

EE319K Introduction to Embedded Systems, Fall 2013 (8/25/2013 version)

[Email all professors and TAs \(f13_ee319k@utlists.utexas.edu\)](mailto:f13_ee319k@utlists.utexas.edu)

Course 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 assembly and C; most labs will be first simulated then run on the real LM4F123/TM4C123 board. Software debugging occurs during the simulation stage. Verification occurs in both stages. Labs 1, 2, 3, 4, 5 are written in ARM assembly language. Labs 8 and 9 are written in a combination of assembly and C. Labs 6 and 10 are written in C.

Yerraballi/Valvano class: ENS127, Monday, Wednesday, 12:00-1:30pm Lab: ENS507

Yerraballi/Valvano class: ACA1.104, Tuesday, Thursday, 3:30-5:00pm Lab: ENS507

Gerstlauer Class: ENS115, Monday, Wednesday, 1:30-3:00pm Lab: ENS507

EE319K is team-taught. All three lecture sections will share the same TAs, PowerPoint slides, homework assignments, lab assignments, Exam 1, Exam 2, and Final Exam. EE319K will give common Exams 1 and 2 in the evening. The lectures for the two Yerraballi sections will be shared between Yerraballi and Valvano, alternating on a weekly basis.

MW12-1:30 Yerraballi/Valvano lecture, ENS127, Unique Numbers (lab times):

16570 T 1:00 to 2:00pm

16575 Th 1:00 to 2:00pm

16580 T 2:00 to 3:00pm

16585 Th 2:00 to 3:00pm

16590 T 3:00 to 4:00pm

16595 Th 3:00 to 4:00pm

MW1:30-3:00 Gerstlauer lecture, ENS115, Unique Numbers (lab times):

16600 Th 12:00 to 1:00pm

16602 W 4:00 to 5:00pm

16603 W 5:00 to 6:00pm

16605 Th 3:00 to 4:00pm

TTh3:30-5:00 Yerraballi/Valvano lecture, ACA1.104, Unique Numbers (lab times):

16610 W 9:00 to 10:00am

16615 W 10:00 to 11:00am

16620 W 11:00am to 12:00noon

16625 W 12:00noon to 1:00pm

Instructors:

Ramesh Yerraballi, ENS106, (512) 471-2080, ramesh@mail.utexas.edu,

<http://www.ece.utexas.edu/~ryerraballi>

Jonathan W. Valvano, ENS627, (512) 471-5141 valvano@mail.utexas.edu,

<http://www.ece.utexas.edu/~valvano>

Andreas Gerstlauer, ACE 6.118, (512) 232-8294 gerstl@ece.utexas.edu,

<http://www.ece.utexas.edu/~gerstl/>

Office Hours (subject to change):

Ramesh Yerraballi, MW: 2:00-4:30pm, Friday 1:30-3:30pm

Jonathan W. Valvano, Tuesday 11-12, Thursday 12-1, Friday 12-1, by appointment

Andreas Gerstlauer, M 3-4:30pm, T 1:30-3pm, or after class/by appointment

TAs (Photos of TAs): TAs will hold office hours in the ENS507 lab. The office hours will be posted in lab. About 32 students will be assigned to each 20-hour TA, and 16 students to each 10-hour TA.

Saugata Bhattacharyya, saugata .at. utexas.edu

Chinmaya Dattathri, d.chinmaya .at. utexas.edu

Prachi Gupta, Prachi.gupta .at. utexas.edu

Emily Ledbetter, emily.k.ledbetter .at. gmail.com

Wooseok Lee, wooseok.lee .at. utexas.edu

Katherine Olin, katherineolin .at. gmail.com

plus one more TBD

Mailing list: f13_ee319k@utlists.utexas.edu (all Professors and TAs)

EE319K Class Web page (Lecture notes, homeworks and lab assignments):

<http://www.ece.utexas.edu/~valvano/Volume1>

Data sheets: <http://www.ece.utexas.edu/~valvano/Datasheets>

Starter files: <http://www.ece.utexas.edu/~valvano/arm/> Look for examples that have **4F120** in the name.

