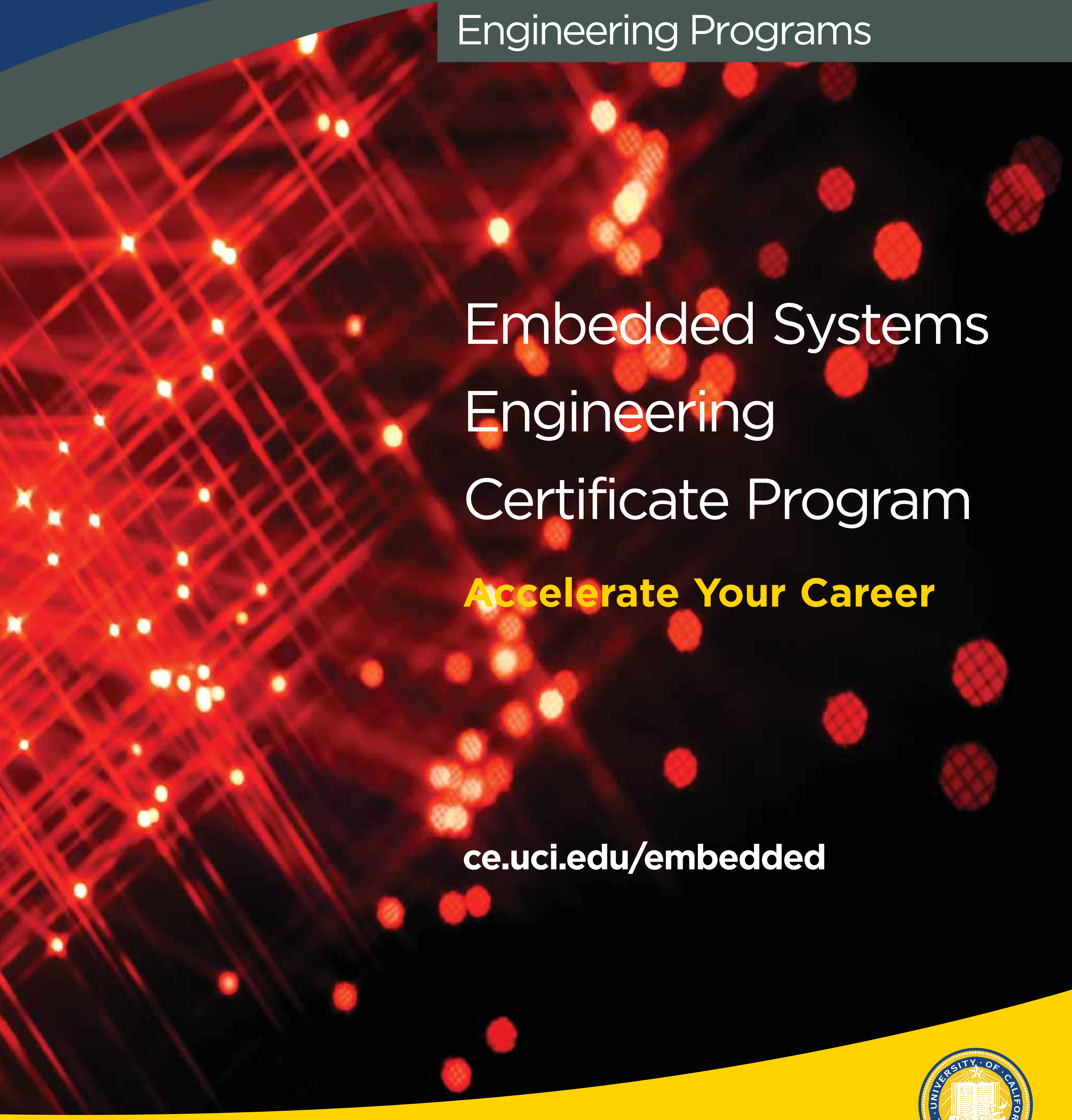
UCI Division of
Continuing Education

Engineering Programs



# Embedded Systems Engineering Certificate Program

**Accelerate Your Career**

**ce.uci.edu/embedded**

**University of California, Irvine**

# Improve Your Career Options with a Professional Certificate

**UCI Division of Continuing Education's professional certificate and specialized studies programs** help you increase or enhance your current skills or prepare for a new career. Courses are highly practical and instructors are qualified leaders in their field. Convenient online courses make it easy to learn on your own time, in your own way. A certificate bearing the UC seal signifies a well-known, uncompromising standard of excellence.

## Embedded Systems Engineering Certificate Program

Today's embedded systems development ranges from microprocessor-based control systems, to system-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.

## Who Should Enroll

This program is designed for working professionals who are interested in transitioning into the embedded systems/system-on-chip (SoC) industry, including hardware/software engineers, computer engineers, communications and networking engineers, control systems engineers and other technical professionals involved in embedded systems design and development.

