■ Information Technology Engineers Examination

Embedded Systems Specialist Examination

(Level 4)

Syllabus

— Details of Knowledge and Skills Required for the Information Technology Engineers Examination —

Version 3.1

IPA
INFORMATION-TECHNOLOGY PROMOTION AGENCY, JAPAN

Corporate names or product names used in this syllabus are trademarks or registered trademarks of each company or organization.
® and TM are not used in the syllabus.

Major category	Minor category	Outline		Required knowledge	Required skills
1 Understanding of the Functional Requirements for Embedded Systems	1-1 Understanding hardware and software requirements specifications according to system functional requirements	(1) Confirmation of requirements to implement the required functions (2) Analysis of reliability and safety requirements (3) Confirmation of requirements for interfacing with other systems	Confirm the requirements of the embedded system, examine the hardware and software requirements specifications in detail which were created based on system functional requirements, and determine the items to include in the hardware design document and software specifications.	• Functions, performance, and characteristics of each component device • Hardware design • Functional allocation • Selection of the optimal implementation means from plural options • Failure and exception handling • High-reliability system configuration method	• Appropriately allocating required functions according to the functions, performance, and characteristics of each component device • Appropriately allocating failure and exception handling • Appropriately determining the interface between component devices
	1-2 Investigation of associated technologies, intellectual property rights of other companies, and related regulations	(1) General technical investigation of associated technology fields (2) Trend investigation of technologies such as IoT, big data, AI, etc. (3) Trend investigation of hardware and software configuration methods and element technologies (4) Trend investigation of development technologies and environment, and tools (5) Investigation of the technologies owned by other companies (6) Consideration of the handling of technologies owned by other companies (7) Investigation of product liability issues (8) Investigation of safety standards	Investigate the associated technologies required to fulfill system functional requirements, the issues, and the technologies conforming to the safety standards to build the development environment. In addition, confirm whether the element technologies to implement the system conflict with the intellectual property rights of other companies or have product liability issues.	• Technical trends in the target area • Trend investigation of technologies such as IoT, big data, AI, etc. • Technical trends in hardware and software configuration methods and element technologies • Integrated development environments and development tools • Laws, regulations, and standards related to intellectual property rights including patent rights, utility model rights, and copyrights • Investigation methods • Latest trends in safety standards	• Determining the associated technology trends to investigate • Determining the related laws to investigate • Evaluating and identifying the importance of necessary technologies • Performing investigations quickly

Major category	Minor category	Outline		Required knowledge	Required skills
2 Design and Development of Embedded Systems	2-1 Determination of the interface between subsystems based on system architecture	(1) Determination of devices to interface with external devices (2) Determination of the communication method between devices (3) Determination of the internal configuration of devices (4) Determination of the interface between units (5) Confirmation of real-time processing (6) Determination of process distribution	Consider and determine the unit of the subsystem, device, and unit (i.e. the minimum unit of a system component that possesses an interface) according to the system architecture specifications. In addition, consider and determine the interface and communication methods between these units. Confirm the real-time processing required for the system.	• Validation of hardware functions, performance, and interface • Hardware restrictions • Hardware characteristics and specifications • Fault tolerance • Functional safety design • Real-time processing	• Selecting appropriate hardware from the system's perspective • Proposing methods to achieve necessary high-reliability and safety • Verifying validity of the interface between the subsystems
	2-2 Determination of function and performance allocation among hardware, software, and firmware	(1) Function allocation among hardware, software, and firmware (2) Determination of interface specifications among hardware, software, and firmware (3) Selection of the microprocessor and micro controller that satisfy hardware functional requirements, and evaluation of system LSI implementation	Determine the optimal allocation of functions which should be implemented by the hardware of each device and unit, functions which should be processed by the software, and functions which should be processed by the firmware. Consider the interfaces among the hardware, software, and firmware, and clarify their interface points. Define the control signals, data, and timing so that their interpretation is not ambiguous and required information is not missing. Consider not only functions, but also the performance required for the system in order to determine the optimal allocation of performance between the hardware and software.	• Hardware performance, functions, and interface • Hardware characteristics • Software characteristics • Usage of OS, languages, and commercially-available libraries • Using software to get better performance out of the hardware • Using hardware to get better performance out of the software • State-of-the-art hardware and trends • Cost • Proper evaluation of characteristics of plural configuration methods • Possibility of long-term supply of hardware components to be used	• Determining the appropriate allocation of functions between the hardware and software, according to the restrictions required for the system • Reexamining the allocation of functions between the hardware and software from the viewpoint of development efficiency and implementation space • Reexamining the allocation of performance between the hardware and software from the viewpoint of hardware characteristics