Text: [Embedded Systems: Introduction to ARM Cortex-M Microcontrollers](#), Fourth Edition. The fourth edition has a lot more information on the LM4F120/TM4C123 compared to the third edition. However, you are allowed to buy/borrow a used third edition, all editions have the same ISBN: 978-1477508992. This means if you buy it used you are likely to get an older edition. [Available from Amazon](#) (not available in local bookstores):

<http://www.amazon.com/Embedded-Systems-Introduction-Arm-Cortex/dp/1477508996>

<http://users.ece.utexas.edu/~valvano/arm/outline1.htm>

Reference materials:

- EE319K will have reading assignments from Patt's Introduction to Computing Systems (textbook for EE306 and BME303).
- Manuals for the board, processor and tools used in class are linked from the [class web page](#).
- Data sheets for most of the devices used in this class are available as PDF files on the [Data sheet site](#).

Equipment to buy:

1. We expect each EE319K student to have a personal laptop. The laptop minimum specifications listed on <http://www.ece.utexas.edu/it/laptop-initiative> will be acceptable for EE319K. We think adding more RAM will make the operating system run faster. This

laptop will greatly simplify lab completion, and it is required to complete Exam 2. However, if having a laptop presents difficulty, please see your professor to see what options are available. The EE319K staff feels very strongly that you should have your own computer on which you develop your software and write your reports. We 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 their EE319K lab work on campus using the computers in ENS507, but most find it convenient to configure their laptops to work with the EE319K hardware. This way they have a common platform while working at home as they have while demonstrating to the TA in lab. To work on your laptop, you will need to install Keil uVision for the ARM. For drawing electrical circuit diagrams some students use PCBArtist. Both of these applications run on a PC computer running Windows XP, Windows 7, or Windows 8. These software development systems will run on any PC with a USB port.

2. Every team of two students will be required to have a microcontroller board (EK-TM4C123GXL, EK-LM4F120XL, or EKK-LM3S1968). Although we allow you to use the LM3S1968, if you haven't purchased a board yet, we recommend either of the two Stellaris LaunchPads (EK-TM4C123GXL or EK-LM4F120XL). You should have the microcontroller board by the start of your lab period the week of 9/16. 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. We recommend you purchase a new Texas Instruments Stellaris LaunchPad, which includes a board, and the microB USB cable. The cost will be \$12.99 plus tax and is available from many sites, but the cost will vary from \$8 to \$23 depending on where you order it. Obtain any of these three kits:

- EK-TM4C123GXL (newer and more costly, but harder to get)
<http://www.ti.com/tool/ek-tm4c123gxl>
- EK-LM4F120XL (the lowest cost and easier to get, but available only in August) <http://www.ti.com/tool/EK-LM4F120XL>
- EKK-LM3S1968 (buy used from a previous EE319K or EE445L student)

The Stellaris LaunchPad can also be purchased at regular vendors like

- <http://www.digikey.com> (EK-TM4C123GXL is part number 296-35760-ND, EK-LM4F120XL is part number 296-34897-ND)
- <http://www.mouser.com> (595-EK-TM4C123GXL or 595-EK-LM4F120XL)

We will allow our students to use either kit, the same code runs on both (the TM4C123 has hardware PWM and USB host). Another possibility is to buy or borrow a EKK-LM3S1968 kit from a previous EE319K/EE445L student. If you obtain a kit from a previous student, you will need the EKK-LM3S1968, and a miniB USB cable. If you do buy or borrow a board from another student, I recommend you let the TA test it to make sure the board is functional. Caveat about the LM3S1968: The TAs will understand the details of the LM3S1968 because we used it the last two semesters, but your lab manual and lectures will focus on the TM4F123/LM4F120. So to use the LM3S1968, some modifications to the lab manual will need to be negotiated with your TA. The biggest difference will be the game in Lab 10. The LM3S1968 has a little 4-bit grey scale display that you will own as part of your board, and the TM4F123/LM4F120 will use a bigger 16-bit color display that you will borrow and return.

