King Fahd University of Petroleum & Minerals
College of Computer Sciences & Engineering
Computer Engineering Department

COE 306: Introduction to Embedded Systems

Syllabus — Term 172

Catalog Description

Introduction to Embedded Systems. Microcontroller Hardware. ARM Processor. CPU Programming. Memory and I/O. Interfacing: Parallel and Serial Communication. A/D and D/A conversion Embedded system design methodologies. Specifications. Designing robust software for embedded systems. RTOS features.

Prerequisite: COE 301 and COE 203

Instructor

Dr. Ahmad Khayyat. Office: 14/254 or 22/150. Phone: 1900 or 8234.
Website: `http://www.ccse.kfupm.edu.sa/~akhayyat/`.

See the Course Website section below for the preferred way of communicating with the instructor.

Office Hours

Sunday, Tuesday, Thursday 11:00 AM – 12:00 PM, or by appointment through calendar invitations.

For more details on how to schedule an appointment, visit `http://www.ccse.kfupm.edu.sa/~akhayyat/schedule/`.

Course Website

- `https://blackboard.kfupm.edu.sa/`
- Additionally, we will be using Piazza for discussion; a system for getting you help fast and efficiently from your classmates and the instructor. Rather than emailing questions, post your questions on Piazza.

  You can find the course page on Piazza using either of these two ways:

  1. From the course page on BlackBoard, go to the *Discussion* link on the left-hand side menu.
  2. Use the direct link of the course page on Piazza:
     `https://piazza.com/kfupm.edu.sa/spring2018/coe306/`

Textbook

Marilyn Wolf, *"Computers as Components: Principles of Embedded Computing System Design"*, Third Edition, Morgan Kaufmann, 2012.

The textbook is fully available online through the university library:

1. Go to the KFUPM library's website at `http://library.kfupm.edu.sa`
2. Click on the SDL link at the top menu
3. Login using your KFUPM account
4. Click on the large SDL banner in the middle of the page
5. Search for "Computers as Components" for the SDL website to recognize your KFUPM account
6. Pick the most relevant search result, or use this permanent link: Computers as Components, 3rd Edition, on SDL

Course Learning Outcomes

- Understand the design process and some of the related UML models.
- Understand the interrupt mechanism, and develop interrupt-driven C programs.
- Understand memory systems: cache mapping, virtual memory, and address translation.
- Understand various system buses, protocols, and peripheral interaction.
- Understand common standard peripheral interfaces.
- Implement and use various common software components of embedded systems.
- Understand the software compilation process and various basic compiler optimizations.
- Understand context switching and scheduling of periodic processes in real-time operating systems.
- Ability to analyze performance at the CPU, platform, and program levels.
- Ability to develop ARM-based bare-metal embedded software in C.

Grading Policy

Assignments	20%
Quizzes	10%
Exam I	10%
Exam II	10%
Final Exam	15%
Lab	20%
Project(s)	15%

- Attendance is required. The tenth unexcused absence results in a DN grade.
- Late assignments penalty: 10% of the assignment points per day, up to three days only.
- Cheating in one assignment or quiz will result in a zero in all assignments or quizzes. Any identical solutions are considered cheating.
- No makeup will be made for missed quizzes or exams.

Course Topics

- **Embedded Computing**

  Introduction to embedded systems; requirements analysis; specifications; design methodologies overview.
- **Microcontroller Organization**

  Computer architectures, ARM organization, ARM instruction set, data operations, control flow, PIC, TI DSPs.
- **CPUs**

  I/O and memory mapping, addressing modes, interrupts and traps, caches, co-processors, memory management unit, virtual memory, address translation, CPU performance, pipelining.
- **Embedded Platforms**

  Hardware and software components of embedded platforms, bus protocols, DMA, system bus configurations, the AMBA and AHB buses, memory devices, example embedded platforms, bandwidth, bus and memory performance, performance bottleneck.

- **Interfacing**

  Analog-to-digital conversion, digital-to-analog conversion, pulse-width modulation, serial communication, serial interfaces such as UART, SPI, and I2C.

- **Program Design and Analysis**

  State machines, circular buffers, queues, models of programs, the compilation process, program performance and optimization.

- **Processes and Operating Systems**

  Tasks and processes, process timing requirements, real-time operating systems, preemptive execution, context switching, scheduling processes, priority scheduling, inter-process communication, shared-memory systems, message passing.

- **Networks and Multiprocessors**

  Multiprocessor systems, distributed embedded systems, CAN bus, I^2C bus, multiprocessor system-on-chip (MPSoC), accelerators.

Lab Topics

- Getting familiar with the development platform
- General-purpose input/output (GPIO)
- Interrupts
- Hardware timers
- Analog input and output
- Pulse-width modulation
- Serial communication
- Building a microcontroller system on an FPGA