EE 319K: Introduction to Embedded Systems
Sections - 16165, 16170, 16175, 16180

Andreas Gerstlauer

Spring 2013

General Information

Class Time	TTh 12:30-2:00pm
Classroom	**ACA 1.104**
Lab	**ENS 507**
Contact	gerstl@ece.utexas.edu
Office	ACES 6.118
Office hours	M 3:00-4:30 pm, W 1:30-3:00pm, or after class/by appointment
Website	UT Blackboard and class site: http://www.ece.utexas.edu/~gerstl/ee319k_s13
TAs	Sparsh Singhai Saugata Bhattacharyya Chinmaya Dattathri Emily Ledbetter Kevin Gilbert Siavash Zangeneh Kamali Yen-Kai Huang Michael Xing Nathan Quang Minh Thai Katherine Olin Mitchell Crooks Prachi Gupta Mark Meserve The TAs will hold office hours in the ENS 507 lab, a schedule will be posted there. About 16 students per 10-hour TA.
Mailing list	sp13_ee319k@utlists.utexas.edu (all Professors and TAs)

Teaching Philosophy

I strongly encourage students to take an active role in this class. Questions are welcome before, during and after class. Please email, visit my office hours or call me if you have any type of question.

Catalog Description

Embedded systems; machine language execution; assembly and C language programming; local variables and subroutines; input/output synchronization; analog to digital conversion and 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 ARM Cortex-M (Thumb) assembly and C; most labs will be first simulated and then run on the real LM3S1968 board. Software debugging occurs during the simulation stage. Verification occurs in both stages. Labs 1 through 6 are written in ARM Thumb assembly language. Labs 8 and 9 are written in a combination of assembly and C. Labs 7 and 10 are written in a C.

Prerequisites

EE306 or BME303 with a grade of at least C-. You should recall:

1. Bits
    a. Numbers - Unsigned and Signed Integer representation in 2's Complement
    b. True/False - Logical Operations
    c. Characters - ASCII representation
2. Gates
    a. AND, NAND, OR, NOR, NOT, XOR
    b. DeMorgan's Laws
3. Computer Components
    a. Central Processing Unit: Arithmetic Logic Unit, Control Unit, Registers
    b. Memory – ROM and RAM
        i. Address Space and Addressability
    c. I/O
4. LC3 Assembly Language
    a. Instruction Set (ISA)
    b. Pseudo-ops
    c. Op-codes and Operands
        i. Memory operations
        ii. Arithmetic and Logic operations
        iii. Control operations – Branches/Jumps
    d. Addressing Modes
5. Programming
    a. Algorithms, Flow-Charts
    b. Data Types
    c. Data Structures (Arrays and Linked Lists)
    d. Interrupts

Course Objectives and Outcomes

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 and C 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 techniques, e.g. breakpoints, scanpoints, profiles, monitors, voltmeters, oscilloscopes, logic analyzers.
7. How input/output actually happens (the students wire up analog and digital signals to the LM3S1968 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., 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 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.

Textbook and Reference Materials

Jonathan W. Valvano, *Embedded Systems: Introduction to ARM® Cortex™-M Microcontrollers*, Volume 1, Third edition, 12/2012, ISBN 978-1477508992. (available from Amazon)

Zyante, *Programming in C*, UT-EE319K edition. Available online:

1. Go to utaspr13.zyante.com on a supported browser (Chrome, Firefox, Safari).
2. Click "Subscribe" at the top-right corner.
3. Complete the required information. Please note:
   - Use your @utexas.edu email address to subscribe.
   - Be sure to select your course section from the "Membership List" drop-down!
4. Please use the coupon "utaspr13" to waive the $40 subscription fee.

Supplementary material:

- Reading assignments will be given from Patt's *Introduction to Computing Systems* (textbook for EE306)
- Manuals for the board, processor and tools used in class are linked from Prof. Valvano's EE319K site.
- Data sheets for most of the devices used in this class are also available as PDF files on Dr. Valvano's site.

Equipment

Board: Every group of two students will be required to have a Texas Instrument LM3S196 kit by the start of your lab period on Tuesday or Wednesday, 2/5 or 2/6. Since this is a new kit and we will be using the kit in EE319K and EE445L for a few years, you will have the option of selling it at the end of the semester. If you can afford it, we suggest each student obtain one kit at the start of the semester. The first possibility is to buy or borrow a LM3S1968 kit from a previous EE319K/EE445L student. If you obtain a kit from a previous student, you will need the LM3S1968 board, USB cable, two male headers (soldered onto the board) and a solderless breadboard. If you do buy/borrow a board from another student, I recommend you let the TA test it to make sure the board is functional. The second possibility is to purchase a new Texas Instruments LM3S1968 kit, which includes a board, USB cable, two male headers (not soldered on) and solderless breadboard. The student cost will be $60.00 per kit plus tax and is available from edu.mouser.com. You will have to log in using a University of Texas email (details on Blackboard).

