MARYLAND APPLIED GRADUATE ENGINEERING

Course:	ENPM 615 – Embedded Systems
Semester:	Spring 2020
Day(s):	Wednesday
Time:	7:00 – 9:40 pm
Location:	KEB 1110
Instructor:	Gang Qu
Phone:	301-405-6703
Email:	gangqu@umd.edu

Course Description

Prerequisite: basic concepts on computer organization, digital logic design, C, and assembly.
Web page: https://elms.umd.edu/
Instructors: Dr. Gang Qu Office: 1417 A.V. Williams Building
Office hours: Wednesday: 6:00 pm – 6:50 pm, 9:40 pm – 10:00 pm
open door policy and by appointment (email is the best way)

Grading Policy:

Homework/Project	45%	**(8-10 homework sets assigned every one or two weeks)**
Midterm Exam	25%	(a 100-minute in-class exam, tentatively April 6, KEB 1110)
Final Exam	**30%**	(a 2-hour in-class exam. Time and place:TBD).

Most **homework** assignments will come from the questions at the end of each chapter of the textbook. They will be assigned after the lecture every one or two weeks. Normally you will be given 1-2 weeks to complete the homework. Assigned homework questions will be made available to you on ELMS.

A couple of C programming homework sets will be assigned to help you improving your programming skills. You will need to complete the homework questions on GLUE UNIX using C.

The midterm exam is scheduled to be in-calss and you will have 100 minutes to answer the questions. **The final exam** will be in-class and last 2 hours. We will discuss about the exam format and coverage later in the semester. The time and location of the final exam will be scheduled by the university later.

Other Information:

- If you have a **documented disability** and wish to discuss academic accommodation, please contact Dr. Qu (in person or by email) as soon as possible and not later than Thursday, February 3, 2022.
- If the midterm or the final exam is scheduled on a **religious holiday** that you are compelled to observe, please contact Dr. Qu (in person or by email) no later than Thursday, February 3, 2022.
- **Academic dishonesty will not be tolerated**

The University Code of Academic Integrity, available at http://www.inform.umd.edu/CampusInfo/Departments/JPO/ prohibits students from committing the following acts of academic dishonesty: cheating, fabrication, facilitating academic dishonesty, and plagiarism. Academic dishonesty in this class also includes outright copying on homework; however, ***discussing homework problems and exchanging tips is permissible and also encouraged.*** *Instances of academic dishonesty will be referred to Office of Judicial Programs.*

Required/Recommended Textbooks

Lecture notes will be available on ELMS. There is no required textbooks, but the following two are highly recommended (some homework questions will come from the Wolf book):

- Marilyn Wolf, *Computers as Components: Principles of Embedded Computing System Design (3rd ed),* Morgan Kaufmann, 2012.

- James Peckol, *Embedded Systems: A Contemporary Design Tool*, Wiley and Sons, 2008.

Course Outline

Introduction to embedded systems

Review of C programming language

Review of basic digital logic design: combination circuit design and analysis, sequential circuit design and analysis.

Introduction to graph theory and algorithm design

CPU components and performance: Instruction sets; Input/output; Buses; Interrupts; Memory system; Caches; Pipelines

Program design and analysis: State machine models; Control/data flow graph; Program optimization techniques; Performance analysis

Processes/tasks and operating systems: Real-time kernels and OS; Real-time task scheduling; Interprocess communication; Power/energy optimization of processes

System design techniques: Microprocessor-based design; Interface to local and remote devices; Co-processors; Accelerators; Multiprocessors; System design flows, requirements, specifications, and quality assurance; UML

Embedded system security and trust (optional): Physical attacks; Side channel analysis; Trusted integrated circuit; Trusted platform module (TPM); Hardware security primitives; Hardware Trojan

Readings related to embedded systems, cyber physical systems, and Internet of Things (optional)

Philosophy of the course aka what we will and won't do in this course:

"As the first textbook on embedded system design, this book will provide invaluable as a means for acquiring knowledge in this important and newly emerging field. It will service as a reference in actual design practice and will be a trusted companion in the design adventures ahead."

