

Physical Computing with Embedded Systems (2025)

Careers and Professionalism

A Standard 1.1 - Develop essential professional skills for embedded systems engineering careers. I.A

- 1.1.1 Demonstrate effective communication skills in both technical and non-technical contexts. I.A.1
 - 1.1.2 Demonstrate integrity in embedded systems engineering practices. I.A.2
 - 1.1.3 Demonstrate collaboration and teamwork skills through embedded systems engineering projects. I.A.3
 - 1.1.4 Identify and develop traits important for success in embedded systems engineering. I.A.4
-

B Standard 1.2 - Investigate physical computing and embedded systems career paths. I.B

- 1.2.1 Research various roles within the embedded systems engineering field. I.B.1
 - 1.2.2 Identify professional certifications relevant to different embedded systems engineering careers. I.B.2
 - 1.2.3 Identify and evaluate routes to become an embedded systems engineer. I.B.3
-

C Standard 1.3 - Develop a professional embedded systems engineering portfolio. I.C

- 1.3.1 Investigate the purpose and importance of a portfolio in career advancement. I.C.1
 - 1.3.2 Analyze examples of professional portfolios to identify key components. I.C.2
 - 1.3.3 Assess platforms and tools for creating and hosting portfolios. I.C.3
 - 1.3.4 Select and organize personal projects and achievements for potential inclusion in a future portfolio. I.C.4
-

Core Embedded Systems Principles

A Standard 2.1 - Explain the core principles and components of embedded systems. II.A

- 1 2.1.1 Define embedded systems and distinguish them from general-purpose computing systems. II.A.1
 - 2 2.1.2 Analyze the key components of an embedded system. II.A.2
 - 3 2.1.3 Discuss the differences between microcontrollers and microprocessors. II.A.3
 - 4 2.1.4 Explain the concept of real-time systems and their importance in embedded applications. II.A.4
 - 5 2.1.5 Explain how embedded systems are utilized across various applications. II.A.5
-

B Standard 2.2 - Analyze and evaluate the architecture and capabilities of microcontrollers. II.B

- 1 2.2.1 Describe the basic architecture of a microcontroller. II.B.1
 - 2 2.2.2 Compare and contrast different types of microcontrollers and their suitable applications. II.B.2
 - 3 2.2.3 Compare and contrast the performance and efficiency characteristics of microcontrollers. II.B.3
-

C Standard 2.3 - Apply Real-Time Operating Systems (RTOS) concepts. II.C

- 1 2.3.1 Describe and analyze the fundamentals of Real-Time Operating Systems. II.C.1
 - 2 2.3.2 Compare RTOS to general-purpose operating systems. II.C.2
-

Sensors, Actuators, and Communication Protocols

A Standard 3.1 - Identify and implement various types of sensors and actuators. III.A

- 1 3.1.1 Classify different types of sensors and explain their operating principles. III.A.1
 - 2 3.1.2 Interface with sensors using communication protocols. III.A.2
 - 3 3.1.3 Gather and analyze data using sensors. III.A.3
 - 4 3.1.4 Compare and contrast different types of actuators. III.A.4
-

B Standard 3.2 - Implement user interfaces and display systems. III.B

- 1 3.2.1 Implement display interfaces to present information in embedded systems. III.B.1
- 2 3.2.2 Program simple user interfaces using various input methods. III.B.2

C Standard 3.3 - Implement communication protocols in embedded systems. III.C

- 1 3.3.1 Apply serial communication in embedded systems. III.C.1
 - 2 3.3.2 Use serial communication protocols for inter-device communication. III.C.2
 - 3 3.3.3 Implement wireless communication technologies in physical computing projects. III.C.3
-

Programming for Embedded Systems

A Standard 4.1 - Apply programming concepts to embedded systems development. IV.A

- 1 4.1.1 Implement embedded-specific data types and operations. IV.A.1
 - 2 4.1.2 Utilize memory-efficient data structures and algorithms optimized for limited resources. IV.A.2
 - 3 4.1.3 Develop state machines for managing complex system behaviors and transitions. IV.A.3
 - 4 4.1.4 Apply software design patterns appropriate for embedded systems. IV.A.4
-

B Standard 4.2 - Interact with hardware components using a text-based programming language. IV.B

