University of Nevada, Reno

COMPUTER SCIENCE & ENGINEERING DEPARTMENT
CPE 301 - EMBEDDED SYSTEMS DESIGN
Fall 2014

Catalog:
Description: Introduction to embedded systems design and applications. Field programmable gate arrays, microcontroller architecture, memory and I/O decoding, timers, interrupt systems, analog to digital converters. Prerequisites: CPE 201

Textbooks: David Russell, *Introduction to Embedded Systems: Using ANSI C and the Arduino Development Environment*, Morgan & Claypool, 2010.

References: I. R. E. Haskell & D. M. Hanna, *Learning By Example Using Verilog: Basic Digital Design With a BASYS FPGA Board*, LBE Books, 2008.
M. Morris Mano, *Digital Design*, 4th Ed., Prentice Hall, 2007.
Jan Axelson, *The Microcontroller Idea Book*, Lakeview Research, 1997. (see: http//www.lvr.com)

Instructor: Dwight Egbert, Professor of Computer Science & Engineering (egbert@cse.unr.edu)

Office Hours: 11:30AM-1:00PM Monday & Wednesday (or by appointment); Room 322 SEM; Tel: (775) 784-6952

Academic Success Services:
Your student fees cover usage of the Math Center (784-4433 or www.unr.edu/mathcenter/), Tutoring Center (784-6801 or www.unr.edu/tutoring/), and University Writing Center (784-6030 or www.unr.edu/writing_center/. These centers support your classroom learning; it is your responsibility to take advantage of their services. Keep in mind that seeking help outside of class is the sign of a responsible and successful student.

If you have a disability for which you will need to request accommodations, please contact me or Mary Zabel at the Disability Resource Center (Thompson Student Services – room 100, 784-6000, or www.unr.edu/drc/), as soon as possible to arrange for appropriate accommodations.

Surreptitious or covert video-taping of class or unauthorized audio recording of class is prohibited by law and by Board of Regents policy. This class may be videotaped or audio recorded only with the written permission of the instructor. In order to accommodate students with disabilities, some students may have been given permission to record class lectures and discussions. Therefore, students should understand that their comments during class may be recorded.

Course Goals:
To become familiar with the architectures of example microcontrollers. To understand the fundamental principles of digital hardware components in system design using microcontrollers and interfacing them to peripheral devices. To understand and be able to build and control embedded systems using both microcontrollers and field programmable gate arrays.

Course Topics	**Approximate Calendar:**	
• Microprocessor and Microcontroller Architectures	Week 1	
• C Language Pointers and Bitwise Operators	Week 1	
• Embedded Systems Design	Week 2	
• I/O Interfacing: via Data/Address busses	Weeks 3 and 4	
• I/O Interfacing: via built-in I/O Ports	Week 5	
• Arduino Atmega Architecture and Programming in C	Weeks 5 and 6	
• Using Special Function Registers	Week 7	
• Systems Programming Considerations	Week 8	Midterm Exam - Date TBA
• System Timers	Week 9	
• Serial Port communications	Week 10	
• System Interrupts	Week 11	
• Analog to Digital Conversion Techniques	Week 12	
• FPGA applications in system design	Weeks 12 and 13	
• Verilog and VHDL design languages	Week 14	
• Review	Week 15	Final Exam

Course Objective:
Students will demonstrate an understanding of and competent use of Microcontroller and FPGA Architectures, interfacing principles, and systems programming.

Student Participation:
The course will contain three basic and interrelated blocks. First, the textbook will provide the framework for the course. Second, as material is reached in the textbook it will be related to supplementary material covering advanced microprocessor topics. Third, each student will design and build hardware to interface to the 8051 single board computer in Lab.

Students are expected to attend all classes and read all of the assigned sections of the textbook. Often, material will not be covered in both lectures and reading assignments. Thus, both are essential to a full understanding of the course content. During most classes a short example problem related to the current topic will be assigned. Students will spend a few minutes working alone on this problem followed by a few minutes discussing their solutions with two or three other students. These solutions will be collected and used as a basis for up to 5% extra credit for the course grade.

Also, completion of homework is essential. Homework will be due each TUESDAY, or the next following class if there is no Tuesday class.

LATE HOMEWORK WILL BE ACCEPTED FOR AT MOST 50% CREDIT.