-- "Forward to the first edition of the (Wolf) book", by Prof. Lynn Conway

"This book takes a more abstract approach to embedded systems. … Rather than connecting on particulars, the book tries to study more generic examples to come up with more generally applicable principles. However, I think that this approach is both fundamentally easier to teach and in the long rum more useful to students. It is easier because one can rely less on complex lab setups and spend more time on pencil-and-paper exercises, simulations, and programming exercises. It is more useful to the students because their eventual work in this area will almost certainly use different components and facilities than those used at your school. Once students learn fundamental, it is much easier for them to learn the details of new components."

-- "A note to teachers" in the preface of the (Wolf) book

"This book takes a developer's perspective to teaching embedded systems concepts. It examines, in detail, each of the important theoretical and practical aspects that one must consider when designing today's applications. These include the formal hardware and software development process (stressing safety and reliability); the digital and software architecture of the system; the physical world interface to external analog and digital signals; the debug and test throughout the development cycle; and finally improving the system's performance."

-- "Organizing the book" in the Preface of the (Peckol) book

- We only have one 160-minute lecture each week for 14 weeks (2240 minutes, or 2100 minutes (that is, 35 hours) if we exclude the 10-minute break in each lecture and the lecture reserved for the midterm exam), so we **will not** be able to cover all we want and need to learn as an embedded system designer. However, we **will** have a balanced curriculum that covers both breadth and depth of the knowledge on embedded system. In another word, you will see all aspects of embedded system design in some sense, and learn details of some techniques for embedded system design.
- This is not a DESIGN course/lab, so we **will not** build any real embedded systems on hardware platforms such as microprocessor, micro-controller, or FPGA. (This is also a result of the limited time we have for the course). However, we **will** touch many practical aspects of the design and evaluation of embedded systems.
- This is not a PROGRAMMING course, so we **will not** have intensive programming assignments (either in C, assembly, or any other languages). However, (embedded) software is a very important component in embedded system design, so we **will** review C and assembly programming. There will be a couple of homework assignments on programming as well as one C programming project. The time to complete these assignment will vary based on your programming background and experience. However, you do not have to be very familiar with C or assembly to succeed as long as you understand the basic concepts in programming.
- This is not a course on some very mature subjects (such as calculus and linear algebra), so it **will not** be sufficient to master only the materials in the textbooks and our lecture notes, we recommend you to read some up-to-date articles, reviews, and technical papers to stay with the latest technologies, market demands, and trends of embedded system design.
- In order to learn the course materials, you **will not** be allowed to simply sit in the lectures, listen and take notes (or naps), you **will** be required to actively participate in the class by asking questions, answering questions, and discussing interesting topics.

- **CoVID-19 related excused absence policy.** In addition to the university's excused absence policy (*https://policies.umd.edu/assets/section-v/V-100G.pdf*), it is required that (by the Nov. 22, 2021 email message from Associate Provost for Faculty Affairs*)*:
    - *Students must notify faculty members regarding any excused absence in a timely manner. Notification may be prior to class or as soon after the absence as possible. In the case of religious observances, athletic events, and planned absences known at the beginning of the semester, the student must inform the instructor during the schedule adjustment period.*
    - *Students must provide appropriate documentation of the absence. For medically necessary absences, students may provide self-documentation once per semester. While faculty members may ask students for documentation of medically related absences, we ask faculty for some latitude during the COVID-19 pandemic.*
    - *Faculty members must provide students with an accommodation for an excused absence. The accommodation may take the form of a make-up exam or another assessment that covers the same material and at the same level of difficulty as the exam or assignment that the rest of the class completed. The makeup assessment must be timely and it must take place at a time and place agreed on by both the instructor and the student. Having a course policy to drop the lowest grade, and using that dropped grade as the accommodation for an excused absence, is not a best practice for ensuring fairness and timely feedback.*

If you have to be absent from the class for a short-term (less than 2 weeks), check the university recommended guidance at https://provost.umd.edu/node/4360 *and contact Dr. Qu immediately for arrangements.*