EE 319K: Introduction to Embedded Systems Sections – 16600, 16603,16605,16610,16613 and 16615

Ramesh Yerraballi

Fall 2011

General Information

Class Time	TTh 12:30-2:00 pm
Classroom	**ACA 1.104** **Lab: ACA1.106**
Office	ENS 106
Contact	ramesh@mail.utexas.edu
Pre-requisites	EE 306.
Office Hrs	Mon/Wed 1:30 - 2:30pm (or by appointment)
Website	UT Blackboard and Dr. Valvano's class site: http://users.ece.utexas.edu/~valvano/EE319KF09.html
TAs	Peter Garatoni, peter.garatoni@gmail.com Manan Kathuria, manan@mail.utexas.edu Yingbo Aron Yu, aronyu07@gmail.com Kurt Fellows, kfellows85@gmail.com James Beecham, james.d.beecham@gmail.com TAs will hold office hours in Lab: ACA 1.106. The office hours will be posted in lab. About 22 students per TA

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.

Overview

EE319K will continue the bottom-up educational approach, started in EE302 and EE306. The overall educational objective is to allow students to discover how the computer interacts with its environment. It will provide hands-on experiences of how an embedded system could be used to solve EE problems. The focus will be understanding and analysis rather than design. The analog to digital converter (ADC) and digital to analog converter (DAC) are the chosen mechanism to bridge the CE and EE worlds. EE concepts include Ohms Law, LED voltage/current, resistance measurement, and stepper motor control. CE concepts include I/O device drivers, debugging, stacks, FIFO queues, local variables and interrupts. The hardware construction is performed on a breadboard and debugged using a multimeter (students learn to measure voltage and resistance). Software is developed in Freescale 9S12 assembly and C; 3 out of the 9 labs are simulated-only in TExaS and other 6 are first simulated on TExaS then run on the real 9S12. Software debugging occurs during the simulation stage. Verification occurs in both stages. Labs 1 through 6 are written in 9S12 assembly language. Lab 7 is written in a C. Labs 8 and 9 are written in a combination of assembly and C.

Outcomes

After the successful conclusion of EE319K students should be able to understand the basic components of a computer, write assembly and C 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.

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*

If you find any mistakes in the book, please let Dr. Valvano know. Known errors can be found at list of textbook errors on Jon's website. If you are the first student to find the mistake you can receive one point bonus on Exam 1. Mistakes will be accepted up to the last class day of the semester, and there will be a maximum of 5 points awarded to each student. We have written a C programming book (C Primer) specific for the Metrowerks compiler and 9S12.

EE319K will have reading assignments from Patt's Introduction to Computing Systems (textbook for EE306 and BME303). Data sheets for most of the devices used in this class are available as pdf files on 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.pdf
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/
C Primer: http://users.ece.utexas.edu/~ryerraballi/CPrimer/

Equipment

Must Buy: We are no longer giving microcontroller boards to students. I can tell you the discussions on how to maintain educational excellence in the face of repeated budget cuts has been long and careful. Every group of two students will be required to have one 9S12 board. Either the Technological Arts Adapt9S12DP512 or the Adapt9S12DG128 will suffice. The **first** possibility is to buy or borrow a board from a previous EE319K/EE445L student. If you obtain a kit from a previous student, you will need the board, power supply, RS232 cable, and a protoboard. If you do buy a board from another student, I recommend you run the hardware checker to make sure the board is functional. The second possibility is to purchase a new Technological Arts Adapt9S12DG128 kit, which includes a board, power supply, serial cable, and solderless breadboard. The student cost will be $69.00 per kit. The

link to pay for this kit is active. The buying process will be:

1. Once you have selected your partner for Labs 3-9, you go to the http://www.technologicalarts.ca/uta/ web site and pay $69 to TechArts for the kit via PayPal. Please pay for your kit at least 7 days before the demonstration lab period. Our experience is that many credit card companies will add $1 to $3 to the transaction because TechArts is a Canadian company. Neither UT or TechArts sees this added fee. The fee is charged by the credit card company.
2. The department will get an email from TechArts that you have paid.
3. You will go to ENS220 (checkout window) to pickup the kit. There will be specific scheduled hours for the pickup phase, which will be around September 12-16.
4. There will be a board demonstration lab between labs 2 and 3, and you should bring your board to this demonstration. Your TA will show you how to test your board to verify it is operational. If the board does not work during the testing procedure, we will replace it. Our experience with EE319K is if the board works during the initial test, almost all students complete EE319K labs without damaging the board. However, if you do damage your board, then you must purchase another one. There is no difference to a EE319K student between the Adapt9S12DP512 and Adapt9S12DG128 board. The DG128 board has a little less memory and is a lot less expensive. You will need a voltmeter (one less than $20 will do), soldering iron (with solder), and a wire stripper. Since you will be making only 5 solder joints all semester, it is acceptable to borrow a soldering iron. However, all EE319K students will need their own voltmeter and wire strippers. The NI box you bought in EE302 can be used for as the voltmeter.

Safety warnings: Due to the lead in most solder, please wash your hands after soldering, before eating or drinking. If you drop the soldering iron, let it fall to the ground. Do not try and catch it. If you are pregnant or think you might be pregnant, have someone else do the soldering.

Should you buy a computer? Let it be perfectly clear that the following remarks are my personal opinion, and do not reflect an official position of the department, college, or university. I feel very strongly that you should have your own computer on which you develop your software and write your reports. I think both software development and report writing should be done without paper, pencil and erasers. Having a computer at home allows you to organize your information (files, directories etc.) as well as your schedule (allocate your software development time for that the time of day during which you are most creative and energetic.) Physiologically most people are more energetic in the morning. On the other hand, there are fewer distractions late at night. Some students do all their EE319K lab work on campus, but most find it convenient to configure their home computers to work with the EE319K hardware.

Which computer should I buy? To work at home you will need to install TExaS and Metrowerks Codewarrior for the 9S12. For drawing electrical circuit diagrams some students use PCBArtist. All three of these applications run on a PC running Windows XP, Vista or Windows 7. The software development systems will run on any PC with a COM port.

EE319K kit handed out by TAs and do not need to be returned

1 7406
6 LEDs (20 mA, 1 red, 1 yellow, and 1 green LED)
6 220 ohm 5%, 0.25 watt resistors
3 push-button switches
3 10k ohm 5%, 0.25 watt resistors
3 1.5k ohm 5%, 0.25 watt resistors
3 12k ohm 5%, 0.25 watt resistors
24 inches 6-Wire 24 gauge used for prototyping
1 0.1 uF ceramic bypass cap
1 headphone jack
1 20k ohm slide pot

Item to get from long term checkout

1 LCD display and serial cable (if you check out either of these two and do not return them by 11/22, we will reduce your overall grade in EE319K one letter grade)

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 11th semester we will be using the application in EE319K, but Jon 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.

Metrowerks Codewarrior for the 9S12: The instructions to download the free (special) version can be found at
http://users.ece.utexas.edu/~valvano/S12C32.htm#Metrowerks
This version needs to be run as a 32-bit application. Freescale is planning a 64-bit release soon. So if you have Windows 7, you should wait for the 64-bit release.

PCBArtist: You will be drawing a 5 or 6 circuit diagrams as part of your lab assignments. You are free to draw these diagrams in any manner you wish as long as the information is presented in a complete and professional manner. One option for drawing circuits is PCBArtist, which can be downloaded at http://www.4pcb.com/ or http://www.4pcb.com/free-pcb-layout-software/. The use of PCBArtist is optional for EE319K students.

Teaching philosophy

I strongly encourage students to take an active role in this class. Questions are welcome before, during and after class. Please feel free to email, visit or call me if you have questions.

Detailed 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.

Legal Stuff

The 12th class day is September 9th. The drop policy has recently changed. See your academic advisor or the Dean of Students for more information. Course evaluation is conducted on the last class day in accordance with the Measurement and Evaluation Center form. The final exam is at the time and place stated in the course schedule. The University of Texas at Austin provides upon request appropriate academic adjustments for qualified students with disabilities. students with disabilities may request appropriate academic accommodations from the Division of Diversity and Community Engagement, For more information, contact Services for Students with Disabilities, 471-6259, http://www.utexas.edu/diversity/ddce/ssd/.

Religious Holy Days By UT Austin policy, you must notify me of your pending absence at least fourteen days prior to the date of observance of a religious holy day. If you must miss a class, an examination, a work assignment, or a project in order to observe a religious holy day, I will give you an opportunity to complete the missed work within a reasonable time after the absence.

