CAEDMI
Central Asia Engineering Dentistry Medicine Institute

Alpha Institute of Technology

Course Syllabus

Course Title:	MeRo 5020: Microcontroller-based Control Systems
Lead Instructor:	Boris Kuznetsov, PhD
Contacts:	boris.kuznetsov@caedmi.com
Office hours:	Wednesday 13.00 - 14.00

1. Course Description

The course covers the fundamentals of digital embedded systems, the architecture of microcontrollers and microprocessors, programming techniques, and the application of various data transfer interfaces, as well as the architectural design of real-time systems and the control of actuators.

2. Basic Information

Course Academic Level:	Master-level
Course Semester:	1st Semester
Number of ECTS credits:	5
Course Prerequisites:	Knowledge of electrical circuits and linear systems control
Type of Assessment:	Graded

Mapping from grades to percentage:

Letter Grade	Numeric Value (GPA)	100-point Scale
A	4.0	93-100
A-	3.7	90-92
B+	3.3	87-89
B	3.0	83-86
B-	2.7	80-82
C+	2.3	77-79
C	2.0	73-76
C-	1.7	70-72
D+	1.3	67-69
D	1.0	63-66
D-	0.7	60-62
F	0.0	<60

3. Course Content

№ of week	Topic	Laboratory Sessions
1	Introduction: - Impact of microelectronics on society. - Digital systems and main development trends. - Embedded systems as a key direction for integrating digital systems into human society. - Issues in modern microelectronics. - Course structure, types of sessions, assignment requirements, and grading system	Introduction to Digital Systems and AVR Microcontrollers: Create a simple C program for AVR microcontroller to blink an LED.
2,3	Basics of Digital Systems: - Fundamental concepts: logic gate, flip-flop, register, finite state machine. - Turing machine and architectural concepts: Princeton and Harvard architectures. - Microcontroller and microprocessor architectures. - Fundamentals of RISC and CISC architectures.	Working with UART and SPI Interfaces: Develop a program to exchange data between two microcontrollers using UART and SPI interfaces.

	- Main types of architectures for modern microcontrollers (ARM, RISC-V, C51). - Basic principles of designing embedded systems and controllers. - The ecosystem concept for microcontroller families.	
4	AVR Microcontrollers and Arduino Platform: - Basic principles of platform construction, hardware diversity, and development environments. - Microcontroller architecture, peripheral blocks (timers, DAC, ADC, etc.). - Microcontroller ports, operating modes, and hardware implementation feature	Measuring Analog Signals Using ADC: Create a program to measure temperature using the LM35 sensor and display data on a serial monitor.
5,6	Microcontroller Programming Techniques: - Architecture of control programs for microcontrollers. - Main loop, external interrupts, timer operations, and real-time operating systems (FreeRTOS). - Programming basics, C/C++ language.	Controlling Servos Using PWM: Develop a program to control a servo using PWM.
7	Interfaces and Data Transfer Protocols in Embedded Systems: - UART interface, COM port, and RS232. - RS485 interface and Modbus implementation. - I2C and SPI interfaces. - Analog data transmission channel "current loop."	Introduction to Raspberry Pi and Using GPIO: Create a Python program to control an LED using GPIO.
8,9	DAC and ADC: - Features of converter construction. - Characteristics and conversion errors of converters. - Mathematical post-processing of the signal after ADC.	Working with Sensors on the Raspberry Pi Platform: Create a Python program to read data from a DHT22 temperature and humidity sensor.
10	Real-time Microcontrollers with DSP: - Purpose and architecture of TMS320 microcontroller family. - Specialized DSP blocks and their functions. - C programming for DSP.	Introduction to the Raspbian Operating System and Network Setup: Install Raspbian, configure Wi-Fi, and install necessary packages.
11	Architectural Design of Real-time Systems: - Signal filtering and processing algorithms for	Creating a Web Server on Raspberry Pi: Set up and

	robotics. - Code optimization for working with data from gyroscopes, accelerometers, and encoders.	configure a web server, create a simple web interface to control an LED.
12	PWM and Motor Control Using TMS320: - Implementation of PID controllers and more complex control algorithms. - Development of a servo control system	Introduction to F28002x LaunchPad Development Kit: Install the development environment, create a simple program to blink an LED.
13	Architecture of Single-board Computers (Raspberry Pi): - Features of programming single-board computers for embedded systems. - Introduction to the Raspbian operating system.	Working with ADC and DAC on F28002x LaunchPad: Create a program to measure and generate analog signals using ADC and DAC.
14	Hardware Capabilities of Raspberry Pi for Embedded Systems: - Using GPIO to control external devices. - Connecting and controlling peripheral devices. - Using Python and other languages for application development. - Libraries for working with GPIO, I2C, SPI, and other interfaces.	Motor Control Using PWM on F28002x LaunchPad: Develop a program to control a motor using PWM on the C2000 platform.
15,16	Embedded Systems and the Internet of Things (IoT): - Features and architectural solutions. - Integration of embedded solutions into external networks and use of external internet services. - Application of AI in the development and as an element of the overall architecture of embedded devices.	Implementing a PID Controller onF28002x LaunchPad:Create a program to implement a PID controller and control a system using it.

