

# CMPE 311:

## C Programming and Embedded Systems

### UMBC

### Fall 2016

#### I. COURSE DESCRIPTION

In this course, students learn about hardware and software aspects of embedded systems. Students learn C programming language through use in an embedded platform. The course builds on CMPE 310, introducing advanced topics including communication interfaces, advanced IO devices and other peripherals, multitasking, firmware, real-time operating systems/embedded operating systems and device drivers. The course will provide a hands-on experience in designing and programming an embedded system using a microcontroller based development platform.

#### II. COURSE DETAILS

##### A. Meeting Times and Locations:

- Lecture
  - M,W 5:30 - 6:45p
  - Engineering 022
- Discussion
  - Tu,W. 4 – 5:15p
  - Information Technology 375

##### B. Instructor

- Instructor: William H. Smith III (Bill)
- Department of Computer Science and Electrical Engineering
- Cell Phone: 484-554-3661 (texting works best!)
- Email: [whsmith003@hotmail.com](mailto:whsmith003@hotmail.com)  
[william.smith3@ngc.com](mailto:william.smith3@ngc.com)
- Office Hours: Because I am adjunct faculty, I won't be able to hold regular office hours. The plan is to be here in the classroom at 5p each evening and remain after the discussion class for as long as required to answer questions. You are welcome and encouraged to talk with me and ask questions during those times! I am also usually only a text away...

##### C. Course Textbook

- Required (?)
  - James K. Peckol, 'Embedded Systems: A Contemporary Design Tool' First Edition, John

Wiley & Sons, Inc. (2008), ISBN: 978-0471721802

- (strongly!) Recommended:
  - Brian W. Kernighan and Dennis M. Ritchie 'The C Programming Language' 2nd edition: ISBN: 013-11036208 (Paperback), ISBN: 013-1103709 (hardback)
  - Jonathan W. Valvano 'Embedded Micro-computer Systems: Real Time Interfacing' 3rd edition ISBN: 978-1111426255

##### D. Logistics

- UMBC Blackboard will host most of the course materials for this lecture.
- Piazza has been setup. The teaching assistants will be the primary responders with backup by the instructor as needed.
- Teaching Assistant: Seyed-Morteza Hosseini, (Morrtaza) email: [js10@umbc.edu](mailto:js10@umbc.edu) Office Hours: TBD
- Undergrad Teaching Assistant: 'Zachary Blaha' email: [blahaz1@umbc.edu](mailto:blahaz1@umbc.edu) Office Hours: TBD

#### III. PREREQUISITES

Required: CMPE 310 - Systems Design and Programming

#### IV. MAJOR LEARNING OBJECTIVES

At the conclusion of the course, students will:

- have developed programming skills in the C language;
- understand the hardware and software requirements for a microcontroller-based embedded system
- be familiar with, and will be able to use, built-in microcontroller peripherals, including coding and building external hardware
- be able to develop code for multi-tasking applications, and will be familiar with processes synchronization, resource sharing, and task scheduling
- be able to install and use a real-time operating system on embedded hardware

- be able to design and build a real-time system performing data capture, communications, and user interface

## V. COURSEWORK, GRADING, AND ASSESSMENT

### A. Grading Scheme

Evaluation of students will be made approximately according to the following scheme, though the instructor reserves the right to make adjustments as is fair and appropriate during the semester:

Grading of students includes the following:

- Midterm Exam: 15%
- Final Exam: 20%
- In-class Quizzes: 10% (2.5% each...)
- In-class Participation: 5%
- Lab Assignments (programming projects): 45% (composed of several assignments with varying amounts of credit based on length and difficulty, to be used to generate a weighted average.)
- Discussion Participation: 5%

### B. Late work policy:

A two day delay policy for Discussion Homework will result in declining percentage awarded. 5% will be lost for one day overdue, 25% will be lost for two days overdue. Unless previously cleared with the Professor, assignments submitted later than two days after their due date will receive a zero. Because of the requirements for subsequent Lab projects, delays of more than a week will VERY seldom be granted.

### C. Incomplete Grades

A grade of incomplete will be given only under exceptional circumstances described by the University policy for granting incompletes. Any such circumstance MUST be brought to the instructor's attention immediately as soon as it is known.

Failure to complete assignments on time is not a sufficient reason for an incomplete. If you feel you are falling behind, seek help immediately.

### D. Academic Misconduct

The assignments in this class are for individual work. The instructor expects that you may minimally discuss some approaches to the projects, but you may not collaborate on writing code or share or copy it with others. You must never copy code or turn in anything that not representative of your learning and mastery of the material. Cheating or academic misconduct related to a assignment will make you subject to the maximum allowed penalty from the university. A zero on the assignment is only the minimum penalty. If you are stuck late and desperately decide you need to copy something to

move on to complete the rest of an assignment, you must cite your source to only receive a grade reduction as appropriate and avoid being subject to academic misconduct penalties. See

<http://www.umbc.edu/undergrad/ed/ai/students.html> for policies and definitions.

All students are expected to be knowledgeable regarding all University policies on academic misconduct. By enrolling in this course, each student assumes the responsibilities of an active participant in UMBC's scholarly community in which everyone's academic work and behavior are held to the highest standards of honesty. Cheating, fabrication, plagiarism, and helping others to commit these acts are all forms of academic dishonesty, and they are wrong. Academic misconduct could result in disciplinary action that may include, but is not limited to, suspension or dismissal. To read the full Student Academic Conduct Policy, consult the UMBC Student Handbook, the Faculty Handbook, or for graduate courses, the Graduate School website.

## VI. TENTATIVE SCHEDULE OF TOPICS

- 1) Introduction
- 2) CH1\_and\_Review
- 3) Microcontrollers
- 4) AVR 8-bit Architecture
- 5) AVR CPU Registers
- 6) AVR IO Ports
- 7) AVR Addressing Modes
- 8) More AVR Assembler
- 9) C Basics
- 10) Functions, Separate Compilation, Macros
- 11) AVRI IO Examples - I think - Discussion IV
- 12) Arrays Argument Passing Promotion Demotion
- 13) C Strings
- 14) Memory Usage
- 15) C Pointer Variables
- 16) Debugging and logging with printf macros
- 17) Pointers And Arrays
- 18) Struct and Union
- 19) Advanced Pointers
- 20) Final C
- 21) Memory-Related Perils and Pitfalls.ppt
- 22) Interrupts
- 23) Timers, Counters
- 24) integers
- 25) RTOS
- 26) Tasks
- 27) Converters
- 28) Communications