- 1 4.2.1 Control digital input/output pins and interface with various components. IV.B.1
 - 2 4.2.2 Configure analog-to-digital conversion for sensor data collection. IV.B.2
 - 3 4.2.3 Control components using pulse-width modulation (PWM). IV.B.3
 - 4 4.2.4 Develop hardware abstraction layers for interfacing with microcontroller peripherals. IV.B.4
-

C Standard 4.3 - Implement interrupt-driven programming for embedded systems. IV.C

- 1 4.3.1 Describe the concept and importance of interrupts in embedded systems. IV.C.1
 - 2 4.3.2 Write interrupt service routines (ISRs) to handle hardware events. IV.C.2
 - 3 4.3.3 Develop programs using timer interrupts for periodic task execution. IV.C.3
 - 4 4.3.4 Apply event-based programming techniques for real-time responsiveness. IV.C.4
-

D Standard 4.4 - Implement debugging and troubleshooting techniques. IV.D

- 1 4.4.1 Utilize embedded-specific debugging tools and techniques. IV.D.1
 - 2 4.4.2 Apply systematic approaches to identify and resolve hardware and software issues. IV.D.2
 - 3 4.4.3 Use logging and tracing techniques for embedded applications. IV.D.3
-

Construct Simple Embedded Systems

A Standard 5.1 - Apply fundamental electrical concepts to embedded systems design. [V.A](#)

- 1 5.1.1 Apply basic electrical laws and principles. [V.A.1](#)
 - 2 5.1.2 Implement electrical circuits in embedded systems. [V.A.2](#)
 - 3 5.1.3 Conduct electrical testing and troubleshooting. [V.A.3](#)
-

B Standard 5.2 - Implement power management and efficiency techniques. [V.B](#)

- 1 5.2.1 Calculate power consumption in embedded systems. [V.B.1](#)
 - 2 5.2.2 Implement sleep modes and wake-up strategies to conserve power. [V.B.2](#)
 - 3 5.2.3 Design power-efficient embedded systems for long-term deployment. [V.B.3](#)
-

C Standard 5.3 - Integrate mechanical components with embedded systems for sensor-actuator applications. [V.C](#)

- 1 5.3.1 Apply mechanical principles to sensor selection and placement. [V.C.1](#)
 - 2 5.3.2 Implement actuator control systems. [V.C.2](#)
 - 3 5.3.3 Design and troubleshoot electromechanical interfaces. [V.C.3](#)
-

D Standard 5.4 - Create and manage comprehensive documentation. [V.D](#)

- 1 5.4.1 Create comprehensive documentation for embedded systems projects. [V.D.1](#)
 - 2 5.4.2 Implement version control practices for managing code and project files. [V.D.2](#)
-

E Standard 5.5 - Implement proper safety practices. [V.E](#)

- 1 5.5.1 Adhere to electrical safety standards and regulations. [V.E.1](#)
 - 2 5.5.2 Adhere to mechanical safety standards and regulations. [V.E.2](#)
-

Advanced Concepts, Emerging Technologies, and Ethical Considerations

A Standard 6.1 - Implement testing, validation, and security practices. [VI.A](#)

- 1 6.1.1 Develop unit tests and system-level testing for embedded systems. [VI.A.1](#)
 - 2 6.1.2 Implement basic security measures in embedded systems. [VI.A.2](#)
 - 3 6.1.3 Discuss techniques for securely updating firmware in deployed systems. [VI.A.3](#)
-

B Standard 6.2 - Research emerging technologies in embedded systems. [VI.B](#)

- 1 6.2.1 Investigate the implementation of basic machine learning algorithms on microcontrollers. [VI.B.1](#)
- 2 6.2.2 Explain the principles of edge computing and its applications in embedded systems. [VI.B.2](#)
- 3 6.2.3 Identify integrated embedded systems in emerging fields and technologies. [VI.B.3](#)

C Standard 6.3 - Consider ethical and environmental implications. VI.C

- 1 6.3.1 Discuss ethical considerations related to data collection, privacy, and security in IoT and embedded systems. VI.C.1
- 2 6.3.2 Assess the environmental impact of embedded systems and sustainable design practices. VI.C.2
- 3 6.3.3 Investigate the role of embedded systems in environmental monitoring and conservation efforts. VI.C.3