



UCIRVINE | EXTENSION

Engineering

Embedded Systems Engineering Certificate Program

Accelerate Your Career

extension.uci.edu/engineer



Improve Your Career Options with a Professional Certificate

Embedded Systems Engineering Certificate Program

Today's embedded systems development ranges from microprocessor-based control systems, to systems-on-chip (SoC) design, and device software development. A myriad of implementations can be found in consumer electronics, medical devices, and commercial and military applications.

This certificate program looks at embedded systems engineering as a synergistic function between hardware and software device development. The curriculum covers the latest embedded technologies and the essential concepts of embedded systems development, through a practical hands-on approach using electronic design automation (EDA) tools and design kits.

In today's competitive business environment,

leaders are appointed based on credentials and experience. To stay ahead of the competition, advance your career and increase your earning potential, enroll in one of University of California, Irvine Extension's professional certificate programs. Convenient and affordable, UC Irvine Extension makes it easy to learn on your own time, in your own way. Courses are designed to ensure you gain mastery of a particular topic, and instructors are highly qualified leaders in their professions.

UC Irvine Extension is the only continuing education provider in Orange County that represents the University of California. A certificate bearing the UC seal signifies a well-known, uncompromising standard of academic excellence.

Program Benefits

- Gain essential knowledge of embedded systems design and programming.
- Learn how to program an embedded device.
- Become proficient in programmable logic design and analysis.
- Increase your understanding of real-time operating systems.
- Learn about the latest embedded technologies.
- Utilize EDA tools to optimize embedded systems designs.

Curriculum

Who Should Enroll

This program will benefit hardware/software engineers, computer engineers, communications and networking engineers, control systems engineers and other technical professionals involved in embedded systems design and development.

Certificate Prerequisites

Candidates must complete EECS 805, C Programming for Embedded Systems; and EECS X497.2, Introduction to Digital Logic and Hardware Architecture; or possess equivalent experience or education.

Certificate Requirements

A certificate is awarded upon completion of 15 credit units (9 required and 6 elective credit units), with a grade point average of “B” or better.

Transfer Credit

This program qualifies for degree credit at our partner universities. Visit extension.uci.edu/partners for information.

Certificate Candidacy

Candidates must have working experience or education in engineering or computer science, and demonstrate proficiency in C programming.

All requirements must be completed within five (5) years after the student enrolls in his/her first course.

Corporate Training

Bring this program to your workplace. Through Corporate Training, we can deliver this program or customize one that fits your company’s specific needs. Visit extension.uci.edu/corporate or call (949) 824-1847 for information.

For more information:

Jennifer Spitzer
(949) 824-9722
jspitzer@uci.edu
extension.uci.edu/engineer

Prerequisite Courses

C Programming for Embedded Systems

EECS 805 (1.5 CEU)

Embedded software can be found in many electronic devices today. Increase your understanding of the essential embedded language features required for embedded systems programming. Embedded software developers benefit from this hands-on course by expanding their knowledge of using pointers and arrays, bit manipulation, using key words such as “volatile” and “register,” and learning more about source code solutions to common embedded software problems.

Introduction to Digital Logic and Hardware Architecture

EECS X497.2 (3 units)

Further your understanding on fundamentals of logic design, boolean algebra and essential Verilog and VHDL statements describing behavioral functions such as counters and other finite state machines. Learn about the ASIC Design Flow from examples of logic and circuit design analysis, computer abstractions, and performance metrics. Participants are provided an overview of typical microprocessor architectures, hardwired versus micro-programmed control unit design, instruction set, addressing, I/O bus interface, hardware-software interfaces, and memory organization.

Required Courses (9 units)

Fundamentals of Embedded Systems Design and Programming

EECS X497.32 (3 units)

Gain an overview of embedded systems applications and design procedures, and learn how to plan and execute complete embedded systems designs that are cost-effective and competitive. You’ll gain the knowledge needed to determine and document system requirements for new designs as well as for improving existing systems. You’ll learn analysis techniques for optimizing system specifications as well as selecting microcontrollers for specific designs. Hands-on development is facilitated with an embedded system development kit.