Major category	Minor category	Outline		Required knowledge	Required skills
	2-3 Verification of feasibility such as overall performance estimation and design review	(1) Feasibility verification by experiment (2) Determination of human interface by prototyping (3) Design review	Verify feasibility and human interface specifications through experiment or prototyping techniques as required. In addition, perform a design review to evaluate the feasibility from the perspective of both hardware and software designs.	• Specific implementation methods • Prototyping techniques • Simulation techniques • Design review techniques • Implementation items and points to keep in mind in system design	• Examining the validity of the functions allocated between the hardware and software, and the validity of the interface • Verifying the validity of the implementation method against system requirements • Coordinating matters under consideration from the perspective of the entire system • Pointing out issues logically • Presenting alternative plans
	2-4 Investigation of implementation functions in consideration of development period and cost	(1) Estimation of work items, workload, and period, as well as development resources and cost (2) Examination of the balance between development period, cost, and function volume	Estimate the work items, workload, period, development resources, and cost, while giving consideration to the hardware development. In addition, examine the balance between the development period, cost, and function volume (including the restrictions) to verify the validity of the estimation.	• Development productivity of hardware and software respectively • Development processes of hardware and software respectively • Co-design • Resource allocation • Cost • Introduction of (critical) technologies	• Calculating development person-hours according to development productivity • Coordinating the hardware and software development schedule • Calculating the appropriate development period by considering the number of resources and cost
	2-5 Work results management and determination of various recording methods	(1) Implementation of progress management and storage of progress performance records (2) Implementation of quality management and storage of quality records (3) Storage of access records to materials of other companies	Have an accurate understanding of the current position within the work plan, identify problems at an early stage, analyze them, and take countermeasures. Manage work so that they are executed in accordance with the development procedure. Also, store the various performance records.	• Data organization and analysis techniques • Intellectual property rights and trade secrets • Record management methods	• Performing project management • Systematically organizing and analyzing information

Major category	Minor category	Outline		Required knowledge	Required skills
	2-6 Process planning	(1) Preparation of the process plan (2) Preparation of the estimation (3) Preparation of the process management plan (4) Preparation of the training plan	Prepare a specific plan to develop the software, focusing on consistency with the hardware development plan. Also, document the completion criteria of each stage of development.	• Process planning procedures • Development procedures • Design and development techniques • Estimation techniques • Staff training • Project planning and management techniques • Cost • Process deliverables	• Collecting, organizing, and analyzing estimation-related information such as specification surveys and past data • Coordinating with other departments to secure resources • Evaluating personnel skills according to development techniques and environment, and training them • Calculating and evaluating costs • Determining the development processes and the development management framework for each process • Coordinating with the hardware development plan • Managing risks
	2-7 Formulation and design of policy for implementing quality requirements	(1) Formulation and design of policy for implementing quality requirements such as real-time property, functional safety, high reliability, environmental safety, security, and energy saving	Establish a required level for each of various quality requirements, and design a method for implementing the level.	• Real-time design, functional safety design, high reliability design, environmental safety design, security design, and energy-saving design • Non-functional requirements	• Understanding an appropriate system--required level for each of quality requirements, such as real-time design, functional safety design, high reliability design, environmental safety design, security design, and energy-saving design, and designing an implementation means
	2-8 Design of development environment	(1) Development tool environment preparation planning (2) Design of the development environment (3) Standardization planning	Prepare a specific plan to establish the development environment, based on the development status of the target hardware. Establish the software development environment. Prepare specific plans for the distributed development environment, work equipment, and testing equipment.	• Development equipment • Development environment and tools • Purchase, rental, and lease • Availability of development environments on the Internet • Development standards • Reuse	• Understanding the impact of installing the development environment • Arranging for the development equipment and tools through appropriate channels • Coordinating with the training department for the installed tools