3. Every student should have a solderless breadboard. There are many options for finding one of these. In the building, you can get one from the IEEE student society. In town, you can get one at Fry's or RadioShack. On the internet there are many options for the solderless breadboard. We recommend the Twin Industries [TW-E40-1020](#), which is available many places:

- Newark 56T0250 \$8.99
- Mouser 589-TW-E40-1020 \$7.11
- Allied Electronics 70012446 \$7.79
- Digi-Key 438-1045-ND \$8.98

There should be four power strips and at least 400 points, so little boards are also ok:

- Newark Twin Industries TW-E40-510 56T0249 \$7.19
- Mouser Twin Industries TW-E40-510 589-TW-E40-510 \$7.18

You can wait to buy the breadboard until classes start, because it will be cheaper to get together with other students to save shipping costs. PLEASE DO NOT BORROW OR BUY A USED BREADBOARD. Used breadboards can develop weird and confusing faults (pins not connected or multiple pins inadvertently shorted) and could cost you hours and hours of frustration. We recommend you wait until classes start to buy the breadboard, this way 3 or 4 students can team up and make one combined order and save on shipping. (On the other hand, we do recommend you get the microcontroller board as soon as possible).

4. Every student should own their own voltmeter and their own wire strippers. A voltmeter less than \$20 will do, see BG Micro <http://www.bgmicro.com/MET1014.aspx>, or Jameco http://www.jameco.com/webapp/wcs/stores/servlet/Product_10001_10001_220812_-1. Harbor Freight has three locations around Austin and 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.

Course 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.

Detailed Objectives of EE319K:

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 and C 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, such as breakpoints, scanpoints, profiles, monitors, voltmeters, oscilloscopes, logic analyzers.
7. How input/output actually happens (the students wire up analog and digital signals to the TM4C123 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 input/output).
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 "1.234 cm".
11. Interrupt synchronization, real-time ADC sampling (periodic timer interrupts), introduction to multithreaded programming.
12. Digital to analog conversion (DAC), used to make simple sounds.
13. Design and implementation of elementary data structures, such as linked lists, stacks and queues.

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
 - c. Instruction Set (ISA)
 - d. Pseudo-ops
 - e. 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. Arrays
 - d. Interrupts

Teaching Philosophy

The EE319K staff 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 us if you have questions.

Grading

10%	Homework/Programming Assignments	Weekly, turned in during class
30%	Laboratory Assignments	Due at lab times (Tue/Wed/Thu)
15%	Exam 1 (in evening, <i>room TBA</i>)	Thursday, October 3, 7:00-8:30pm
20%	Exam 2 (in lab, <i>room TBA</i>)	Thursday, November 7, 7:00-9:00pm
25%	Final Exam (common exam)	<i>Time and place to be announced</i>

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

Cutoff scores for the letter grades will not be determined until after the final exam.

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 first few pages of the [Lab manual](#).

Exams: All students in EE319K will take Exam 1, Exam 2 and the Final Exam at the same time and place. Exam 1 is a regular written test, but outside of the normal class time and room. Exam 2 is a practical programming exam during which you will design, implement, and debug a software system in assembly. Your exact room will be assigned to you by your professor. Each student will need a laptop that can run for 100 minutes on battery power. You will need wireless internet connection for the first and last 10 minutes of the exam. You will be writing software using Keil uVision in simulation mode. No microcontroller hardware will be needed. During the exam you will not have access to the book, to the internet, to any starter code, or to any of your previously downloaded or written software. You will have access to the two page instruction sheet handout (like Exam 1), and the help system in Keil uVision. Please contact your professor by October 10 for any of the following four reasons: a) you do not own a laptop that can run simulation of uVision; b) your laptop runs but will need AC power because of a weak battery; c) your laptop runs but does not have wireless connection; d) you have a schedule conflict that prevents you from taking the test at the given time.

Tentative Lecture Schedule

The following schedule is preliminary; please check the web for the most recent version.