## 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
- Explore the latest embedded technologies
- Utilize EDA tools to optimize embedded systems designs

# Curriculum

## Program Fees

The total cost of the program varies depending on prerequisite and elective courses chosen. Actual fees may differ from the estimate below. Fees are subject to change without prior notice.

| | |
|---|---|
| Course Fees (4.5 prerequisite, 9 required and 6 elective units) | $4,800 |
| Textbooks | $1,350 |
| Candidacy Fee | $125 |
| **Total Estimated Cost** | **$6,275** |

**IEEE members receive a 10% discount.**

## Certificate Eligibility and Requirements

The Embedded Systems Engineering Certificate Program is designed for individuals with working experience or education in engineering or computer science, and can demonstrate proficiency in C programming.

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 in engineering or computer science, and demonstrate proficiency in C programming.

A certificate is awarded upon completion of 15 credit units (9 required and 6 elective credit units), with a grade of “C” or better in each course.

To become an official candidate in the program, students pursuing the certificate must submit a **Declaration of Candidacy**. To receive the certificate after completing all program requirements, students must submit a **Request for Certificate**. All requirements must be completed within 5 years after the student enrolls in his/her first course. Students not pursuing the certificate program are welcome to take as many individual courses as they wish.

## Transfer Credit

Graduates from UCI Continuing Education’s Embedded Systems Engineering Certificate Program are eligible to transfer credits to:

- **University of Nebraska - Lincoln Master of Online Engineering** (MEM)
- **University of Wisconsin - Platteville Online Master of Science in Engineering**

NOTE: Any student wishing to transfer credits must obtain a “B” or better in each course.

## Prerequisite Courses

### C Programming for Embedded Systems

EECS 805 (1.5 CEU)

Embedded firmware can be found in many electronic devices today. Increase your understanding of the essential embedded language features required for embedded systems programming. Embedded firmware 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. We will also learn some best practices in firmware design, such as revision control, code review and debugging.

### 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 microprogrammed 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. Learn how to plan and execute complete embedded systems designs that are cost-effective and competitive. Gain the knowledge needed to determine and document system requirements for new designs as well as for improving existing systems. Acquire analysis techniques for optimizing system specifications as well as selecting microcontrollers for specific designs. Hands-on courseware is facilitated through the use of an embedded system development kit.

For more information:
Jennifer Mortensen
(949) 824-9722
j.mortensen@uci.edu

[*] Prerequisite: EECS 805, C Programming for Embedded Systems, or equivalent experience

[#] Course requires hardware or software, please refer to online listing for details.

### 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)

After taking course X497.32 and/or X497.36, or gaining equivalent knowledge from your job training, this course is designed to take you to the next level of real-time embedded system engineering, with a combination of lectures and associated labs. Lessons 1 – 5 cover embedded project development tools, programming of GPIOs, data structures, booting sequence, hardware interrupt, and ADC modules; labs 1 – 5 are non-RTOS (Real-time Operating System) based projects. Lessons 6 – 10 introduce various software architectures including RTOS, and build RTOS based projects. The labs are facilitated with a rich-featured lab kit from Texas Instruments, LAUNCHXL-F28069M, at $24.99, with a free full featured IDE tool – Code Composer Studio v6. Lectures in pdf and video format, and detailed lab instructions and demo videos are provided. At the end of the class, students will be able to apply the embedded development skills directly to their jobs, eliminating the gap between the academic education and real job skill requirements.

## Elective Courses (Minimum 6 units)

## Embedded Hardware Development

### 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. 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 hardware 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.

## Embedded Software Development

### C ++ for Embedded Systems

EECS X497.5 (3 units)

Embedded software can be found in many electronic devices today. Increase your understanding of the up and coming embedded language features required for embedded systems programming in C++. Embedded software developers will benefit from this hands-on course by expanding their knowledge of using C++ in an embedded system while avoiding common pitfalls.

### Motor Control Algorithms & Applications#

EECS X497.3 (3 units) – Part 1*

EECS X497.33 (3 units) – Part 2*

Although the topic of motor control has been around for decades, there are more development activities in motor drives and control technologies today than ever in the past. This is partially because of the increasing use and demanding requirements in applications ranging from domestic washing machines, HEV (Hybrid Electric Vehicles), aerospace flight controls, ultra-fast computer servos to adjustable-speed pumps, and many others. On the other hand, thanks to the rapid development of both power electronic switching devices and DSP/DSCs, motor drive technologies have been developed with great emphasis on performance, cost, efficiency, and controllability. From this effort the permanent magnet synchronous motor (PMSM) drive has emerged as a top competitor because of its high efficiency, low torque ripple, superior dynamic performance, and high power density. To realize such complex technologies, a group of experts including motor control experts and software engineers are indispensable. However, there is often a knowledge gap between these two disciplines. It is for the purpose of bridging this gap, that this course has been created. Starting from a review of motor types and motor control techniques, this course will focus on the theory and applications of Field Oriented Control (FOC) (also known as Space Vector Control) algorithms and embedded programming techniques. This course will demonstrate step-by-step how to create a DSP-based motor control project from scratch, write control blocks in C programming language, and finally,

complete a sensorless motor speed control project for a PMSM through eight incremental lab exercises. The course will be conducted in two sessions. Session one is the lecture, session two is the hands-on projects, eight week for each session. Each lecture is supplied with narrated PowerPoint presentation, and each lab is provided with recorded project demonstration.
*Students will register into two courses.