Scholastic dishonesty: "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 encouraged to study together and to discuss information and concepts with other students. You can give "consulting" help to or receive "consulting" help from such students in oral form. However, this permissible cooperation should never involve one student having possession of a copy of all or part of work done by someone else, in the form of an email, an email attachment file, a portable storage device, or a hard copy. Copying of any part of a program is cheating without explicit reference to its source. We do enter lab assignments turned in by EE319K students through a plagiarism checker, comparing them to assignments of this and previous semesters. 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:

- I strongly encourage you to study existing software.
- All applications and libraries must be legally obtained. E.g.,

    - You may use libraries that came when you bought a compiler.
    - You may use software obtained from the web.
    - You may copy and paste from the existing source code.
- 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.

The University Honor Code is "The core values of the University of Texas at Austin are learning, discovery, freedom, leadership, individual opportunity, and responsibility. Each member of the University is expected to uphold these values through integrity, honesty, trust, fairness, and respect toward peers and community."
http://registrar.utexas.edu/catalogs/gi09-10/ch01/

Grading Criteria

Task	Date	Percentage
Homework Assignments (10)	Homeworks are usually due on Tuesdays;	10%
1st Test (In Class)	Thursday, September 29, 2011	15%
2nd Test (During class time, ACA 1.102 & 1.106) - Programming	Thursday, October 20, 2011	20%
Final Exam	Friday, December 9 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. Cutoff scores for the corresponding letter grades will not be determined until after the final exam.

Homework Assignments: There will be 10 homework assignments all involving programming in Assembly or C.

Attendance

Students are expected to attend lectures. The book covers more information than the class, and we will use lectures to map our way through the book. If you miss class you may find it difficult to catch up.

Partial Lecture/Reading Schedule

Date	Topic
Week 1	Lecture 1: Introduction (Chapter 1)- Course administration; Embedded systems, development cycle; Flow charts, data flow and call graph
	Lecture 2: TExaS (Appendix 1.1-1.8.2, Chapter 2 and 3) - TExaS simulator; Numbers, Hexadecimal

Date	Topic
Week 2	Lecture 2: Architecture (Chapter 2 and 4) – 9S12 architecture and execution; Simple addressing modes
	Lecture 3: I/O (Chapter 2 and 3)Execution, subroutines - Parallel ports, direction registers; Logical and shift operations
Week 3	Lecture 4: Debugging (Section 4.7 and 5.7) - Debugging in TExaS
	Lecture 4: Arithmetic (Chapter 3 and 5) - Arithmetic operations; Condition code bits
Week 4	Lecture 5: Board (Appendix 2, Section 2.6) - Demo of the board (bring board to class);Switch input and LED output
	Lecture 5: Modular programming (Section 5.1, 5.2, 6.11) - If-then, loops; Subroutines and the stack; Debugging dump
Week 5	**Lecture 6: Pointers (Section 6.1-6.3, 6.11) -** Indexed addressing; Arrays; Strings
	Lecture 7: Timers (Section 4.5, 5.7) – Timers; Advanced functional debugging
Week 6	Review for Test 1: Chapter 1 – 5
	September 29 - Test 1 (closed book): Chapter 1 – 5
Week 7	Lecture 8: FSMs (Section 6.8, 10.1) - Finite state machines (FSMs); Fixed-point numbers
	Lecture 8: Locals, parameters (Section 7.1-7.5, 9.1) - Local variables; Stack frames and parameter passing; I/O synchronization
Week 8	Lecture 9: C programming (C Primer) - Introduction to C; Structure of a C program; Functions, expressions
	Lecture 9: C programming (C Primer) - Variables in C; Structures and data types; FSMs in C
Week 9	Lecture 10: LCD interface (Section 8.4, 10.1, 10.5) - LCD programming; Number conversions; Review for Test 2
	Test 2 (closed book): In lab (during class time), ACA 1.102 & 1.106 Chapter 5 and 6
Week 10	Lecture 10: Interrupts (Section 9.2, 9.4, 9.6, 9.10) -9S12 interrupts and interrupt processing; Timer and periodic (output compare) interrupts
	Lecture 10: C programming – Metrowerks; C/assembly interface
Week 11	Lecture 11: D/A conversion (Section 11.2-11.3) - Digital to analog conversion (DAC) on the 9S12; Sound generation
	Lecture 11: A/D conversion (Section 11.1, 11.4) - Analog to digital conversion (ADC) Lab 7 design methods
Week 12	Lecture 12: Numerical calculations (Section 10.2) - Multiplication and division; Table lookup and interpolation
	Lecture 12: Serial I/O (Section 8.1) - Serial communications interface (SCI); SCI programming and interrupts; Lab 8 introduction