Week	Lecture	Lab	Reading	Lecture Schedule
1 (8/26)	Lec1	-	Ch. 1, Ch. 2	Introduction – Course administration; Embedded systems, development cycle; Flow charts, data flow and call graphs; Electronics, logic, Ohm's law; ARM programming and Keil uVision tools.
2 (9/2) no class 9/2	Lec2	-	Ch. 3 (3.1-3.3), Sec. 4.1.3, 4.2.1 (Patt Ch. 11, 12)	Architecture – ARM architecture and execution; Simple addressing modes; Introduction to C programming; Structure of a C program. I/O – Parallel ports, direction registers; Logical and shift operations; C variables, assignments and Boolean expressions.
3 (9/9)	Lec3	Lab 1 Lock	Sec. 4.6, 5.8, Sec. 4.2.2 (Patt Ch. 15)	Debugging – Debugging techniques, monitors, breakpoints, single stepping; Debugging using Keil uVision. Board – Demo of the board (in class); Switch input and LED output
4 (9/16)	Lec4	Board Demo	Ch. 5 (5.1-5.3), Sec. 3.3.6, 3.3.8 (Patt Ch. 13, 14)	Modular programming – Abstraction and refinement; Subroutines, parameters, stack; C functions. Arithmetic – Arithmetic operations; Overflow, condition code bits; Control structures, branches, if-then, loops.

5 (9/23)	Lec5	Lab 2 Toggle	Ch. 6 (6.1-6.3), Sec. 6.9 (Patt Ch. 16)	Pointers – Indexed addressing; Arrays; Strings; C pointers, arrays, indexing, strings. Advanced functional debugging – Debugging dump.
6 (9/30)	Lec6	Lab 3 Board	Sec. 4.4 Ch. 1-5	Timers – SysTick Timers; Review for Test 1. Test 1 (closed book), Thursday, 10/3, 7-8:30pm, room TBA
7 (10/7)	Lec7	Lab 4 Debug	Sec. 6.4, 6.5, Sec. 9.1 (Patt Ch. 19)	FSMs – Finite state machines (FSMs); Linked structures; C structures and data types; FSMs in C. I/O synchronization – Blind cycle, busy wait, interrupts; Phase locked loop (PLL).
8 (10/14)	Lec8	Lab 5 FSM	Ch. 9 (9.1-9.4), Sec. 9.6, Sec. 10.1-10.3	Interrupts – ARM interrupts and interrupt processing; Interrupt controller; Thread synchronization; Periodic interrupts; C/assembly interface; D/A conversion – Digital to analog conversion (DAC); Resistor networks; Sound and music generation.
9 (10/21)	Lec9	Lab 6 DAC	Ch. 7 (7.1-7.8) (Patt Ch. 17)	Variables, parameters – Locals; Scope (local/global); Stack frames, recursion; Parameter passing, call-by-value, call-by-reference. LCD interface – LCD programming; Fixed-point numbers, number conversions.
10 (10/28)	Lec10	Lab 7 LCD	Sec. 10.4-10.6 Sec. 7.6	A/D conversion – Analog to digital conversion (ADC); Nyquist theorem; Lab 8 design methods. Numerical calculations – Multiplication and division; Interpolation and calibration.
11 (11/4)	Lec11	- Exam 2	Ch. 1-7	Review for Test 2. Test 2 (closed book), Thursday 11/7, 7-9pm, in lab (room TBA).
12 (11/11)	Lec12	Lab 8 ADC	Ch. 8 (8.1, 8.2), Ch. 11 (11.1-11.4) (Patt Ch. 19, skip Sec. 19.4)	Serial I/O – Universal asynchronous receiver transmitter (UART); UART programming and interrupts; Lab 9 introduction Thread communication – Producer-consumer problems; FIFO queues; FIFOs and queues in C.
13 (11/18)	Lec13	Lab 9 UART	Sec. 6.7, 9.7	Game programming – 2-D arrays, bitmaps, sprites; Game engine design; Data structures, timer interrupts; Lab 10.
14 (11/25)	Lec14	-		Floating Point; Advanced Embedded System Design. <i>Thanksgiving holiday, no class Thursday 11/28</i>
15 (12/2)	Lec15	Lab 10 Game	Ch. 1-11	Review for Final Exam. Game Competition, Friday, 12/6, 12-1pm, room TBA

Lab Schedule

Labs are due during your lab time (Tue/Wed/Thu), and are demonstrated to the TA.