Embedded Systems Architecture

EECS X497.36 (3 units)

Learn about the architecture of embedded systems and explore the difference between embedded design and traditional electronic device design. The special demands on embedded systems including real-time programming, portability, low power usage, and miniaturization dictate a different approach. The course introduces models and architectures, and covers such topics as specification, system partitioning, design quality, and developing synthesizable models.

Real-Time Embedded Systems Programming

EECS X497.34 (3 units)

Increase your understanding of the limitations and risks associated with embedded systems, and the methods and tools used to implement a successful design. Participants learn about the software process, with an emphasis given to the requirements definition, design and implementation phases, and limitations imposed by hardware design and real-time issues. Additional topics include: software architecture issues; development and debugging tools; advantages of languages commonly used in embedded systems; and verification methods. Hands-on development is facilitated with an embedded system development kit.

Elective Courses (Choose 6 units)

Logic Design and Analysis using Verilog

EECS X494.92 (3 units)

Expand your knowledge of gate level modeling, data flow modeling, behavior modeling, advanced modeling techniques, test benches, and logic synthesis. Learn the essentials of the Verilog hardware description language, syntax, and practical design scenarios. Participants learn fundamental and advanced usage of Verilog as a design capture and simulation development tool, and the use of the Programming Language Interface (PLI). The course will emphasize how Verilog is used in each step of the design automation process.

VHDL Design and Modeling of Digital Systems

EECS X494.94 (3 units)

Familiarize yourself with the analysis and synthesis of digital systems using VHDL to simulate and realize VLSI systems. Participants learn the fundamental concepts of VHDL and practical design techniques. The VHDL methodology and design flow for logic synthesis addresses design issues related to component modeling, data flow description in VHDL and behavioral description of hardware. An emphasis is placed on understanding the hard-

ware description language, VHDL design techniques for logic synthesis, design criteria, and VHDL applications.

FPGA Design with Hardware Description Languages

EECS X494.95 (3 units)

Gain a comprehensive understanding of Field Programmable Gate Arrays (FPGAs) architectures. Explore VHDL, Verilog, and variations of C as a hardware description language. Learn about design flow, simulation, and FPGA implementation. Engineers will enhance their knowledge of the CMOS process, trade-offs between FPGA's, metallized gate arrays, standard cells, and custom design. Gain insight into testability issues and boundary scan, termination, interfacing and timing issues, and methods of performance enhancement of a digital design.

Advanced VHDL Design and Modeling of Digital Systems

EECS X494.97 (3 units)

Expand your knowledge of advanced VHDL concepts. Participants will learn advanced concepts of VHDL and practical design techniques such as, file I/O, design reuse and verification, and use of industry standard libraries. The essence of VHDL methodology and design flow for logic synthesis is elevated to address design issues related to component modeling, data flow description in VHDL, and behavioral description of hardware and complex systems. An emphasis is placed on developing an expertise of the hardware description language, implementing advanced VHDL design techniques for logic synthesis, analyzing design criteria, performing verification, employing reuse, and evaluating VHDL applications.

Integrating Embedded Systems Successfully

EECS X497.24 (1.5 units)

This course takes participants through the hardware and software development integration process. Participants learn how, when, where and why using start integration including parallel development and integration techniques including co-simulation. Topics include: Parallel development and leveraging co-simulation to convert the engineering process from being serial to parallel, how and when to prototype as well as lab debug techniques.

Writing Portable Device Drivers

EECS X497.19 (3 units)

Become proficient at writing portable device driver source code and gain a competitive edge. Participants will gain practical knowledge of what constitutes a device driver, how to build one from a hardware datasheet, and how to write the code that will be readily portable across multiple platforms and operation systems. Increase your knowledge of timing, interrupt handling, direct memory access



(DMA), how to avoid pitfalls, and other critical issues fundamental to writing device drivers. Hands-on lab exercises reinforce code writing skills.

