyi Universytet. Pratsi, 1.
4. Houcine Hassan, Juan M. Martínez, Carlos Domínguez, Angel Perles, Juan V. Capella, José Albaladejo (2015). “m-IC: a Mobile Device based Multimedia Learning Methodology for Industrial Computing”, INTERNATIONAL JOURNAL OF ENGINEERING EDUCATION, Journal Citation Reports. Accepted for publication, May, 2, 2015.
5. Yessengaliyeva Zh., Mussiraliyeva Sh. (2015) “A Methodology for the Formation of Highly Qualified Engineers at Masters Level in the Design and Development of Advanced Industrial Information Systems”. Proceedings of the conference “Integration of education, science and business as the basis of innovation development of economy”, Almaty, Kazakhstan, 2015.
6. Book (2015): “Engineering Experiences in the Design of Advanced Industrial Informatics Systems”, ISBN: 978-84-606-9807-4, Authors: J.V. Busquets, J. Albaladejo, A. Perles, J.V. Capella, C. Domínguez, J.M. Martinez, H. Hassan, Mário Sousa, Luis Almeida, Paulo Portugal, Armando Sousa, Slobodanka Cenevska, Mahnaz Malekzadeh, Abhilash Thekkilakattil and Radu Dobrin, Michael Seyfarth, Yessengaliyeva Zhanna, Mussiraliyeva Shynnar, O.N. Galchonkov, N.V. Loziienko.