### Writing Portable Device Drivers*#

EECS X497.19 (3 units)

Gain a competitive edge on writing portable device drivers source code. 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 (RTOS)*#

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. Gain 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. Develop 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. Gain 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.

### Linux Driver Primer

EECS X497.12 (3 units)

Gain a competitive edge by learning how to write Linux Device Drivers. Participants will gain practical knowledge of what constitutes a device driver in Linux, basic Linux device driver building blocks, and how to build and grow a framework that can be used to develop a Linux device driver from scratch. Increase your knowledge of timing, interrupt handling, memory access, 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. 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 a 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)

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. Gain how-to knowledge in deriving and applying practical control theory algorithms. Z Transforms are introduced as a way of developing the needed difference equations for optimal designs. 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*#

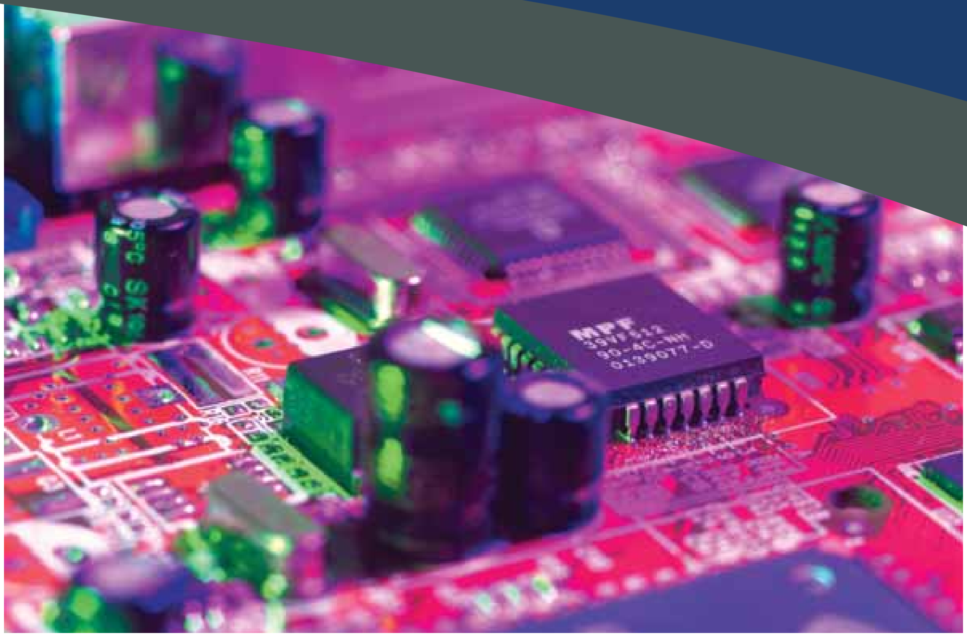
EECS X498.61 (3 units)

Advance your level of expertise in embedded digital signal processing as well as DSP programming techniques. Participants learn about adaptive filtering, signal generation and detection, echo cancellation, speech processing, audio processing and image processing for embedded applications. A mixture of C, Matlab and DSP assembly-language programming is employed to examine implementation and performance trade-offs. A commercial DSP development kit is used for hands-on learning.

### Corporate Training

Our Corporate Training specialists can deliver this program or customize one that fits your organization's specific needs. Visit ce.uci.edu/corporate or call (949) 824-1847 for information.

* Prerequisite: EECS 805, C Programming for Embedded Systems, or equivalent experience
# Course requires hardware or software, please refer to online listing for details.



## Advisory Committee

- **Aaron Baranoff,** Chief Software Architect, VubIQ Networks
- **Bob Condie,** Engineering Firmware Development Consultant
- **Daniel D. Gajski,** Professor Emeritus and Recall Professor, Electrical Engineering and Computer Science, University of California, Irvine
- **Ian Harris,** Professor, Computer Science, University of California, Irvine
- **Joseph Hwang,** Senior Engineering Manager, Panasonic Avionics Corporation
- **Henry Jacobs,** Senior Software Engineer, Applied Medical
- **John Koelsch,** Senior Architect, Safenet Government Solutions, LLC.
- **Farhad Mafie,** President and CEO, Savant Company Inc.
- **Rich Newman,** Sales Engineer, Synopsys Inc.
- **Robert Weber,** Software Consultant, R&B Embedded Technologies
- **Saleem Yamani,** Engineering Manager, Western Digital

## Academic Management

**Dave Dimas,** Ph.D., Director, Engineering, Sciences, and Information Technologies

# Embedded Systems Engineering Certificate Program

UCI Division of Continuing Education

Jennifer Mortensen ▪ (949) 824-9722 ▪ j.mortensen@uci.edu

09.20.16

ce.uci.edu/embedded