USC Viterbi School of Engineering

EE 250L Distributed Systems & the Internet-of-Things

Units: 4
Fall 2021

Location: GFS 106 and online at http://bytes.usc.edu/ee250

Instructor: Mark Redekopp
Office: EEB-222 & https://usc.zoom.us/j/307718126
Office Hours: See website (likely outdoors or on Zoom)
Contact Info: redekopp@usc.edu;

Teaching Assistant and UG Mentors:
See website

Course Description

This is a 4-unit undergraduate course aimed at electrical engineering, computer engineering and computer science students with at least a sophomore standing. Students will design, build and evaluate systems that incorporate hardware, operating systems, network and application layer protocols. Focus will be placed on distributed system applications for the Internet of Things. Through hands-on lab activities, assignments, projects, as well as through guest lectures spanning research and practice, the course provides a comprehensive introduction to many relevant topics including printed circuit board design, real-time operating systems, programming microprocessors, signal processing, wireless communications, networks, control systems, publish-subscribe middleware, cloud computing, databases, machine learning and security.

Weekly labs will provide the opportunity for students to apply networking and embedded programming concepts. Several written homeworks will also be given to help reinforce the concepts taught in lecture.

Learning Objectives

Below are the specific, measurable skills a student will demonstrate by the end of the course. These objectives will be both taught and assessed in the course and are aligned with the assignments, assessments and learning materials.

1. Develop applications that use the TCP/IP protocol suite and socket programming to make embedded devices communicate over the Internet.
2. Use publish-subscribe protocols such as MQTT to share messages in real time between distributed devices all over the Internet
3. Explain what real-time embedded operating systems do and the kind of programmer-support interfaces they provide
4. Analyze how low power wireless links behave in terms of metrics such as signal quality, error rates, throughput, as a function of link distance, channel and power settings.
5. Apply and describe various physical signaling schemes and codes used in the transmission of digital data.
6. Explain how randomized medium access (MAC) protocols such as CSMA, TDMA, etc. that allow multiple wireless devices to share a medium
7. Create circuit schematics and use those schematics to create a printed circuit-board (PCB) design and layout and test a PCB's functionality
8. Apply sleep cycling and asynchronous MAC protocols to prolong the battery lifetime of an embedded device
9. Process raw data / signals from sensors to minimize noise and to estimate useful parameters such as a device's location
10. Compare and contrast cryptographic encryption schemes and security protocols to protect the confidentiality and integrity of data.

Prerequisite(s): EE 109L and either EE 155L or CS 103L
Co-Requisite(s): None.
Concurrent Enrollment: None.
Recommended Preparation: None

Course Materials

A set of lecture notes will be made available on the course website at http://bytes.usc.edu/ee250.

Raspberry Pi

Students will be expected to have access to a Raspberry Pi embedded computer that contains the following:

- Raspberry Pi 3 (Regular or B+) or 4
- Micro USB Power Supply
- 16GB+ Mini SD Card

A kit containing a Raspberry Pi 3b may be purchased at via Amazon (RPi 3b kit) though you will need a 16GB or greater SD card (via Amazon - SD card) if you don't have a free one already. A Raspberry Pi 4 should also work (though we may not offer full support) and can be purchased at: via Amazon (RPi 4 kit). If your laptop does not have an SD card reader, you will likely want to purchase one. Finally, you will also use a sensor kit: GrovePi Starter Kit from SeeedStudios. The sensor kits will be available and provided on loan, though some student enjoy purchasing one themselves for other projects and future use. If you'd like to do work remotely you will also need access to a wireless access point for which you can control the settings (home wifi access point or your cell phone).

Laptop

Students must also have a laptop capable of running Linux natively or via a virtual machine. A virtual machine will be made available on the course website.

Technological Proficiency and Hardware/Software Required

Students are expected to have basic proficiency in command line usage (primarily in Linux) as well as a firm grasp of the C programming language including functions, arrays, and pointers.