Tools: Every student should own their own voltmeter and their own wire strippers. The NI MyDAC you bought in EE302 can be used as a voltmeter. Otherwise, a voltmeter less than $20 will do, see for example:

BG Micro:
http://www.bgmicro.com/MET1014.aspx

Jameco:
http://www.jameco.com/webapp/wcs/stores/servlet/Product_10001_10001_220812_-1

Harbor Freight, has 3 locations around Austin, usually sells voltmeters for less than $10:
http://www.harborfreight.com/7-function-digital-multimeter-90899.html

Since you will be making only 5 solder joints all semester, all students will be able to share the soldering iron in lab. However, all EE319K students will need their own voltmeter and wire strippers.

Safety warning: 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.

Kit handed out by TAs:

- 1 7406, six open-collector drivers used for the LED interfaces
- 6 LEDs (20 mA, 2 red, 2 yellow, 2 green)
- 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
- 2 feet of 4-Wire 24 gauge solid wire, used for prototyping
- 1 0.1 uF ceramic bypass cap, place across power and ground of the 7406
- 1 headphone jack, used in Lab 7 to create audio
- 1 20k ohm slide pot, used in Labs 8 and 9 to measure distance

Those do not need to be returned.

Items to get from TAs:

- Serial cable for Lab 9, used to connect one LM3S1968 board to another

Long term checkout:

- 1 LCD display

LCDs need to be returned by 5/4 or your grade will be reduced by one letter.

Computer: We expect every EE319K student to have a personal laptop. In addition to being required to complete Exam 2, a personal laptop on which you develop your software and write your reports will greatly simplify lab completion. For example (and these are personal opinions of the EE319K instructors), both software development and report writing should be done without paper, pencil and erasers. Also, 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.

You will need to install Keil uVision for the ARM on your laptop. For drawing electrical circuit diagrams some students use PCBArtist. Both of these applications run on a PC computer running Windows XP, Vista or Windows 7. The software development systems will run on any PC with a USB port. However, the more money you invest, the happier you'll be. All software will also run on a Macintosh with an Intel processor; see http://users.ece.utexas.edu/~valvano/On_a_Mac.htm.

Lab computer usage: Computers in ENS507 are available for your usage. TAs in the laboratory are checking off programs and supervising while on duty, thus you can expect to have only a brief consultation with them. You should learn to develop software while on the computer. This course involves some projects that require extended periods of time to complete and a project cannot be done just overnight. Get started on an assignment early so you can get help if you need it. There are not enough machine hours to give everyone more than enough time to complete each program. We expect students to have a laptop to run at home. Unfortunately, expect the laboratory to be crowded and machine time to be scarce if you attempt to work in the lab a day or two before an assignment is due. You will be competing with students in other courses for computer time. We need your assistance in the laboratory. Please report any equipment problems to the TA if they are present. If a TA is not present use the form provided in the lab. If you do this we can rapidly get repair service. Please help to keep the lab clean.

Software

Keil uVision: Instructions on how to download and install the (free) tool chain for our board can be found at http://users.ece.utexas.edu/~valvano/Volume1/uvision/.

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 is available at http://www.4pcb.com/free-pcb-layout-software. The use of PCBArtist is optional for EE319K students.

Grading Criteria

Task	Date	Percentage
Homework/Programming Assignments	Usually due on Tuesdays	10%
Laboratory Assignments	Due at lab times (Tuesdays/Wednesdays)	30%
Test 1 (in class)	Thursday, February 21, 12:30-2:00pm	15%
Test 2 (in lab, room TBA)	Thursday, March 28, 7:00-9:00pm	20%
Final Exam	Saturday, May 11, 7:00-10:00pm	25%

There will be no re-tests, make-ups, or incompletes.

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.

Laboratory Policies and Lab Partners: See the EE319K Laboratory Manual, available at http://users.ece.utexas.edu/~valvano/Volume1/EE319K_LabManualSp13.pdf

Homework Assignments: There will be approximately one homework programming assignment each week. All homeworks will be submitted in class as one piece of printed paper. You may work in groups and discuss problems with others, but every student has to independently develop and submit their own, separate homework solution. Late homework submissions will not be accepted under any circumstances.