Major category	Minor category	Outline		Required knowledge	Required skills
3 Design and Manufacturing of Embedded System Software	3-1 Creation of software specifications	(1) Determination of task configuration and priority of tasks to ensure real-time processing (2) Examination of inter-task communication methods and necessity of exclusive control, and determination of their implementation methods (3) Memory allocation and memory map etc. (4) Preparation of software specifications	Examine the software configurations that meet software requirements specifications, and furthermore examine the task configurations that ensure real-time processing. Determine the data communication and exclusive control methods between tasks, and document them as software specifications.	• Items which should be determined as software specifications • Real-time OS • Task design • Critical section and exclusive control • System calls used by the real-time OS • Hardware restrictions • Software design techniques • Task configuration, common data, OS resources, and class design concept • Object-oriented design and structured design • Software characteristics	• Systematically organizing and documenting the matters considered • Extracting real-time requirements from the target system requirements • Designing task configurations that satisfy functional requirements • Designing the optimum communication method between tasks • Evaluating the validity of preemption from the relationship between interrupts and real-time processing • Examining the existence of critical sections, and designing optimal exclusive control • Making use of necessary system calls • Arranging the requirements for hardware from the viewpoint of software • Considering measures in response to exceptions, errors, and hardware failures
	3-2 Design of device driver	(1) Design a device driver for dedicated hardware	Make a device driver for hardware as necessary when no device driver is available.	• Real-time OS • Device driver	• Understanding interfaces required for hardware operations • Designing device drivers
	3-3 Review at each stage of the software life cycle	(1) Review planning (2) Validity judgment of deliverables from each stage and issues pointed out (3) Response to issues pointed out in the review (4) Judgment of validity in moving to the next stage	Review the work at each stage in software development based on the performance records and deliverables to ensure that it satisfies the requirements. Evaluate the review results and the validity of the response to issues pointed out in the review, and judge whether or not the development project can proceed to the next stage.	• Software life cycle process • Implementation items and points to keep in mind in system design • Validity judgment based on software deliverables • Industry and products • Review techniques • Functional safety design	• Determining the required development level according to system functional requirements • Presenting alternative plans • Presenting an optimal plan from a comprehensive viewpoint • Evaluating the design quality • Evaluating the development process based on the progress results and quality records

Major category	Minor category	Outline		Required knowledge	Required skills
	3-4 Creation of the program and extraction of program test items	(1) Determination of the standard source program description format (2) Preparation of the program creation environment (3) Program creation (4) Selection of test methods (5) Extraction of program unit test items (6) Extraction of program test items (7) Determination of fault isolation procedures	Determine the standard description format for the source program in order to create programs. Also, select test methods which satisfy the quality level required for the program, and extract the test items to be executed.	• Programming techniques • Programming efficiency • Program quality • Program test techniques • Program test procedures • Program test environment	• Clarifying programming guidelines according to software specifications • Introducing measures to improve programming efficiency and ensure quality • Extracting and systematically organizing program test items • Determining the program test environment • Determining the program test order
	3-5 Construction of the test environment	(1) Preparation of test equipment (2) Securing the test location (3) Securing test staff	Confirm the number of debugging tools, test staff, and the location to deploy the system, and build the test environment.	• Integration test equipment • Integration test environment and tools • Purchase, rental, and lease • Connection of the integration test equipment • Integration test process	• Arranging for the test equipment and tools through appropriate channels • Cooperating with the test tool training department • Connecting the test equipment to build the test environment
	3-6 Test execution	(1) Construction and maintenance of the test environment (2) Test execution (3) Negotiation with other departments (4) Handling of unexpected events (5) Fault isolation (6) Confirmation test with the final version	Execute the test according to the test specifications. Plan and implement measures in response to failures that have occurred during the test.	• Points to confirm in the integration test • Test equipment • Test techniques • Test procedures • Test environment • Measures in response to failures • Quality evaluation and forecasting methods • Analysis of the situation in the event of failure • Utilization methods of quality data • Writing presentation skills	• Teaching test methods including test equipment usage • Analyzing the cause of a failure (hardware and software isolation) • Collecting and analyzing the situation in the event of failure • Reconsidering the test items in response to the situation of failures or the content of the failures • Managing the situation in the event of failure • Collaborating with the person in charge of the hardware • Coordinating with other departments • Analyzing the content and situation of failures, and evaluating quality • Analyzing the quality situation