Designing Embedded Software Using Real-time Operating Systems

EECS X497.31 (3 units)

Learn how to write real-time systems software in relation to the architectural design of a complete embedded system utilizing a real-time operating system kernel. Participants will gain a practical knowledge of how to use a real-time kernel to accomplish the design goals of a real-time system. Learn how a real-time kernel is used to satisfy hard real-time constraints in comparison to soft real-time constraints. Gain a greater insight into the concepts of task scheduling, resource management, inter-task communications, task synchronization, and interrupt handlers.

Fundamentals of Embedded Linux

EECS X497.10 (3 units)

Further your understanding of Linux and its adoption as an embedded OS platform. This course provides an overview of methods and techniques to design and create embedded systems based on the Linux kernel. The essentials of the Linux operating system are discussed from the embedded system point of view including selecting, configuring, cross-compiling, installing a target-specific kernel; licenses; drivers and subsystems; the GNU development toolchain; and tools used to build embedded Linux systems. *Prerequisite: Familiarity with software programming and hardware design.*

Linux Driver Primer

EECS X497.11 (1.5 units)

Gain a competitive edge by learning how to develop and write code for Linux device drivers. Participants will gain practical knowledge of what constitutes a device driver in Linux and basic Linux device driver building blocks. In addition, learn how to build and grow a framework from scratch that can be used to develop a Linux device driver. Increase your knowledge of timing, interrupt handling, direct memory access (DMA), how to avoid pitfalls, and other critical issues fundamental to writing Linux device drivers. Hands-on lab exercises reinforce code writing skills.

Embedded Systems Design Using ARM Technology

EECS X497.39 (3 units)

Increase your understanding of how system-on-chip (SoC) and microprocessors are designed and used in embedded systems development. Participants will learn about the 16/32-bit embedded RISC processor ARM architecture and discover its wide applicability in embedded applications. Concepts and methodologies employed in designing a SoC based around a microprocessor core are thoroughly discussed. Practical hands-on lab exercises based on the ARM instruction set are used to reinforce the concepts learned. Architectural support for high-level languages, systems development, operating systems, and survey of ARM processor cores are discussed. A commercial ARM evaluation kit is used to demonstrate cost sensitive embedded applications.

Applied Control Theory for Embedded Systems

EECS X497.4 (3 units)

Learn how to apply modern control theory to optimize your embedded system designs using microcontrollers or DSP devices. The majority of embedded designs are closed loop control systems, as opposed to open loop control. This course provides practical how-to knowledge in deriving and applying practical control theory algorithms. Z Transforms are introduced as a practical way of developing the needed difference equations for optimal designs. Participants learn to evaluate and select the best control algorithm for desired control applications such as proportional-integral-derivative (PID), fuzzy logic or Z Transform-derived difference equations.

Real-Time Embedded Digital Signal Processing

EECS X498.61 (3 units)

Increase your level of expertise in embedded digital signal processing as well as DSP programming techniques. Participants learn about several important algorithms, such as Digital FIR, IIR filters, FFTs and advanced digital signal processing algorithms for embedded applications. An overview of fixed and floating point DSP processors is presented including memory use and management. Practical hands-on lab exercises using DSP assembly language programming is emphasized in conjunction with development tools that allow DSP code written in higher level languages, such as C and Matlab. A commercial DSP evaluation kit or a simulator is used to perform the labs.



Advisory Committee

- **Aaron Baranoff**, Co-founder, MataiTech, LLC
- **John Carey**, Director of Marketing, ATMEL Corporation
- **Bob Condie**, Senior Director, Engineering Firmware Development, Western Digital
- **Daniel D. Gajski**, The Henry Samueli Endowed Chair and Director, Center for Embedded Computer Systems, UC Irvine
- **Ian Harris**, Professor, Computer Science, UC Irvine
- **William Hohl**, University Relations Manager, ARM, Inc.
- **Hank Jacobs**, Software Engineer, General Monitors
- **John Koelsch**, Senior Architect, Safenet Government Solutions, LLC.
- **Farhad Mafie**, President and CEO, Savant Company Inc.
- **Rich Newman**, Test Management Product Specialist, Wind River Systems
- **Robert Weber**, Software Consultant, R&B Embedded Technologies

Embedded Systems Engineering Certificate Program



UCIRVINE | EXTENSION

extension.uci.edu/engineer ■ (949) 824-5414

