Tentative Syllabus

EE 319K: Introduction to Embedded Systems
Sections - 16160, 16165, 16170

Mattan Erez

Spring 2010

General Information

Class Time	TTh 3:30-5:00 pm
Classroom	**ENS 127**
Office	ENS 538
Contact	mattan.erez@mail.utexas.edu
Pre-reqs	EE 306 and EE 312 with a grade of at least C.
Office Hrs	W 4:30 – 5:30 pm and Th 2:00 – 3:00pm (or by appointment)
Website	UT Blackboard and Dr. Valvano's class site: http://users.ece.utexas.edu/~valvano/EE319KF09.html
TAs	Evgeni Krimer Eshwaran Kumar Fábio Fernandes Mehmet Basoglu Karthik Sankar

Catalog Description

Embedded systems; machine language execution; assembly language programming; local variables; input/output synchronization; analog to digital conversion, digital to analog conversion; debugging; and interrupts.

Text and Reference Materials

Introduction to Embedded Microcomputer Systems: Interfacing to the Freescale 9S12, Cengage Publishing 2009, ISBN-10: 049541137X | ISBN-13: 9780495411376, by J. W. Valvano

Data sheets for most of the devices used in this class are available as pdf files on my Dr. Valvano's site. Please make sure you have access to the CPU12 instruction manual, either in print (from 2nd floor lab or in my office) or pdf form. You should download these data sheets and have them available when you are developing code
32 page CPU12 quick reference:
http://users.ece.utexas.edu/~valvano/Datasheets/CPU12rg.pdf
458 page CPU12 programming reference:
http://users.ece.utexas.edu/~valvano/Datasheets/S12CPUV2.pdf
9S12DP512 datasheets:
http://users.ece.utexas.edu/~valvano/Datasheets/MC9S12DP512.zip
Tech arts board information:
http://users.ece.utexas.edu/~valvano/Datasheets/TechArts9S12DP512.pdf
Short Movies on TExaS: http://users.ece.utexas.edu/~valvano/Readme.htm

Web Video Lessons: http://users.ece.utexas.edu/~valvano/Lessons/

Equipment

Buy: You will need a voltmeter (one less than $20 will do), soldering iron (with solder), and a wire stripper. We will be giving you a 9S12DP512 development board, a prototyping board, and some external components.

Kit handed out by Daryl Goodnight (Room ENS 234)

9S12DP512 board

Power adapter

RS232 9-pin cable

Protoboard

Kit handed out by TAs

1 7406 Jameco 200 49091

6 RESISTOR, 1/4W 5%, 220 OHM, 690700

3 RESISTOR, 1/4W 5%, 10K OHM, 691104

3 Resistor, 1/4W 5%, 12K ohm, 691121

3 Resistor, 1/4W 5%, 1.5K ohm, 690902

1 0.1 uF capacitor ceramic Z5U Jameco 200 25523

3 B3F-1052 SWITCH TACT, Digikey SW405-ND

2 Red LEDs, T1 3/4, 20mA Digikey 160-1087-ND

2 Yellow LEDs, T1 3/4, 20mA Digikey 160-1088-ND

2 Green LEDs, T1 3/4, 20mA Digikey 160-1089-ND

1 30mm slide pot, Alpha RA300BF-10-20D1-B54 Mouser# 312-9100F-50K

18in 6-wire 24 guage solid wire All Electronics 6C24

1 Stereo Jack All Electronics MJW-11 $0.10

Long term checkout

All Electronics LCD-100 - 16 character by 1 line LC;

All Electronics Two- CON-244 4-pin connector

Software

TExaS: The simulator application, called Test EXecute And Simulate, is not freeware, so please don't post it on the net or otherwise send it to others. On the other hand, Jon Valvano grants EE319K students indefinite usage of the software, including installing the application on each of your personal computes. If you know of someone interested in the application have them contact him directly. It is the eighth semester we will be using the application in EE319K, but I will be make updates to fix bugs or add features. Please work through the tutorials and examples to bring you up to speed on the various aspects of the system. Instructions for installing/upgrading TexaS will be provided on Blackboard. The TExaS application itself runs on Vista, but the help system does not run on Vista without a patch. Ask your TA how to get the TExaS help system to run under Vista.