Students are encouraged to study together, but each person must prepare his or her solutions and have a firm understanding of any work turned in. When you put your name on your homework you are stating that it is your own work and not the work of another person. As a reminder of UNR academic standards, please read from the ***UNR on-line Catalog, University Code of Conduct and Policies, POLICIES AND GUIDELINES, ACADEMIC STANDARDS*** http://www.cis.unr.edu/ecatalog/Default.aspx?article_list_id=29351 defining these standards. Specifically, the following: "*Plagiarism is defined as submitting the language, ideas, thoughts or work of another as one's own; or assisting in the act of plagiarism by allowing one's work to be used in this fashion*." This means that if another student asks to borrow your work to copy - JUST SAY NO - or you are participating in plagiarism.

Course Grade Structure:
Each course activity will contribute to the course grade as shown below. All activities will be graded on a scale of 0-100 points, and the final course grade will be determined as shown below.

STUDENTS MUST PASS BOTH LECTURE AND LAB IN ORDER TO PASS THE COURSE

Passing the lecture means that the average score for all exams is passing (>=50)

Passing the Lab means that the average for all labs is passing (>=50)

All quizzes and exams given in this course will be closed notes and closed books. Only calculators and materials handed out at the time of the exam may be used. Normally, plus/minus grades are not given in this class. The instructor reserves the right to assign plus/minus grades under special circumstances involving borderline grades based upon class participation. Your grade will never be lower than defined here unless you have an excessive number of un-excused absences from class or lab, however, positive class participation can be used as a basis for raising your grade.

HOMEWORK	20%
MIDTERM EXAM	30%
COMPREHENSIVE FINAL EXAM	30%
LAB GRADE	20%
= COURSE GRADE	100%

90 - 100 points = A | 80 - 89.9 points = B | 65 - 79.9 points = C | 50 - 64.9 points = D | 00 - 49.9 points = F

Course Outcomes:

The course outcomes are skills and abilities students should have acquired by the end of the course. These outcomes are defined in terms of the Computer Science and Engineering ABET Accreditation Program outcomes which are relevant to this course. All outcomes are listed below and those relevant to this course are identified in the following Table.

1. an ability to apply knowledge of computing, mathematics, science, and engineering.
2. an ability to design and conduct experiments, as well as to analyze and interpret data.
3. an ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs, within realistic constraints specific to the field.
4. an ability to function effectively on multi-disciplinary teams.
5. an ability to analyze a problem, and identify, formulate and use the appropriate computing and engineering requirements for obtaining its solution.
6. an understanding of professional, ethical, legal, security and social issues and responsibilities.
7. an ability to communicate effectively with a range of audiences.
8. the broad education necessary to analyze the local and global impact of computing and engineering solutions on individuals, organizations, and society.
9. a recognition of the need for, and an ability to engage in continuing professional development and life-long learning.
10. a knowledge of contemporary issues.
11. an ability to use current techniques, skills, and tools necessary for computing and engineering practice.
12. an ability to apply mathematical foundations, algorithmic principles, and computer science and engineering theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices.
13. an ability to apply design and development principles in the construction of software systems or computer systems of varying complexity.

Program Outcomes	Course Outcomes	Course Strategies & Actions
7	Students demonstrate that they can communicate how their engineering solutions work.	Students must present an oral report for each lab assignment and answer questions from the teaching assistant about the assignment.
8	Students demonstrate that they understand how microprocessors work and how they are integrated into consumer products.	Study of CPU registers and memory & I/O mapping. Homework & labs requiring students to design, build, and test interfaces.
9	Students demonstrate that they can learn systems programming for more than one microprocessor.	Homework requiring specific program functions. Labs requiring programs to activate hardware interfaces. Examination of systems programming for both Intel 8051 and ARM 7 RISC processors.
10	Students demonstrate that they can interface external digital devices to different microprocessors based on current popularity of processors.	Study of single board computer schematic diagrams. Homework & labs requiring students to design, build, and test interfaces.
11	Students demonstrate that they can design and write systems programs for a microprocessor to make it perform pre-defined tasks.	Homework requiring specific program functions. Labs requiring programs to activate hardware interfaces.
12	Students demonstrate that they can design interfaces for a microprocessor and a single board computer using different devices.	Study of CPU registers and memory & I/O mapping necessary for different interface devices. Homework & labs requiring students to design, build, and test interfaces