Tentative Lecture Schedule

Week	Reading	Topic
1	Ch. 1 Syllabus	Introduction: Course administration, Embedded systems Electronics, Logic, Numbers: Binary, Decimal, Hexadecimal
	Ch. 2 Zyante Ch. 1	Development cycle, Flow charts, Data flow and call graphs Program development, Introduction to C, Keil tools
2	Ch. 3, Sec. 5.1 Patt Ch. 11, 12	ARM architecture and execution, Simple addressing modes C Programming: Structure of a C program
	Sec. 4.1.2, 4.2 Zyante Ch. 2	I/O: Parallel Ports, Direction registers, Logical and shift operations C variables, assignments and boolean expressions

Week	Reading	Topic
3	Sec. 4.6, 5.8 Patt Ch. 15, 13	Debugging: Monitors, Breakpoints, Single stepping Debugging using Keil tools
	Ch. 3, Sec. 5.3 Zyante Ch. 3, 4	Arithmetic: Addition/subtraction operations, Condition code bits Conditionals, Control structures: If-then, Loops
4	Sec. 4.2 Patt Ch. 14	Board: Demonstration of the board (bring your board to class), Switch Input and LED Output
	Sec. 5.2 Zyante 6.1-6.6	Modular programming: Subroutines and the Stack C functions
5	Sec. 6.1-6.3 Patt Ch. 16	Pointers: Indexed Addressing, Arrays, Strings C pointers
	Sec. 4.4, 6.9 Zyante Ch. 5	Timers, Advanced functional debugging (dump) C arrays and strings
6	Ch. 1 - 5	Review for Test 1
	Ch. 1 - 5	**Test 1 (closed book), in class**
7	Sec. 6.4, 6.5 Zyante Ch. 7	Finite State Machines (FSMs): Structures, Linked structures C structures
	Patt Ch. 19 Zyante Ch. 8	Abstract data types, Linked lists FSMs in C
8	Sec. 7.1-7.7 Patt Ch. 14	Local variables, Stack frames and Parameter passing C calling conventions, parameter passing, local/global scoping
	Sec. 7.6-7.8, 9.1 Zyante 6.7-6.13	Fixed-point numbers, Conversions LCD interface: LCD programming, I/O Synchronization
9	-	*Spring Break*
10	Patt Ch. 17 Zyante Ch. 10	Recursion, Recursive algorithms Recursion in C, Recursive stack frames
	Sec. 9.1-9.4, 9.6 Keil docs	Interrupts: Interrupts and interrupt processing, Thread synchronization Timer and periodic (SysTick) interrupts, C/assembly interface
11	Sec. 10.1-10.3 Previous Exam 2	Digital-to-Analog Conversion (DAC): Sound Generation Review for Test 2
	Ch. 1-7 Previous Exam 2	**Test 2 (closed book), in lab (room TBA), 7-9pm** Bring laptop with wireless internet access and Keil tools installed
12	Sec. 10.4-10.6	Analog-to-Digital Conversion (ADC): Lab 8 design methods
	Sec. 6.8, 7.6	Numerical calculations: Multiply/divide, Table Lookup, Interpolation
13	Sec. 8.1-8.2	Serial I/O: Serial Communications Interface (SCI), Serial interrupts
	Sec. 11.1-11.4	Thread communication: Producer-consumer problems, FIFO queue

Week	Reading	Topic
14	Patt Ch. 11-19	Advanced C programming: algorithms & data structures
	Zyante Ch. 1-10	Advanced C programming: software engineering & testing
15	-	Embedded system design trends & technologies
	-	Wrapup, Outlook
16	Ch. 1 – 11	Review for Final Exam
	Ch. 1 – 11	**Design Competition (in class)**
Finals	Ch. 1 – 11	**Final Exam, regularly scheduled time and place**