Course Objectives

1. Understanding how the computer stores and manipulates data (characters, integers, and fixed-point numbers), the basic arithmetic and logical operations performed by the computer,

2. The understanding of embedded systems (a system with the computer hidden inside) using modular design and abstraction,

3. Assembly language programming: considering both function and style,

4. Understanding how the computer executes instructions (fetch opcode, fetch operand, read data, operate, and write data)

5. The use of a microcontroller (strategic use of RAM ROM and I/O) Microcontrollers typically have a little RAM and a lot of ROM. Globals, locals and the heap go in RAM. Constants and programs go in ROM.

6. Debugging and verification using a simulator and on the microcontroller (embedded systems typically do not have a print function) debugging using breakpoints, scanpoints, profiles, monitors, voltmeters, oscilloscopes, logic analyzers

7. How input/output actually happens (the students wire up analog and digital signals to the 9S12 and measure them with a voltmeter), synchronization, including switches, LEDs, LCDs, DACs, ADCs, and serial ports,

8. The implementation of an I/O driver (a set of programs that perform I/O)

9. Understanding, from an architecture standpoint, how local variables and parameters work (e.g., a space on the stack is dynamically created, the local variable is accessed using stack-pointer relative addressing, then the space is deallocated.)

10. Analog to digital conversion (ADC) e.g., the students interface a slide potentiometer to the ADC, and write software that measures the position of the slide, creating a display like “1.23 cm”

11. Interrupt synchronization, real-time ADC sampling (periodic timer interrupts), introduction to multithreaded programming

12. Simple motors (e.g., open and closed-loop stepper motor control)

13. Digital to analog conversion (DAC), used to make simple sounds

14. Design and implementation of elementary data structures, such as linked lists, stacks and queues.

After the successful conclusion of EE319K students should be able to understand the basic components of a computer, write assembly language programs that perform I/O functions and implement simple data structures, manipulate numbers in multiple formats, and understand how software uses global memory to store permanent information and the stack to store temporary information.

Honor Code

"*Faculty in the ECE Department are committed to detecting and responding to all instances of scholastic dishonesty and will pursue cases of scholastic dishonesty in accordance with university policy. Scholastic dishonesty, in all its forms, is a blight on our entire academic community. All parties in our community -- faculty, staff, and students -- are responsible for*

creating an environment that educates outstanding engineers, and this goal entails excellence in technical skills, self-giving citizenry, an ethical integrity. Industry wants engineers who are competent and fully trustworthy, and both qualities must be developed day by day throughout an entire lifetime. Scholastic dishonesty includes, but is not limited to, cheating, plagiarism, collusion, falsifying academic records, or any act designed to give an unfair academic advantage to the student. The fact that you are in this class as an engineering student is testament to your abilities. Penalties for scholastic dishonesty are severe and can include, but are not limited to, a written reprimand, a zero on the assignment/exam, re-taking the exam in question, an F in the course, or expulsion from the University. Don't jeopardize your career by an act of scholastic dishonesty. Details about academic integrity and what constitutes scholastic dishonesty can be found at the website for the UT Dean of Students Office and the General Information Catalog, Section 11-802."
You are allowed to talk to your classmates about the lab assignments, but you are NOT allowed to look at each other's written work. Oral discussion about an assignment is encouraged and is not considered to be cheating. Copying of any part of a program is cheating without explicit reference to its source. If we find two programs that are copied, there will be a substantial penalty to both students, e.g., failure in the course. Students who cheat on tests or in lab will fail. Prosecution of cases is very traumatic to both the student and instructor. It is appropriate to use software out of the book, class website as long as all copy-pasted software is explicitly referenced. Copy-pasting software from current or past EE319K students is scholastic dishonesty. Policies concerning the use of other people's software in this class:

- You may use any existing source code that is clearly referenced and categorized:
  original: completely written by you,
  derived: fundamental approach is copied but it is your implementation,
  modified: source code significantly edited to serve your purpose,
  copied: source code includes minor modifications.