There will be a sign-up for Lab checkout times. Each student will have a specific checkout time which will be when their lab will be demonstrated to their TA.

You can only switch lab sections using official add/drop procedures.

Week	Lab	Task
8/26		None, do not go to lab.
9/2		Go to ENS 507 for demonstration, install Keil tools on your laptop before.
9/9	Lab 1	Digital lock, I/O, parallel port, direction register and logic function, written in assembly (simulated, groups of two)
9/16		Real board demonstration, bring your board to lab.

9/23	Lab 2	For-loops, while-loops, if-then branching, subroutines, and time delays, written in assembly (simulated and board, groups of two)
9/30	Lab 3	LED and switch interface, circuit measurements, schematics, written in assembly (board, groups of two)
10/7	Lab 4	Debugging techniques, one switch, one LED, written in assembly (simulated and board, groups of two)
10/14	Lab 5	Traffic Light Controller, FSMs, written in assembly (simulated and board, groups of two)
10/21	Lab 6	Digital Piano using a 4-bit DAC, written in C (simulated and board, groups of two)
10/28	Lab 7	LCD device driver, decimal fixed-point output, local variables, written in assembly (simulated and board, groups of two)
11/4	Exam 2	Thursday, 11/7, 7-9pm, room to be announced (closed book)
11/11	Lab 8	Real-time position monitor, ADC, interrupts, LCD, mixture of assembly and C (simulated and board, groups of two)
11/18	Lab 9	Distributed DAS, serial port interrupts, FIFO queue, mixture of assembly and C (simulated and board, groups of two)
11/25		<i>Thanksgiving, no lab</i>
12/2	Lab 10	Game design competition, written in C (simulated and board, groups of two)

Tentative Homework Schedule

The following schedule is preliminary; please check the web for the most recent version.

Homework is 1-2 pages printed and handed to the TA at the start of the first lecture of the week.

Week	Homework	Topic
8/26	-	-
9/2	Homework 0	
9/9	Homework 1	
9/16	Homework 2	
9/23	Homework 3	
9/30	Homework 4	
10/7	Homework 5	
10/14	Homework 6	
10/21	Homework 7	Practice Exam 2
10/28	Homework 8	Practice Exam 2
11/4	Homework 9	
11/11	Homework 10	
11/18	Homework 11	
11/25	-	
12/2	-	Pop quiz (course evaluations)

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 brief consultations 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 or have access to a machine 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.

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

- 1 7406, six open collector drivers used for the LED interfaces
- 6 LEDs (20 mA, 2 red, 2 yellow, and 2 green LEDs)
- 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 to prototyping (you can get more wire, just ask your TA)
- 1 0.1 uF ceramic bypass cap, place across power and ground of the 7406
- 1 headphone jack, used in lab 6 to create audio
- 1 20k ohm slide pot, used in labs 8 and 9 to measure distance

Items to get from long term checkout

- 1 Kentec EB-LM4F120-L35 display (if you check out the LCD and do not return it and its bag by 12/6, we will reduce your overall grade in EE319K one letter grade). If you loose this display it will also cost you about \$45 to buy us a new one.

Items to get from your TA

Serial cable for Lab 9, used to connect one LM4F123/TM4C123 board to another (this cable is simply 2 wires about 2 feet long).

Software to install on your laptop

- **Keil uVision.** Install Keil uVision for the ARM on your personal computer (it is free):
 1. Go to <https://www.keil.com/demo/eval/arm.htm>
 2. Enter your contact information with valid address, phone and email:
Company = University of Texas, devices = TI, architecture = Cortex-M4
 3. Click submit.
 4. Right-click on mdk472_a.exe (actual file may be a newer version) and save it to your computer.
 5. Run the exe to install.
- **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/free-pcb-layout-software/> or <http://www.4pcb.com/> or. The use of a drawing program like PCBArtist is required for EE319K students. You are allowed to use a different circuit drawing program. However, if you plan to use a different circuit drawing program, please get it approved by your TA.