Course Websites

1. **Primary website**: All course assignments, content, office hour information, etc. will be posted at our main website: http://bytes.usc.edu/ee250. This site will contain a link to a shared Google Drive folder where we will post all slides, labs, and homeworks. Please ensure you are able to access it.
2. **Q&A website**: A Q&A and announcement website will be utilized: http://www.piazza.com/. All official announcements regarding assignments, lectures, exams, etc. will be made via Piazza. It is your responsibility to check this site often.
3. **Shared Google Folder**: All slides, HWs, Labs, and other relevant documents will be posted in this folder. We will give access to your @usc.edu account.
4. **Vocareum**: Lab and project code submissions will be made via http://www.vocareum.com. You will receive an invitation link to create your account sometime in the second week.
5. **Blackboard**: Blackboard (http://blackboard.usc.edu) will ONLY be used to record grades and for conducting class in the event of an emergency. Zoom recordings will be available on Blackboard.

Zoom

Lectures will be recorded (screen recording) though some aspects may be lost since video of other students and my movements and gesturing may not be captured. Recordings should post automatically on Blackboard. One of the lab introductions will also be recorded and posted to Zoom in case you work remotely.

Required Readings and Supplementary Materials

The following textbooks are **recommended** but not required. We recommend you read the sections listed on the course schedule below for the corresponding week **BEFORE** attending the first lecture of that week.

1. Practical Python Programming for the IoT, Gary Smart, Packt Publishing, 2020. Freely available via USC Libraries - will need to sign in to the libraries website to access the book
   Practical Python Programming
   Additional materials on Github:
   https://github.com/PacktPublishing/Practical-Python-Programming-for-IoT
2. IoT and Edge Computing for Architects, 2nd Edition, Perry Lea, Packt Publishing, 2020. Freely available via USC Libraries (sign-in required) IoT and Edge Computing
3. Fundamentals of Wireless Sensor Networks, Thoery and Practice. Dargie and Poellabauer. Wiley Publishing, 2010. ISBN 978-0-470-99765-9. ***Freely available*** *here: Fundamentals of Wireless Sensor Networks.*
4. Distributed Systems, Third Edition by Tanenbaum and Van Steen, 2017. ISBN 978-90-815406-2-9). ***Freely available*** *here: Distributed Systems .*
5. *Other online resources provided by the instructor.*

Homeworks

Availability: Assignments will be made available on the course shared folder. These assignments are predominantly for your own practice and thus will be graded as CR/NC. **No penalty will be applied for missing 1 homework**. You are strongly encouraged to work on these individually and seek help from a course staff if you struggle to answer the question (as opposed to looking at the solutions or the work of another student).

Solutions: Solutions to the homework problems will be available on the course shared folder.

Labs

Overview: There will be approximately nine lab assignments. Lab assignments are the primary out-of-class work and should challenge you to learn NEW concepts somewhat beyond the scope of the class as well as integrate what is taught in class. **They are usually NOT simply step-by-step procedures** but involve problem solving and application of various concepts. As such you should be prepared to have to read and search out further information online as you perform the labs. Some labs may involve designing and building circuits, others may solely involve writing software.

Assessment: For each lab you will need to demonstrate certain portions of the functionality via a video recording (posted to Youtube or Google folder) and linked in your submission on Vocareum. The remaining portion of the credit will come from visual inspection of your code after you submit it. **All code should be neatly indented and have ample comments.** Failure to indent or include comments will lead to point deductions.

Before Attending Lab: It is expected that you will read the lab procedure COMPLETELY and watch any posted videos related to the lab **before you attend**.

Demonstration Due Date: Labs are assigned during the Tuesday and Wednesday lab sessions and **must be demonstrated via a video (made with your phone) demonstrating the required tasks and capturing your screen to show you are compiling/running your code. These videos should be uploaded to Youtube or a Google folder and a viewable link provided with your code submission** (a few labs may span two weeks). The teaching assistants hold office hours in the VHE 205 classroom. We will explore holding Zoom office hours and update you via Piazza. A schedule of TA hours is posted on the class web site.

Submitting Your Code/Answers: Code and answers to lab questions (write-ups and/or program source code) must be submitted online (via Vocareum**) by midnight Friday of the week the assignment is due**.

Collaboration and Academic Integrity: Indicated lab assignments are to be completed either individually or in teams of two unless otherwise noted. **NO TEAMS OF 3 or MORE ARE ALLOWED**. When groups are used, pair-programming is expected (this means active participation of the student not physically at the keyboard). In addition, each person should have their own copy of the code so they can work in future weeks. During demonstrations, the TA may choose EITHER student to provide an explanation (rather than the team deciding who will provide an explanation) and will likely ask both students to provide input. **Copying (and then modification) or even LOOKING at or for any portion of code from Internet sources, fellow, or past students is prohibited** unless cleared with the instructor. We will be clear: **You are not to share or look at the code from another team**. See the Statement on Academic Conduct.