4. Learning Outcomes

- Understand the architecture and fundamental principles of microcontrollers and microprocessors.
- Apply microcontroller programming techniques using C/C++ languages to develop control applications.
- Develop and debug programs for various data transfer interfaces, such as UART, SPI, I2C.
- Utilize Raspberry Pi and similar single-board computers to create embedded systems and web servers.
- Understand and apply principles of digital systems design and embedded solutions for the Internet of Things (IoT).

- Implement microelectronics projects, such as automation systems, smart homes, and environmental monitoring systems.

5. Assignments and Grading

Assignment Type	% of Final Course Grade
Laboratory Work	40
Midterm Tests	20
Final Project	40

6. Assessment Criteria

Laboratory Work
Task Completion Accuracy: This includes evaluating the correctness of laboratory task implementation, adherence to technical requirements and specifications, and minimizing errors in code and circuit design.
Report Documentation: The quality and completeness of the laboratory report, including the description of methods used, results obtained, and conclusions drawn.
Teamwork: For group work, the contribution of each team member, interaction, and task distribution are evaluated.
Punctuality: Timely submission of laboratory work also affects the final grade.

Midterm Tests:
Competency in Theoretical Material: This will be assessed through test scores reflecting the understanding of key concepts covered in lectures and the ability to utilize them.
Problem-Solving: The capability to apply theoretical knowledge to solve practical problems in programming and microcontrollers.
Reasoning Quality and Justification: The quality of reasoning behind chosen solutions and the ability to explain the approach to problem-solving.

Final Project:
Complexity and Depth: Evaluation of the project's complexity, thoroughness in working out components, and how comprehensive the final solution is.
Functionality: Assessment of how accurately and reliably the developed system operates, and whether the project meets its objectives.
Innovation: The originality and novelty of the project idea and approach, along with the use of modern and innovative solutions.
Documentation Quality: The readiness and depth of project documentation, including technical implementation descriptions, diagrams, code, and testing results.
Project Presentation: The ability to clearly present the project, explain goals, approaches, and results, and effectively answer questions from the audience.

7. Textbooks and Internet Resources

Required Textbooks:

1) Lambert, Tyler Ross. "An introduction to microcontrollers and embedded systems." Auburn University. July 344 (2017).

Recommended Textbooks:

1) Schmidt, Maik. "Arduino: a quick-start guide." 2015.
2) Wallace, Shawn, Matt Richardson, and Wolfram Donat. Getting started with raspberry pi. Maker Media, Inc., 2021.

8. Facilities

Required Course Materials (software or equipment):

- Arduino IDE 2.x
- Code Composer Studio Version: 12.x
- Raspbian (Raspberry Pi OS)

Optional:

- GNU Octave

9. Additional Notes

Project Examples:
1. Logic Analyzer: Create a simple logic analyzer using AVR microcontroller and Arduino platform, including signal capture, analysis, and visualization.
2. Automated Lighting Control System: Develop an automated lighting control system using motion sensors and microcontrollers.
3. Smart Thermostat: Create a smart thermostat using a microcontroller and temperature sensor to regulate room temperature based on set parameters.
4. Robotic System Using PWM: Develop a robot using TMS320 microcontroller and PWM to control motors, including implementing PID controllers for precise movement control.
5. Environmental Monitoring System Based on Raspberry Pi: Create an environmental monitoring system using Raspberry Pi and various sensors (temperature, humidity, air quality).
6. Smart Home Based on IoT: Develop a smart home system using microcontrollers and single-board computers connected via the internet, including control of lighting, heating, and security.
7. Automated Plant Watering System: Create an automatic plant watering system using microcontrollers and soil moisture sensors.
8. Smart Security System: Develop a home security system using microcontrollers and motion, sound, and door/window opening sensors.
9. Energy Consumption Management System: Create a system for monitoring and managing energy consumption in a home or office using microcontrollers and single-board computers.
10. Smart Greenhouse: Develop an automatic system for managing conditions in a greenhouse using microcontrollers and sensors (temperature, humidity, lighting).

Practical Sessions:
- Weekly laboratory sessions related to lecture topics.
- Continuous project work starting from week 5.