Date	Topic
Week 13	Lecture 13: Thread communication (Section 12.1-12.4) - Producer-consumer problems; FIFO queue
	Lecture 14: Stepper motors; More C programming
Week 14	More C Programming Thanksgiving
Week 15	More C Programming
	Review for Final Exam: Chapter 1 - 13

Lab Schedule

Labs are due on Wednesdays for all sections during your scheduled lab time.

Date	Task
08/24/11	*None*
08/31/11	*Go to ACA1.106 for Lab grading policy and demonstration*
09/07/11	**Lab1**: Digital Lock I/O, parallel port, direction register and logical function(simulated, individual)
09/14/11	***Lab2**: LED and Switch interfacing in Assembly (simulated, groups of two)*
09/21/11	*Real board demonstration, bring your board to lab*
09/28/11	***Lab3**: LED and Switch interfacing in Assembly (simulated and board, groups of two)*
10/05/11	**Lab4**: Debugging, one switch, one LED (simulated and board, individual)
10/12/11	**Lab5**: DNA sequence detector using the real 9S12 (simulated and board, Groups of two)
10/20/11	**Programming Test 2 (closed book), during regularly scheduled class time;** ACA **1.102 & 1.106 – No Lab due this week**
10/26/11	**Lab6**: LCD device driver, decimal fixed-point output, local variables (simulated and board, Groups of two)
11/02/11	*Go to ACA1.106 for Metrowerks demonstration*
11/09/11	**Lab7a/b**: Digital Piano/Pacemaker using 4-bit DAC (simulated and board, groups of two) – Mix of Assembly and C
11/16/11	**Lab8**: Real-time Position Monitor, ADC, Interrupts, LCD (simulated and board, groups of two) – Mix of Assembly and C
11/23/11	**No Lab**
11/30/11	**Lab9**: Distributed DAS, Serial Port Interrupts, FIFO queue (simulated and board, groups of two) – Mix of Assembly and C

Laboratory policies: See the Lab manual at **Link**

Lab Partners: See the Lab manual at **Link**

ABET Accreditation material

Three lecture hours and one laboratory hour a week for one semester.
Design Assignments: Labs 4, 7, 8 (1 week each)
Laboratory Projects: Labs 1, 2, 3, 5, and 6
SCH Engineering Topics 3 (Including: 1 SCH of Engineering Design)
Relationship of the Course to ABET EC2000 Program Outcomes:

√	**ABET EC2000 Program Outcomes**
√	a. An ability to apply knowledge of mathematics, science, and engineering
√	b. An ability to design and conduct experiments, as well as to analyze and interpret data
√	c. an ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability
	d. An ability to function on multi-disciplinary teams
√	e. An ability to identify, formulate, and solve engineering problems
	f. An understanding of professional and ethical responsibility
√	g. An ability to communicate effectively
	h. The broad education necessary to understand the impact of engineering solutions in a global, economic, environmental, and societal context
√	i. A recognition of the need for, and an ability to engage in life-long learning
√	j. A knowledge of contemporary issues
√	k. An ability to use the techniques, skills, and modern engineering tools necessary for engineering practice

ABET Criterion 9: Program Criteria for Electrical Engineering Curriculum Achieved:

Electical Engineering Program Criteria:

√	**Programs must demonstrate that graduates have a knowledge of:**
	1. Probability and statistics, including applications appropriate to the program name and objectives;
√	2. Mathematics through differential and integral calculus, basic sciences, computer science, and engineering sciences necessary to analyze and design complex electrical and electronic devices, software, and systems containing hardware and software components, as appropriate to program objectives.
√	3. (Electrical) Advanced mathematics, typically including differential equations, linear algebra, complex variables, and discrete mathematics.