Legal Notes

The 12th class day is September 13. The drop policy is extremely complicated. 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. November 5 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.

Students with disabilities: 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 the instructor 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, we will give you an opportunity to complete the missed work within a reasonable time after the absence.

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, we will make an adjustment if it occurs close to the due date.

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.

A practical guide to ethics involving software development for EE319K labs

Activities you can and should do

1. You are supposed to share everything with your partner. If you form a Lab 10 team with a different student from your partner in labs 3-9, then you can share your lab 3-9 code with the new partner. Make sure the code contains the proper names of who actually helped develop it.
2. You can and should discuss strategy, theory, approach, and debugging tips with past, present, and future students in this class. This includes flowcharts, data flow graphs, call graphs, pictures and descriptions of data structures.
3. You can and should read books and search the internet for existing software similar to the problem you have been asked to solve. You may copy-paste that software into your lab as long as you include a full reference of its source. In fact, we all stand on the shoulders of giants and in (large) software development projects, code reuse and building on top of rich libraries is often a key strategy to success. However, proper credit needs to be given and permission to (re)use code from others needs to be obtained. Anything else is cheating and stealing. You are of course also responsible for understanding all software used in your lab solution.
4. You can look at up to 3 or 4 lines of code from another EE319K student as part of the discussion described in 1), as long as you are not observing the entire function.

Activities you cannot and should not do

1. If you form a Lab 10 team with a different student than your partner in Labs 3-9, then you cannot share any newly written lab 10 code with the old partners.
2. You cannot copy any software either electronically or optically (look at it and type it) from any past, present or future EE319K students. This includes lab solutions that another student has inappropriately posted on the internet. You are not allowed to look at

another student's lab solution and reproduce it with different formatting or variable names. This can be simply stated "do not look at Lab solutions of another student", with the 3-4 line exception listed above.

3. You may not copy-paste old lab solutions to another student. For example, you may not share your LCD driver from Lab 6 to another group while working on Labs 7-10. You cannot give your lab solutions you develop this semester to a student taking the course in the future.
4. When copy-pasting code from a book or the internet, you should never remove the author's name. See rule 2) above.
5. When copy-pasting code from a book or the internet, you should not expect the code to work (most likely it won't work as expected anyway); it is your responsibility to understand and debug all code used in your lab solutions.

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." <http://registrar.utexas.edu/catalogs/gi09-10/ch01/>

Abet 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:

<input checked="" type="checkbox"/>	ABET EC2000 Program Outcomes
<input checked="" type="checkbox"/>	a. An ability to apply knowledge of mathematics, science, and engineering
<input checked="" type="checkbox"/>	b. An ability to design and conduct experiments, as well as to analyze and interpret data
<input checked="" type="checkbox"/>	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
<input checked="" type="checkbox"/>	e. An ability to identify, formulate, and solve engineering problems
	f. An understanding of professional and ethical responsibility
<input checked="" type="checkbox"/>	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
<input checked="" type="checkbox"/>	i. A recognition of the need for, and an ability to engage in life-long learning
<input checked="" type="checkbox"/>	j. A knowledge of contemporary issues
<input checked="" type="checkbox"/>	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:

ELECTRICAL ENGINEERING PROGRAM CRITERIA

<input checked="" type="checkbox"/>	Programs must demonstrate that graduates have a knowledge of:
	1. Probability and statistics, including applications appropriate to the program name and objectives;

<input checked="" type="checkbox"/>	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.
<input checked="" type="checkbox"/>	3. (Electrical) Advanced mathematics, typically including differential equations, linear algebra, complex variables, and discrete mathematics.

Curious about Valvano's research? See <http://users.ece.utexas.edu/~valvano/research>

Curious about Gerstlauer's research? See <http://www.ece.utexas.edu/~gerstl/research.html>