San José State University
Computer Engineering Department

CMPE 244: Embedded Software

Section TBD, Fall Semester, 2021

Course and Contact Information

Instructor:	Harry Li, Ph.D. Professor, Computer Engineering Department, San Jose State University
Office Location:	Engineering Building 267A
Telephone:	(408) 924-4060
Email:	hua.li@sjsu.edu
Office Hours:	Mondays and Wednesdays, 3:40 pm – 4:40 pm (by appointment)
Class Days/Time:	September 29 – December 15 Wednesdays from 4:00 pm -8:00 pm
Classroom:	On-line Zoom Based
Prerequisites:	CMPE 200, 240, 264 with passing grades

Course Description

OS kernel-level software design for embedded systems using Linux-based ARM SoC platform with custom IPs.

Course Format

The class uses the online conferencing system *Zoom* for lecturing, and the online learning management system *Canvas* for disseminating the instructor's announcements, teaching materials, assignments, grading results, as well as collecting students work and conducting exams/presentations. Students must have Internet connectivity with *Zoom* and *Canvas* accessibility. Students should also pay attention to the course instructor's email messages sent to the class through *MySJSU*.

Course Learning Outcomes (CLOs)

Upon successful completion of this course, students will be able to:

1. Understand how an integrated IP design works in an embedded Linux system.
2. Design and debugging an industry grade Linux Kernel Driver.
3. Design an application program to talk to the driver for verification.

Required Text & Readings

Course Materials

Instructor's teaching materials and online resources.
1. Professor's git: https://github.com/hualili/CMPE244
2. Jetson NANO Jetpack download https://developer.nvidia.com/embedded/downloads

Other Equipment / Material

1. Hardware Equipment: You may choose any one of the following options. For detailed selection information, I will cover it in the introduction session of the class. Option 1. Nvidia Jetson NANO Board with minimum 2 GB RAM; or Option 2. Pie 3B+, or Pie 4; Option 3: Nvidia Jetson Tx2 developer kit; or Option 4: LPC1769 CPU Module: https://www.mouser.com/NXP-Semiconductors/Embedded-Solutions/Engineering-Tools/Embedded-Processor-Development-Kits/Development-Boards-Kits-ARM/_/N-cxd2t?P=1z0jm4m&Keyword=LPC1769&FS=True&gclid=Cj0KCQjwqKuKBhCxARIsACf4XuHyN8WfqtQ24WGgtoMdKd6n-kI7c-YNz-r1hTcPt0ErdZN62jrMQmgaAtXZEALw_wcB or Option 5: Samsung ARM11 developer platform.
2. Linux Host Machine (Ubuntu 18.04) .

Course Requirements and Assignments

The course is built upon previous courses in this cohort program; a thorough review of CMPE200, 240, 264 materials is a must. Course works include weekly assignments, two quizzes, and a course project. Assignments, in forms of readings, problem solving, and hands-on design/implementation/verification, are aligned with topics discussed in class. Each assignment should be submitted to *Canvas* by the specified deadline. In general, success in a course is based on the expectation that students will spend, for each unit of credit, a minimum of 45 hours over the length of the course (normally three hours per unit per week) for instruction, preparation/studying, or course related activities, including but not limited to internships, labs, and clinical practice. This course, specifically, expects minimum of 6 hours per week to be spent outside of class.

Examination or Evaluation

Students will be evaluated based on their performance on projects/assignments and final examination, which is a culminating activity requiring creative work based on knowledge and techniques learned in this class.

Grading Policy

Grading is based on the following: 1) assignments and projects, 2) midterm and final exam. The distribution is shown below:

Assignments and projects:	30%
Midterm Exam:	30%
Final Exam:	40%
Total:	100%

Grade Determination

Final grades will be assigned based on the scales shown below:

Grade	Percentage
A plus	98 to 100%
A	92 to 97%
A minus	90 to 91%
B plus	87 to 89 %
B	81 to 86%
B minus	79 to 80%
C plus	76 to 78%
C	69 to 75%
C minus	66 to 68%
D	60 to 65%
F	Less than 60%

Classroom Protocol

1. Attend lectures.
2. Submit assignments and project reports on time.
3. Participate Demo, Show-and-tell, presentation in class.

University Policies

Per University Policy S16-9 *(http://www.sjsu.edu/senate/docs/S16-9.pdf)*, relevant university policy concerning all courses, such as student responsibilities, academic integrity, accommodations, dropping and adding, consent for recording of class, etc. and available student services (e.g. learning assistance, counseling, and other resources) are listed on Syllabus Information web page (http://www.sjsu.edu/gup/syllabusinfo), which is hosted by the Office of Undergraduate Education. Make sure to visit this page to review and be aware of these university policies and resources.

CMPE 244 Fall 2021 Course Schedule

The schedule (and related contents/assignments) may subject to change with fair notice. Any changes will be announced in due time in class and on the course's Canvas site.

Course Schedule

Week	Date	Topics
1	09/29	• Course orientation • Introduction to Embedded Software, (1) Selection of Target Development Platform Criteria, (2) Embedded Software Core Concepts: (2.1) CPU architecture; (2.2) memory map; (2.3) special purpose registers; *Assignment 1 (development platform preparation)*
2	10/06	• Memory map, peripheral controllers,; • CPU datasheet for initialization and configuration • Firmware and Embedded Software development environment • Kernel development environment, IDE, jetpack • Introduction to Linux Kernel Space and User Space programming *Assignment 2 ("hello, the world" code on user space, and "hello, the world" code in kernel space.)*
3	10/13	• Linux Kernel Architecture • Development environment IDE • embedded Linux kernel source distribution • menuconfig and scripting *Assignment 3 (GPP driver development for target platform)*
4	10/20	• Embedded Linux kernel Architecture • Development of customer defined driver
5	10/27	• Continuation of Linux Kernel Development • Development of PWM driver for the target platform • Adding computer vision capability to embedded software environment with OpenCV interface to GPP and/or PWM driver(s) *Assignment 4 (PWM driver and LED/Buzzer Integration on the embedded target platform)*
6	11/03	• Continuation on Linux Kernel Development • Development of drivers for Industrial IoT applications by establish sensor interface • Course project 1, Preparation and discussion for stepper motor controller design
7	11/10	• Continuation on Linux Kernel Development • Adding the capability to drive NIMA stepper motor and to be able to have sensor dual sensor interface to form a feedback control loop
8	11/17	• User space programming and kernel space driver development for optic sensor interface to capture the state information of stepper motor • Programming in kernel space for Acceleration Sensor interface
9	12/01	• Adding the graphics to embedded software environment • Customer Graphics library development • OpenGL
10	12/08	• Prepare and presentation of semester long project
11	12/15	Final Exam