Late Submission: No credit will be given for demonstrations or code submissions after the due date(s). **It is your responsibility to submit on time. Don't procrasticate but submit early and often!** No excuses for WiFi connectivity, broken laptops, etc. will be allowed. This can easily be mitigated by pushing to Github regularly!

Grading Disputes: We will work hard to post LAB scores and feedback within 1 week of the lab's due date. Any disputes with posted grades must be raised within 7 days of the score posting. Notice that any regrade request will result in us trying to give the fairest possible grade to you, which could be higher or lower than the one you received originally. Please see your TA to start the process and hopefully resolve the issue.

Project

Overview: During the last week of two of the semester students will work on a more difficult lab or project selected by the instructor. If students have a particular project idea, they are welcome to propose it to the head TA, and may be allowed to purse the project in place of the given assignment. The project is expected to be a challenging lab including software and hardware components used during the semester.

Assessment: Projects will be evaluated not just on whether or not it works, but to a large extent on the quality of the hardware and software incorporated in it. One that operates well, is robust and reliable, and is user-friendly will receive more points than one that does not. A rubric will be posted along with the assignment. About half of the points will be based on the working demonstration. The other half will be based on the quality of the software code implementation as judged by visual inspection of particular rubric elements by our teaching staff.

Late Submission: No credit will be given for demonstrations or code submissions after the due date(s).

Exams

Time and Location: There will be one midterm. The date of the midterm is shown on the attached schedule but may be moved to a different date in exceptional cases. The exams may also be moved to a different classroom. Always check with the instructor as the listed exam date approaches to confirm the date and time. The exam dates will be announced in class and on the web site. You are responsible for finding out when and where the exams will be held. Makeup exams will be given if you have a valid excuse (e.g. serious illness or accident, urgent trip, but proof will be required).

Academic Accomodations: If you have USC approved academic accomodations, please check with your instructor 2 weeks before the exam to determine when and where you will take the exam.

Exam Style: Exams are designed to not only test your retention of the material but your ability to apply it to design and analyze new or novel problems. In this way, your mastery and depth of understanding of the course content will be assessed. Some portion of the exam will contain general knowledge questions and be fill-in/multiple choice. However, the majority of points will come from design problems or apply the skills learned in class. This is where struggling with the homework and lab problems on your own and until you truly understand and feel comfortable with each concept will greatly pay off. *Students who simply "get the lab done" without reviewing and understanding each facet will often struggle on the exam*s.

Grading Breakdown

Assignment	**% of Grade**
Homeworks	5%
	Note: Drop lowest HW
Labs	40%
Project	5%
Midterm 1	25%
Final	25%

Grading Scale:

Course final grades will be determined using the following scale. If the grade distribution is lower than expected the scale may be shifted downward but will never be shifted upward.

A	94-100	B+	87-89	C+	77-79	D+	67-69	F	59 and below
A-	90-93	B	83-86	C	73-76	D	63-66		
		B-	80-82	C-	70-72	D-	60-62		

Assignment Rubrics

Most labs will contain a point rubric at the end of the document. Ensure you have met those requirements before submission.

Grading Timeline

Labs and the project will be graded on the Vocareum website (http://www.vocareum.com) with feedback and comments annotated inline with your code submission. The grade and feedback will be usually be posted within 1-2 weeks of submission.

Course Schedule: A Weekly Breakdown

Below is a detailed course calendar —readings, assignments, examinations, etc., broken down on a weekly basis. **For each unit of in-class contact time, the university expects two hours of out of class student work per week over a semester.**