Grading Criteria

Task	Date	Percentage
Homework Assignments	Usually due on Fridays	10%
1st Test(In Class)	Thursday February 25	15%
2nd Test (During class time location TBD) - Programming	Tuesday March 30	20%
Final Exam	Tuesday May 18th 9:00-12:00 noon	25%
Programming Lab Assignments	Due at respective lab times	30%

When programming labs are performed as a team (of two) only one solution must be turned in. All exams are closed-book.

Homework Assignments: There will be 10 homework assignments of which one lowest scoring assignment will be dropped. This should give you a chance to skip one assignment if need be. They will be administered over the web (active after 1/22). You may work in groups of any size but each of you MUST submit your individual solution online:

Step 1: Log into Quest Homework Service at: https://quest.cns.utexas.edu/student
Click on the "Get Started" link and login with your EID, Unique #: 16160

Step 2: Download Student's Instructions and First Homework after Friday 1/22. Subsequent assignments will be posted on successive Wednesdays unless otherwise announced in class.

Step 3: Work on the homework offline

Step 4: Log in any time (possibly multiple times) to submit (resubmit) answers, before due time (usually Friday 11:55 pm); First homework will be due Friday 1/29

Step 5: Download solutions after due time

Partial Lecture/Reading Schedule

Date	Reading Assignment	Topic
Week 1 (1/19,1/21)	Ch 1	Course Description, Flow Charts, Data Flow Graphs, Call Graphs 9S12 Programming, Memory Map, Registers
Week 2	A 1.1-1.8.2 Ch 2,3	TExaS Simulator, Design of a NOT gate Numbers, Hexadecimal, TExaS simulator
	Ch 2,4	9S12 Architecture, Memory Allocation, Simple Addressing Modes, Immediate, Extended Addressing
Week 3	Ch 2, 3	Parallel ports, Direction Registers, Logical and Shift operations
	Ch 4.6, 5.8	Debugging in TExaS
Week 4	Ch 3,5 A 2	Arithmetic operations, Condition Code bits. If-then; Demonstration of the board (bring your board to class): Switch Input and LED Output
	Ch 5.3,6.11	Subroutines and The Stack, Debugging dump, Activation Records, Pointers and Index Registers
Week 5	Ch. 6.1-6.3	Pointers
	Ch. 4.4,5.8	Timers, functional debugging
Week 6	Ch 1 to 5	Review for Test 1
	Ch 1 to 5	**Test 1 (closed book) in class, during regularly scheduled class time**

Date	Reading Assignment	Topic
Week 7	Ch. 6.8,6.11	Finite state machines
	Ch. 7.1-7.4,9.1	Local variables
Week 8	Ch. 10.1, 10.5, 8.4	LCD interface, number conversions
	Ch. 9.1, 9.2, 9.4, 9.6	Timers, interrupts, debugging interrupts
Week 9	10.2, 11.1, 11.4	C/assembly interface, device drivers, analog to digital conversion
Week 10	Ch 5 and 6	**Test 2 (closed book) during regularly scheduled class time; Location TBD**
		C/assembly and Metrowerks
Weeks 11, 12	Ch. 9.6, 10.1, 11.4, 8.1, 12.1 – 12.4, 11.2, 11.3	Design methods, SCI interface, FIFO queues, real time interfaces, digital to analog conversion and sound generation
Weeks 13, 14	Ch. 8.4	Stepper motors, TBD
Week 15	Ch 1 to 13	TRobots Competition Finals and Review

Lab Schedule

Date	Task
1/20	Go to ACA1.106 for Lab grading policy and demonstration
1/27	Go to ACA1.106 for TexaS demonstration
2/3	**Lab1**: Digital Lock I/O, parallel port, direction register and logical function(simulated, individual)
2/10	Real board demonstration, bring your board to lab
2/17	**Lab2**: LED flashing in Assembly (simulated and board, individual)