Major category	Minor category	Outline		Required knowledge	Required skills
	3-7 Integration test of hardware and software (system verification test)	(1) Determination of the integration test commencement period (2) Preparation of a test location plan (3) Creation of a test preparation plan (4) Preparation of a personnel plan (5) Execution of the integration test	Define the conditions to execute the hardware and software integration test, and prepare a specific plan to execute the integration test. Also, clearly state the responsibilities assigned to each group. Execute the integration test. Plan (i.e. determine the date, reviewers, and location) and conduct a review of the deliverables and performance records from the hardware and software integration test. Take measures in response to issues pointed out in the review.	• Hardware and software development processes • Purpose and confirmation items of the integration test • Integration test methods, tools, and equipment • Productivity in test execution • Evaluation criteria • Review techniques	• Presenting policies for the test environment, including the purpose, method, and completion criteria of the test • Coordinating the process plan with hardware development • Extrapolating necessary equipment from the test plan, and making arrangements to procure them • Extracting items to be executed in the integration test, and organizing them systematically • Evaluating system quality • Evaluating the development process based on the progress results and quality records
	3-8 Configuration management	(1) Extraction of the configuration management target document (2) Preparation of rules for administering configuration management	Prepare specific plans and rules to implement configuration management.	• Purpose and method of configuration management and points to keep in mind • Configuration management tools	• Selecting the appropriate configuration management tools for the system
	3-9 Review of program creation and program test	(1) Review planning (2) Reviewing (3) Response to issues pointed out in the review (4) Completion criteria for program creation work (5) Completion criteria for program test work	Plan (i.e. determine the date, reviewers, and location) and conduct a review of the deliverables and performance records from the program creation and program test, and respond to issues pointed out in the review. In addition, judge whether program creation and program test are complete based on the review results.	• Execution items and points to keep in mind in program creation and program test • Development processes • Evaluation criteria • Review techniques	• Addressing issues logically • Presenting alternative plans • Presenting an optimal plan from a comprehensive viewpoint • Evaluating program quality • Evaluating the development process based on the progress results and quality records

Major category	Minor category	Outline		Required knowledge	Required skills
	3-10 Work results management and records	(1) Implementation of progress management and storage of progress performance records (2) Implementation of quality management and storage of quality records (3) Management of the program change history (4) Storage of records relating to the root cause of failures and measures	Have an accurate understanding of the current position within the work plan, identify problems at an early stage, analyze them, and take countermeasures. When doing so, keep records of specification changes, occurrences of failure, and measures, as well as the minutes of general progress meetings.	• Development management target data • Data organization and analysis techniques • Record management methods • Version control	• Performing project management • Systematically organizing and analyzing information • Understanding the software components and managing the history
4 Design and Manufacturing of Embedded System Hardware	4-1 Hardware design document	(1) Analysis of hardware requirements specifications (2) Determination of the hardware architecture (3) Design of the interfaces between subsystems and units (4) Determination of the I/O method (5) Safety and reliability design	Consider and determine the hardware architecture to satisfy the safety and reliability in accordance with the system functional requirements specifications. Examine the hardware requirements specifications and determine the appropriate hardware configuration. Determine the interface between subsystems and units. Determine the I/O methods for the system and the rated I/O performance. Organize them into a hardware design document.	• Reliability engineering • Fault tolerant systems • Fail-safe design • Multi-processor systems • Serial communications • Serial interface standards • I/O circuits • Timer functions • Analog signal processing	• Designing a system which satisfies reliability and safety requirements • Evaluating system reliability

