

## Embedded Systems Hardware Architectures, Introduction

This course covers the hardware components and interfaces in a typical embedded system. The course begins with an inside look at some typical embedded systems and the functional blocks within those systems. The course addresses design considerations and several approaches to system building. Also covered are the various types of memory commonly used in embedded systems, basic concepts in microprocessors, microcontrollers and DSP, and an introduction to the typical buses used at the system level.

Course 21319

## IO Concepts and Protocols: PCI Express, Ethernet, and Fibre Channel

This course focuses on IO technologies and walks students through the complexities of IO subsystems in modern computers, and the networking and storage subsystems to which they are attached. Topics include basic concepts of IO; PCI Express, Ethernet and Fibre Channel; operation and protocols; and an exploration of how these technologies work.

Course 22177

## Jitter Essentials

Learn the definitions of various types of jitter (including phase noise), understand which type of jitter is important to your application and why, and learn how to propagate jitter through a system, create jitter budgets, measure and minimize jitter, and more.

Course 21321

## Linux Device Drivers

This course covers the design and implementation of device drivers for the Linux operating system, and the interfaces to the Linux kernel for writing them. Topics include compiling the kernel; building and running; character and block device driver concepts; PCI, I/O, and timer routines; hardware management and interrupt handling; networking drivers; PROC file system and IOCTL interface; and kernel mechanisms and advanced topics.

Course 2470

## Linux Device Drivers, Advanced

This course will enhance your understanding of the art of writing Linux device drivers. Students explore the framework that can be used to develop a commercial grade driver. The course includes detailed discussion of the USB and PCI family (PCI-E, PCIe) subsystems, as well as Linux kernel services and facilities. The course includes a board project, and code review of real-world drivers. Additional discussions cover live debugging with Kprobes, profiling with Oprofile, and a survey of industry trends including virtualized drivers, iSCSI and serial IO buses.

Course 1016

## Embedded Linux Design and Programming, Introduction

This course covers designing, testing, and customizing embedded Linux 2.6 on an ARM 9 processor platform. Topics include how the scheduler is implemented, writing for kernel modules and remotely debugging the embedded applications.

Course 3364

## Multicore Programming

This course builds a basic understanding of the multicore architecture and operating system factors when designing multi-threaded applications. You will gain the practical knowledge of Intel software tools to write applications, debug, analyze and profile them on multicore platforms.

Course 22180

## Printed Circuit Board Design for Signal Integrity and EMC Compliance

This course for both experienced and entry level engineers presents simplified hands-on techniques for the design and layout of printed circuit boards to achieve both signal integrity and electromagnetic compatibility (EMC). Instructional emphasis is placed on real-life examples that demonstrate good layout practices that can be incorporated immediately for high-technology designs and products.

Course 21943

## Real-Time Embedded Systems Programming, Introduction

This introductory course begins with a review of embedded system hardware, including a discussion of system requirements for real time systems. It follows up with real-time programming techniques and architectures. The course covers the use of real-time operating systems (RTOS) to effectively structure large programs as well as designing device drivers to meet real-time hardware requirements. There will also be discussion of multi-threading, cooperative vs. preemptive multi-tasking and inter-process communications. Students will gain hands-on experience with a programming project on a board.

Course 5381

## Smart Grid, Introduction

To address shortcomings in the traditional power grid, the smart grid brings together information technology, communications and control technology and power system engineering. This course introduces the building blocks of an end-to-end smart grid system. Topics covered include advanced metering infrastructure, smart meter technology and Home Energy Management Systems. The instructor presents a complete and up-to-date review of smart grid architecture, promising applications, the microgrid and energy storage technologies. Students learn the important standards being developed for smart grid and the Smart Grid innovation zones.

Course 22867

## Software Testing: Techniques, Tools and Practices

This course covers the fundamentals of software testing with an emphasis on test techniques, test tools, and testing practices. For the various phases of software development, the course introduces testing strategies, or test levels. Open source and commercially available tools will be used to demonstrate concepts such as test generation and test coverage. Students gain hands-on testing and analysis experience with sample code using state-of-the-art software analysis and testing tools.

Course 20501

## SystemVerilog for ASIC and FPGA Design

This course prepares hardware engineers, ASIC and FPGA designers, and design-support staff to use the high-level syntax of SystemVerilog to design, debug, and synthesize digital logic for ASICs, FPGAs, and IP cores. Students will learn SystemVerilog's basic building blocks and language constructs, including synthesizable data types and operators, structures and unions, 2-D arrays and loops, and the bus interface unit. In lab sessions, they will write code and synthesize it into digital logic and bus fabric, using both ASIC and FPGA tools.

Course 20095

## USB Device Interface: Architecture, Protocols and Programming

This course covers USB architecture, protocols and features. Communication, control, and application layer protocols for generic USB devices are covered. You will learn how to pick the optimal USB protocols to meet performance and bandwidth requirements. The course introduces the basic USB programming model, USB libraries and the higher level USB software required to implement specific device functionality. The course emphasizes effective design and test techniques and the use of USB debugging tools. The course focuses on the device side USB interface, and includes a class project based on a PIC USB kit.

Course 2179

## Wireless Technologies for Embedded Systems: Bluetooth, WiFi and ZigBee

This course provides practical knowledge of embedded system programming for wireless technologies. Instruction includes UDP and TCP based communications and socket programming, which is the foundation of network programming. The course addresses tradeoff considerations of bandwidth, cost and power among Bluetooth, WiFi and ZigBee. For each technology, the discussion includes architecture, key protocol, programming API and debugging. In the hands-on portion of the course, students design, implement, and demonstrate a wireless-enabled application on a Rabbit development board.

Course 23093

### Program Updates and Enrollment Information

Visit [ucsc-extension.edu/engineering](http://ucsc-extension.edu/engineering), for the most up-to-date information about our courses and programs, including textbooks, instructors, schedules and locations.

Enroll online at [ucsc-extension.edu](http://ucsc-extension.edu).

Send questions to [program@ucsc-extension.edu](mailto:program@ucsc-extension.edu)