Date	Task
2/24	Go to ACA1.106 for Exam 1 study help
3/3	**Lab3**: Debugging, one switch, one LED (simulated and board, individual)
3/10	**Lab4**: Traffic Controller using the real 9S12 (simulated and board, individual)
3/24	**Lab5**: LCD device driver, decimal fixed-point output, local variables(simulated and board, groups of two)
3/30	**Programming Test 2 (closed book), during regularly scheduled class time; Location TBD**
3/31	Go to ACA1.106 for Metrowerks demonstration
4/7	**Lab6**: Real-time Position Monitor, ADC, Interrupts, LCD (simulated and board, groups of two) – Mix of Assembly and C
4/14	**Lab7**: Distributed DAS, Serial Port Interrupts, FIFO queue (simulated and board, groups of two) – Mix of Assembly and C
4/21	**Lab8**: Real-Time sound generation, 4-bit DAC (simulated and board, groups of two) – Mix of Assembly and C
4/28	Goto ACA1.106 for Debugging help on Lab 9
5/4	**Lab9**: Introduction to TRobot, data structures, layered software, stepper motors (simulated, groups of two) – NO LATE TURN IN FOR LAB 9 – **Due on Blackboard by 10pm (Note: This is a Monday)**
5/6	**TRobot Final competition** (in class)(simulated, groups of two)

Laboratory policies

This is a programming class. Therefore, the quality of the software you write will significantly affect your grade. When you get the program finished, make a printout of your assembly listing, staple a grading sheet to the front, then demonstrate it to a TA. The TA will record the performance and demonstration grades on the assembly listing printout. Labs are due during your scheduled lab period on Thursday. A detailed schedule follows. Late assignments incur a penalty of 10% of the maximum score per day for the first three days and will not be accepted afterwards. Exceptionally "good" programs may be given extra credit. If you do extra functions in your program, point out and demonstrate the extra functions to the TA when you demonstrate your program. Partial credit, not to exceed a maximum of 75% of full credit, may be awarded to a program that does not meet all specifications. Often it is better to demonstrate an incomplete program, rather than incur late penalties while fixing all its bugs. There are four components to the lab grade:

1. Deliverables 20% (graded later)

2. Performance 30% (graded by the TA at the time of checkout):
   How well does it work? Does it crash?
   Does it handle correctly all situations as specified?
   How clean is the user interface?
   Possible 5% Bonus, does it do more than specified?

3. Demonstration 30% (graded by the TA at the time of checkout):
   Can you explain to the TA how your software works?
   Can you explain why you made certain software engineering tradeoffs?
   Both partners if applicable must be present during the demonstration.

4. Software quality 20% (graded later):
   Clearly describe the data and program structures in the software comments. Simple well-structured software with descriptive labels is much easier to understand than complex software with voluminous comments. Use either top-down or bottom up structure. Modular programming divides the problem into "natural divisions" with minimal coupling. The interfaces should be simple and natural. The software will be graded on the correctness of the program function, the comments, the interface to the user, the style, and organization. The effort spent to write high quality software will be rewarded both during debugging and when it is time to add features in the future.

5. Late penalties (-10% per day)

6. There will be a ***one grade level reduction*** for not returning the LCD device or the 4-pin serial connector.

Lab Partners: The first 3 labs must be performed individually. Lab 1 will be collected on a first come first served basis. For Labs 2 and 3 you will be assigned a 5-minute checkout time during the regularly scheduled lab period. If you are late or miss your slot your grade will likely be lower because you will not have time to present your lab to the TA. You will work in a group of two for the remaining labs. The TA will randomly select partners for Lab 4 to 8, and the partnerships will be rotated week to week. The TA will assign each partnership a 10-minute checkout time during the regularly scheduled lab period. In cases where there are an odd number of students in a section, we will attempt to create a partnership with another section and assign a 10-minute checkout time convenient for both students and the TA. Note that we cannot guarantee that you will have a partner for each of the labs. Both partners must be present during the TA demonstration. If you and your partner did not get along, you are still both responsible for the lab. The TA may assign a different grade to each partner based on your performance during your checkout demonstration. You will be allowed to choose your own partner for Lab 9.

Class Drop Deadline

The deadline for dropping without possible academic penalty is Monday March 29, 2010.