<table>
<tr><td>Course Title:</td><td colspan="5">Real Time Embedded Systems</td></tr>
<tr><td>Course Code:</td><td colspan="5">ACOE343</td></tr>
<tr><td>Course Type</td><td colspan="5">Compulsory</td></tr>
<tr><td>Level</td><td colspan="5">Bachelor (1st Cycle)</td></tr>
<tr><td>Year/ Semester</td><td colspan="5">3rd Year/ 5th Semester</td></tr>
<tr><td>Teacher's Name</td><td colspan="5">Dr. Konstantinos Tatas</td></tr>
<tr><td>ECTS</td><td>6</td><td>Lectures/week</td><td>2</td><td>Laboratories/week</td><td>2</td></tr>
<tr><td>Course Purpose</td><td colspan="5">The purpose of the course is to introduce students to embedded system design concepts and tools. Distinguish between embedded and general-purpose computing systems in terms of requirements. Furthermore, develop the requirements and specifications for an embedded system. Finally, implement embedded systems using sensors, actuators and embedded processors.</td></tr>
<tr><td>Learning outcomes</td><td colspan="5">1. List the distinguishing differences between computers and embedded systems in terms of implementation and constraints.
2. Identify the key technologies involved in embedded systems and Internet-of-Things design
2. Demonstrate the concept of real-time processing and identify the strengths and limitations of the various types of processors and platforms used in embedded systems.
3. Produce efficient real-time designs using off-the-shelf processors.
4. Apply real-time algorithmic design techniques for embedded applications using assembly and C/C++ programming.
5. Implement, test and verify real-time control and DSP applications in laboratory experiments.</td></tr>
<tr><td>Prerequisites:</td><td colspan="2">ACOE201 and ACSC182</td><td>Corequisites:</td><td colspan="2">None</td></tr>
<tr><td>Course content</td><td colspan="5">• Embedded C/C++ and Assembly Languages: Embedded C and assembly for programming microcontrollers and DSPs. Real-Time Design Techniques. Software optimization techniques for constrained embedded systems
• Microcontrollers vs GPPs vs DSPs: Architecture and ISA of the microcontroller, General Purpose Processor and DSP.
• Sensors and Actuators: Two state and analog sensors and actuators</td></tr>
</table>

	employed in typical embedded systems. Interfacing through chip-to-chip serial buses and protocols such as the SPI, I2C and I2S. MEMS devices. • **Real-Time Embedded and IoT Applications:** Timing constrains for real time applications. Analog and Digital Input and Output, Remote Control and Sensing and Real-Time Data Acquisition. Energy issues and interrupts. Communication interfaces for IoT systems: wi-fi, Low Energy Bluetooth, LoRa, Zigbee. • **DSP Techniques:** FIR and IIR filter for real time applications. Digital modulation techniques and Applications. • **Laboratory Work:** Individual or small group experiments based on using a variety of EDA tools for programming, debugging and testing applications developed on common microcontroller boards
Teaching Methodology	The course is structured in lectures that are conducted with the help of both computer presentations and traditional means. Practical examples and exercises are included in the lectures to enhance the material learning process. Often short post-lecture quizzes are used to assess the level of student understanding and provide feedback. Student questions are addressed during the lecture, or privately after the lecture or during office hours. Lecture notes are available through the web for students to use in combination with the textbooks. Students are assessed continuously and their knowledge is checked through tests with their assessment weight, date and time being set at the beginning of the semester via the course outline. Furthermore, individual design assignments as well as a group embedded system design project are used to develop practical engineering skills. Laboratory experiments are carried out in small groups and lab reports are required two weeks after the laboratory class resulting in a cumulative mark. The first laboratory exercises are totally structured (cookbook) in order to familiarize the students with the equipment, while later exercises are less structured, allowing the student to create their own designs or programs for a given application. Laboratory exercises include interfacing with digital and analog sensors and actuators, digital filtering, and connecting an embedded system to the internet.
Bibliography	Textbooks: o Edward Ashford Lee and Sanjit Arunkumar Seshia, "**Introduction to Embedded Systems, a Cyber-Physical Systems Approach**", Second Edition, MIT Press, 2017 o Marilyn Wolf, "**Computers as components: Principles of embedded computing design**", 4th edition, Morgan Kaufmann Publishers, 2016 References:

	○ M. Margolis, **Arduino Cookbook**, O'Reilly, 2011.
Assessment	The final assessment of the students is formative and summative and is assured to comply with the subject's expected learning outcomes and the quality of the course. In order to continuously assess students, and given the practical nature of the course, coursework weight is set at 70%, which comprises assignments, a mid-term exam and laboratory work assessment. Assignments range from simple problems to work out, to circuit design assignments that require demonstrate concept understanding as well as problem-solving skills. The assessment weight, date and time of each type of continuous assessment is being set at the beginning of the semester via the course outline. An indicative weighted continuous assessment of the course is shown below: • Assignments 20% • Test: 10% • Laboratory Work: 40% • Final Exam: 30%
Language	English