Major category	Minor category	Outline		Required knowledge	Required skills
	4-2 Design of processing units	(1) Selection of the MPU, MCU, and DSP (2) Consideration of system LSI (3) Memory hierarchy (4) Examination of the internal bus structure (5) Use of peripheral LSI (6) Use of programmable LSI (7) Power-saving design (8) Thermal design	Select an MPU, MCU, and DSP which satisfies the performance requirements for the unit. Design the memory hierarchy in accordance with the cost and processing performance. Design an optimal internal bus considering data movement between internal structure elements. Judge whether system LSI is appropriate and whether a programmable LSI is available considering space-saving benefit and future product development. Make proper use of peripheral LSIs. Design the circuit considering power-saving and thermal design.	• MPU, MCU, DSP, and peripheral LSI • Audio and imaging devices • Codec • System LSI • Programmable LSI • Memory hierarchy • Speeding up of MPUs • High-level hardware description languages • Power-saving design • Heat radiation and thermal design	• Appropriately selecting and using MPU, MCU, DSP, and peripheral LSI • Making proper use of audio and imaging devices • Fully utilizing codec to improve quality and reduce costs • Judging whether to implement the circuit components as a system LSI or a programmable LSI • Understanding system LSI design using high-level hardware description languages, and judging feasibility • Designing a memory hierarchy to realize higher speed and lower cost • Understanding and using methods to make a faster MPU to improve processing efficiency

Major category	Minor category	Outline		Required knowledge	Required skills
	4-3 Design of interfaces to the software	(1) Interrupts (2) I/O table (3) Interface for the power-saving mechanism (4) Communication protocol (5) Human interface	Appropriately allocate interrupts required by the system according to priority level, and link them with the software interrupt processing program. Determine the I/O table specifications for interfacing with the system I/O and peripheral LSI. Design the interface requirements with the software according to the transmission specifications. Determine the specifications to utilize the power-saving mechanism. Select and use a communication protocol suitable for the requirements specifications, and evaluate the performance. Propose an appropriate human interface, and determine the software specifications.	• Interrupt control • I/O methods • Design of the communication protocol • Evaluation of the transmission rate • Power-saving technologies • Human interface	• Designing an interrupt control circuit • Selecting and designing an appropriate I/O method • Designing a communication protocol suitable for the required specifications, and evaluating the performance • Designing the system utilizing power-saving technologies • Proposing an appropriate human interface based on system functional specifications
	4-4 Design of the I/O interface	(1) A/D and D/A (2) USB (3) Wired/wireless communication interface (4) Flash memory (5) Auxiliary storage devices	Design a circuit to obtain physical quantities using an A/D or D/A conversion circuit and to drive actuators. Make proper use of various devices with an USB interface. Make effective use of wired/wireless communication interface. Make effective use of flash memory. Make effective use of auxiliary storage devices.	• A/D and D/A conversion circuits • Various devices using an USB interface • Wired/wireless communication interface • Flash memory • Auxiliary storage devices	• Designing a circuit to obtain physical quantities using an A/D or D/A conversion circuit and to drive actuators • Making proper use of wired/wireless communication interface • Making proper use of USB interface products and flash memory • Using auxiliary storage devices based on performance conditions

Major category	Minor category	Outline		Required knowledge	Required skills
	4-5 Manufacturing of hardware	(1) System verification test (2) Preparation of the development environment and test environment (3) EMC evaluation (4) Issues concerning electricity and machinery	Use test tools to test the manufactured hardware to check whether or not it satisfies the requirements specifications. Execute an integration test with the software to ensure that the product satisfies the functional requirements. Prepare the development and test environments used for manufacturing and testing. Confirm that the EMC-related standard is satisfied. Evaluate the tolerance toward disturbances in the electrical and mechanical environment such as voltage fluctuation and vibration, and take measures if necessary.	• Hardware unit test • Integration test with the software • Product development environment and test environment • EMC standard • EMC test • Disturbances around electrical and mechanical systems such as voltage fluctuation and vibration, and countermeasures • Occurrence mechanism of latch-up and its countermeasures	• Conducting the hardware unit test • Preparing and executing the integration test with the software, and judging the test results • Preparing the development and test environments used for manufacturing and testing • Executing and evaluating the EMC test • Executing the environmental disturbance test around electrical and mechanical systems and taking countermeasures
5 System Evaluation	5-1 Preparation of an inspection plan	(1) Preparation of an inspection plan	Create a specific plan to prepare for the inspection (arrangement of deliverables and management results).	• Purpose of inspection • Inspection procedures • Deliverables and management work results necessary for inspection • Estimation techniques	• Extracting and systematically organizing the work items necessary for inspection, based on the organization guidelines for the deliverables and management work results • Determining the division of roles and work procedures • Calculating a valid estimate according to the work details, period, cost, and resources • Coordinating with other departments over the timing of inspections