Tentative Lab Schedule

Week	Task
1	None
2	Go to ENS 507 for demonstration, install Keil tools on your laptop before coming to lab
3	**Lab1**: Digital I/O, parallel port, direction register and logical function, written in assembly (simulated, groups of two)
4	Real board demonstration, bring your board to lab. Sometime after Lab 2 and before Lab 3 arrange to have your TA solder the 66 pins onto the board.
5	**Lab2**: For-loops, while-loops, if-then branching, subroutines and time delays, written in assembly (simulated and board, groups of two)
6	**Lab3**: LED and switch interface, written in assembly (board, groups of two)
7	**Lab4**: Debugging techniques, one switch, one LED, written in assembly (simulated and board, groups of two)
8	**Lab 5**: Traffic Light Controller, written in assembly (simulated and board, groups of two)
9	*Spring Break*
10	**Lab 6**: LCD device driver, decimal fixed-point output, local variables, written in assembly (simulated and board, groups of two)
3/28	**Test 2 (closed book), 7-9pm, room TBA,** bring laptop with internet access and Keil tools.
12	**Lab 7**: Digital piano using a 4-bit DAC, written in C (simulated and board, groups of two)
13	**Lab 8**: Real-time position monitor, ADC, interrupts, LCD, mixture of assembly and C (simulated and board, groups of two)
14	**Lab 9**: Distributed DAS, serial port interrupts, FIFO queue, mixture of assembly and C (simulated and board, groups of four)
15	
16	**Lab 10:** Design competition, written in C

Tentative Homework Schedule

Date	Reading	Task
1/15		
1/22	Zyante Ch. 1	**Homework 1:** Definitions and numbers, Introduction to C
1/29	Zyante Ch. 2	**Homework 2:** Variables, statements, expressions
2/5	Zyante Ch. 3, 4	**Homework 3:** Control structures, functions
2/12		**Homework 4:** Arithmetic
2/19		**Homework 5:** Pointers, Indexing
2/26	Zyante Ch. 5	**Homework 6:** Arrays and Strings
3/5	Zyante Ch. 7	**Homework 7:** Structures
3/19		**Homework 8:** Practice Exam 2
3/26		**Homework 9:** Practice Exam 2
4/2	Zyante Ch. 6	**Homework 10:** Solve Exam 2 in C (locals, parameter passing)
4/9	Zyante Ch. 8	**Homework 11:** Data structures
4/16	Zyante Ch. 10	**Homework 12:** Recursion
4/23		
4/30		

Legal Notices

Add/Drop: The 12th class day is Wednesday, January 30, 2013. The drop policy is extremely complicated. See your academic advisor or the Dean of Students for more information. April 1 is the last day an undergraduate student may, with the dean's approval, withdraw from the University or drop a class except for urgent and substantiated, nonacademic reasons.

Course Evaluations and Final Exam: Course evaluation is conducted in the last class week in accordance with the Measurement and Evaluation Center form. The final exam is at the time and place stated in the course schedule.

Academic 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 and homework 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, in homeworks 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.

University Honor Code: "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." (see the university catalog)

Electronic Mail Notification Policy: In this course e-mail will be used as a means of communication with students. You will be responsible for checking your e-mail regularly for class work and announcements. The complete text of the University electronic mail notification policy and instructions for updating your e-mail address are available at http://www.utexas.edu/its/policies/emailnotify.html.

Use of Blackboard and Class Web Site:This course uses the class web page and Blackboard to distribute course materials, to communicate and collaborate online, to submit assignments and to post solutions and grades. You will be responsible for checking the class web page and the Blackboard course site regularly for class work and announcements. As with all computer systems, there are occasional scheduled downtimes as well as unanticipated disruptions. Notification of disruptions will be posted on the Blackboard login page. Scheduled downtimes are not an excuse for late work. However, if there is an unscheduled downtime for a significant period of time, I will make an adjustment if it occurs close to the due date.

Religious Holidays: Religious holy days sometimes conflict with class and examination schedules. If you miss an examination, work assignment, or other project due to the observance of a religious holy day you will be given an opportunity to complete the work missed within a reasonable time after the absence. It is the policy of The University of Texas at Austin that you must notify each of your instructors at least fourteen days prior to the classes scheduled on dates you will be absent to observe a religious holy day.

Students with Disabilities:The University of Texas at Austin provides upon request appropriate academic accommodations for qualified students with disabilities. For more information, contact the Office of the Dean of Students at 471-6259, 471-4641 TTY or the College of Engineering Director of Students with Disabilities at 471-4382.

ABET Accreditation Material

Three lecture hours and one laboratory hour a week for one semester.

Design Assignments: Labs 5, 8, 9, 10 (1 week each)

Laboratory Projects: Labs 1, 2, 3, 4,, 6 and 7

SCH Engineering Topics 3 (Including: 1 SCH of Engineering Design)

Relationship of the Course to ABET EC2000 Program Outcomes*:*

√	**1. 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:

√	**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.