	Topics/Daily Activities	Readings and Homework	Lab to be Introduced
Week 1	Overview of IoT, Networks, and Embedded OSs; ***Python and Git Video Intro (Video Lecture)***	FSWN: Ch. 1-2 (Overview, Examples) IoT&EC: Ch. 1 Practical Python Programming	Lab 1 – Raspberry Pi
Week 2	(OSI) Layered Model of Networks Application layer & Transport Layers	IoT&EC: Ch. 10	Lab 2 – GrovePi
Week 3	**Holiday: Labor Day (9/6)** Pub/Sub Middleware (MQTT)	DS: 193-203 (Sockets)	Lab 3 – TCP/UDP
Week 4	MAC Layer (802.11, 802.15)	FWSN: Ch. 5 IoT&EC: Ch. 5-7	Lab 4 – MQTT
Week 5	More MAC + PHY Layer	FWSN: Ch. 5 IoT&EC: Ch. 5-7	Lab 5 – REST-ful APIs
Week 6	PHY Layer	FWSN: Ch. 6 IoT&EC: Ch. 4	Lab 6 – Web Services
Week 7	ADC	FWSN: Ch 3.1 IoT&EC: Ch. 3	Lab 7 – ADCs and DACs
Week 8	Signal Processing and Filtering	Class Notes	No Lab – Study for Midterm
Week 9	Security **Midterm: Wed. Oct. 20th in class**	DS: 501-529 (Security) FWSN: Ch. 11 IoT&EC: Ch. 13	Lab 8 – Signal Processing
Week 10	More Security Cloud & Distributed Computing	Class Notes IoT&EC: Ch. 8,11	Lab 9 – Instrumentation and Dashboards
Week 11	Cloud and ML	Class notes IoT&EC: Ch. 12	Lab 10 – ML and containers
Week 12	Energy Efficiency and Sleep Scheduling	FWSN: Ch. 8, 9 DS: 76-101 (System Arch)	
Week 13	Embedded Operating Systems Real-time OS and Threads	FWSN: Ch. 3.2, Ch. 4 DS: 104-116 (Threading) IoT&EC: Ch. 8	Lab Project
Week 14	Synchronization & IPC	Class Notes	Lab 11 – RIOT OS Overview and Threading
Week 15	More Threading/Synchronization Review	Class Notes	
FINAL	**Final on Mon. Dec. 13rd at 8 a.m.**		Date: For the date and time of the final for this class, consult the USC *Schedule of Classes* at classes.usc.edu.

Statement on Academic Conduct and Support Systems

Academic Conduct:

Plagiarism – presenting someone else's ideas as your own, either verbatim or recast in your own words – is a serious academic offense with serious consequences. Please familiarize yourself with the discussion of plagiarism in *SCampus* in Part B, Section 11, "Behavior Violating University Standards" policy.usc.edu/scampus-part-b. Other forms of academic dishonesty are equally unacceptable. See additional information in *SCampus* and university policies on scientific misconduct, http://policy.usc.edu/scientific-misconduct.

Support Systems:

Student Counseling Services (SCS) – (213) 740-7711 – 24/7 on call
Free and confidential mental health treatment for students, including short-term psychotherapy, group counseling, stress fitness workshops, and crisis intervention. engemannshc.usc.edu/counseling

National Suicide Prevention Lifeline – 1 (800) 273-8255
Provides free and confidential emotional support to people in suicidal crisis or emotional distress 24 hours a day, 7 days a week. www.suicidepreventionlifeline.org

Relationship and Sexual Violence Prevention Services (RSVP) – (213) 740-4900 – 24/7 on call
Free and confidential therapy services, workshops, and training for situations related to gender-based harm. engemannshc.usc.edu/rsvp

Sexual Assault Resource Center
For more information about how to get help or help a survivor, rights, reporting options, and additional resources, visit the website: sarc.usc.edu

Office of Equity and Diversity (OED)/Title IX Compliance – (213) 740-5086
Works with faculty, staff, visitors, applicants, and students around issues of protected class. equity.usc.edu

Bias Assessment Response and Support
Incidents of bias, hate crimes and microaggressions need to be reported allowing for appropriate investigation and response. studentaffairs.usc.edu/bias-assessment-response-support

The Office of Disability Services and Programs
Provides certification for students with disabilities and helps arrange relevant accommodations. dsp.usc.edu

Student Support and Advocacy – (213) 821-4710
Assists students and families in resolving complex issues adversely affecting their success as a student EX: personal, financial, and academic. studentaffairs.usc.edu/ssa

Diversity at USC
Information on events, programs and training, the Diversity Task Force (including representatives for each school), chronology, participation, and various resources for students. diversity.usc.edu

USC Emergency Information
Provides safety and other updates, including ways in which instruction will be continued if an officially declared emergency makes travel to campus infeasible. emergency.usc.edu

USC Department of Public Safety – UPC: (213) 740-4321 – HSC: (323) 442-1000 – 24-hour emergency or to report a crime.
Provides overall safety to USC community. dps.usc.edu