Major category	Minor category	Outline		Required knowledge	Required skills
	5-2 System evaluation	(1) Analysis of the inspection results and countermeasures (2) Determination of system evaluation items and scale (3) Performing system evaluation	Evaluate the developed system to confirm that it provides functions initially defined in the specifications, and evaluate the manuals supplied with the system, the achievement of technical goals, intellectual property rights, and quality objectives.	• Purpose of system evaluation • System evaluation framework and items • System evaluation criteria and evaluation methods • Problem analysis and solution techniques • System industry and products • Quality management and evaluation • Record management methods	• Evaluating consistency between the system functional requirements and customer demands • Managing the evaluation results as records
6 Maintenance	6-1 Preparation of a maintenance plan	(1) Preparation of a maintenance plan	Based on the development plan, prepare a specific plan to perform maintenance. Prepare a specific plan by extracting and considering the work items and work details according to the maintenance details (functional changes and responding to complaints).	• Purpose of maintenance • System industry • Types of maintenance work • Response methods according to the purpose of maintenance • Product lifecycle	• Extracting specific maintenance items according to the definition of maintenance work (the corresponding work), and organizing them systematically • Clarifying the scope of impact caused by the maintenance work • Determining the division of roles, work procedures, and maintenance framework • Calculating a valid estimate according to the work details, period, cost, and resources
	6-2 Organization of development information for maintenance	(1) Organization of specification documents (2) Organization of design documents (3) Storage of the program (4) Storage of development records (5) Organization of materials for the operation of the development environment (6) Storage of revision records	Organize and store documents necessary for maintenance.	• Document management method and procedures • Change management	• Systematically organizing and managing information for maintenance • Clarifying procedures to change maintenance information

Major category	Minor category	Outline		Required knowledge	Required skills
	6-3 Preparation of the maintenance environment	(1) Preparation of a version control system (2) Preparation of a customer information management system (3) Preparation of the maintenance environment	Build an environment necessary for maintenance (version control system, customer information management system, maintenance development environment, etc.).	• Version control methods • Customer information management methods • Maintenance of the development environment	• Considering and determining version control procedures • Considering and determining customer management procedures • Considering the method to maintain the development environment to ensure recovery after maintenance
	6-4 Implementation of maintenance	(1) Instructions for the maintenance work (2) Impact analysis and implementation of maintenance work (failure correction and functional improvement) (3) Implementation of change management and complaint management (4) Reflection in the next project	Confirm that work for failure correction and functional improvement does not deteriorate reliability, and perform the work. Record and make use of the details of changes and complaints. Furthermore, analyze the complaints, and reflect the results in the next project.	• Specific implementation items according to type of maintenance work • Update procedures • Analysis of impact due to software function change • Writing presentation skills	• Analyzing a failure, ascertaining the causes, and determining necessary maintenance details • Evaluating the quality of the determined contents of maintenance • Building a necessary framework according to the maintenance details, and taking appropriate measures • Coordinating with other departments • Considering and implementing emergency measures • Determining the scope of impact caused by implementing maintenance, and taking appropriate measures

■ **Embedded Systems Specialist Examination (Level 4)**
Syllabus (Version 3.1)

Information-technology Promotion Agency, Japan
IT Human Resources Development Headquarters,
Japan Information-Technology Engineers Examination Center (JITEC)

15th Floor, Bunkyo Green Court Center Office, 2-28-8, Hon-Komagome,
Bunkyo-ku, Tokyo 113-6591 Japan
Tel: 03-5978-7600 (main switchboard) Fax: 03-5978-7610
Website: http://www.jitec.ipa.go.jp/

2016-09