1. Course Title (Course Code)

Special Programming Exercises (2247)

2. Instructor

Markon Sandor

3. Term

Spring 1

4. Outline and Objectives

An important area of programming for both developed and developing countries is using embedded systems, also called dedicated computers, that can help solving a wide range of technical and social issues. Some examples include wireless sensor networks for remote data collection or disaster prevention, RFID for inventory control, human body sensors for remote medicine etc. When compared to common PCs, embedded systems use different hardware and software platforms, and to use them require understanding of special concepts like direct hardware I/O (input-output), interrupts, real-time control, cross-development environments etc.

By finishing this course, the student will be able to recognize opportunities of effectively using embedded systems to solve ICT4D issues, and to propose and initiate projects, based on concrete experience with embedded systems.

5. Goals (Attainment Targets)

(1) Become able to recognize suitable application areas and make proper choices for embedded systems.
(2) Become able to design for reliability, maintainability, safety, and security in developing embedded systems.
(3) Can develop for an embedded platform and use its development environment.
(4) Can develop programs with embedded concepts like hardware control, sensors, actuators, timers, interrupts etc., through hands-on experience.
(5) Find and solve a development problem using embedded systems.

6. Correspondence relationship between Educational goals and Course goals

Educational goals			Course goals
High level ICT skill	Basic academic skills		(1), (2)
	Specialized knowledge and literacy		(3), (4), (5)
Human skill (Tankyu skill)	Ability to continually improve own strengths		(2)
	Ability to discover and resolve the problem in society	Problem setting	(5)
		Hypothesis planning	
		Hypothesis testing	
		Practice	
	Fundamental Competencies for Working Persons	Ability to step forward	(5)
		Ability to think through	(5)
		Ability to work in a team	(5)
Professional ethics			(2)

7. Course Requirements

Computer Programming Exercises (achievement of attainment targets is required)

8. Textbooks

None. This course uses only material freely available on the Internet (links will be provided on Moodle).

9. Reference Books

None

10. Evaluation

Goals	Evaluation method & point					
	term-end exam	quiz	report	presentation	deliverable	other
(1)			○			
(2)				○	○	
(3)			○	○	○	
(4)			○	○	○	
(5)				○	○	
Allocation			30	30	40	

11. Notes

Students not comfortable with basic electric circuit concepts (currents and voltages, Ohm's law etc.) are advised to take part in voluntary lab sessions before taking the course.

Course Schedule

(Notice) This schedule is a tentative plan, there might be changes, additions, revisions etc. at the time of delivering the course.

Lesson 1: Orientation and Introduction (lecture and demonstration, 90min)

Review through examples the basic concepts of embedded systems, from TV remote controls and rice cookers to airplane controls and power plants; the reasons for using them, principles, problems, evolution, and current status. Explain the course plan: discussions, practice, work group forming, projects, evaluation, and further studies. Students are shown some of the devices used in the course.

Lesson 2: Motivation by Examples (lecture and group discussion, 90min)

Introduce examples of embedded systems in products and services through case studies. Student groups are formed, and based on understanding some typical applications, discuss other uses in fields familiar to them, listing up possible project topics, analyzing their merits and difficulties, their novelty, feasibility, profitability and sustainability.

Student groups start their discussions on their project topic.

Lesson 3: Platforms (a) – Arduino (practice, presentations, 90min)

Students learn to develop a simple application using the open source / open hardware Arduino platform as an example of working with a simple stand-alone embedded environment. Groups receive sets of Arduino boards, sensor and actuator devices, and after installing the development environment on their PCs, design, program, and test a simple embedded application according to the provided requirement specifications.

Student groups present their project proposals with Tankyu charts and overall plans.

Lesson 4: Platforms (a) - Arduino (continued, 90min)

Student groups review the possibilities available with the provided hardware and software tools, and compete to design the most surprising embedded application with their resources on Arduino.

Lesson 5: Review of embedded concepts (1) (lecture and group discussion, 90min)

Starting from their experience with the Arduino board, student groups discuss basic embedded system concepts, reviewing and analyzing the handling of I/O devices, timings, process flow etc. Motivation is given for searching for more advanced capabilities like interrupts, multi-threading, guaranteed real-time response etc.

Lesson 6: Platforms (b) - Raspberry Pi (practice, 90min)

Student groups receive Raspberry Pi boards and install Linux on them. By connecting them to the network, they learn how to use such a self-development environment on a small embedded platform. Students also discuss common features and differences between the Raspberry Pi and the Arduino environment, and try to find out advantages and limitations through reading the documentation and performing experiments.

Lesson 7: Platforms (b) - Raspberry Pi (continued, 90min)

Student groups re-build their Raspberry Pi Linux OS from source, and add a new device driver, file system, or other extra capability of their choice that was not present in the original distribution. Discuss the experience from the point of view of open source vs. proprietary systems.

Lesson 8: Review of embedded concepts (2) (lecture and group discussion, 90min)

Starting from their experience with the Raspberry Pi board, student groups discuss using an operating system like Linux for embedded systems, reviewing the advantages and drawbacks of isolating the hardware from the application, and of providing basic development capabilities on the deployed embedded system itself. Soft- and hard real-time OS concepts will be introduced here.

Lesson 9: Mid-term project review (presentations, mid-term exam, 90min)

Student groups present the progress of their embedded application projects, reviewing their current status and roadmap. Different groups share their experiences and exchange advices.

Lesson 10: Review of embedded concepts (3) (lecture and group discussion, 90min)

Students review and compare the platforms that they have explored, and develop their own guidelines for choosing the proper environment for different applications. These will consider not only technical features, but also economic and social factors, IP issues, job creation, long-term local sustainability etc.

Lesson 11: Workshop: Wireless Sensor Networks (1)

Visiting lecturers from ICTP introduce the concept of Wireless Sensor Networks (WSN) and their role in developing countries, through examples from their practice.

Lesson 12: Workshop: Wireless Sensor Networks (2)

Visiting lecturers demonstrate and explain using Arduino, Zigbee and other embedded platforms for WSN. Student groups explore creating and operating WSN using the provided equipment, and review their project topic in light of what they have learnt about WSN.

Lesson 13: Workshop: Wireless Sensor Networks (3)

Student groups discuss their ideas and application areas for WSN with visiting lecturers, and review their experience with experiments on Arduino and various sensors.

Lesson 14: Workshop: IoT topics for developing countries

Visiting lecturers review currently important Internet of Things topics for developing countries, including TV White Spaces, Delay Tolerant Networks (DTN) and successful applications of WSN.

Lesson 15: Final project review (presentations, 90min)

Each group presents its project results, demonstrating the device and system they have built, explaining how it will solve an actual problem of development, how it can be put into practice, and how can its effects be evaluated and verified. Each team member takes part in the presentations in turn.