

**SRM VALLIAMMAI ENGINEERING COLLEGE**  
**(An Autonomous Institution)**

SRM Nagar, Kattankulathur – 603 203

**DEPARTMENT OF ELECTRONICS AND COMMUNICATION**  
**ENGINEERING**

**QUESTION BANK**



**VI SEMESTER**

**PEC503 – IoT Processors**

**Regulation – 2023**

**Academic Year 2025-2026 (Even Semester)**

*Prepared by*

**Dr. B. Sridhar, Associate Professor/ECE**

**Mr. Murugesan. D, Assistant Professor/ECE**

# SRM VALLIAMMAI ENGINEERING COLLEGE

(An Autonomous Institution)

SRM Nagar, Kattankulathur – 603 203

## DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

### QUESTION BANK

**SUBJECT CODE & TITLE: PEC503 – IoT Processors**

**SEM / YEAR: V / III**

#### UNIT – I: OVERVIEW OF ARM AND CORTEX-M3

ARM Architecture – Versions, Instruction Set Development, Thumb 2 and Instruction Set Architecture, Cortex M3 Basics: Registers, Stack Pointer, Link Register, Program Counter, Special Registers, Operation Mode, Exceptions and Interrupts, Vector Tables, Stack Memory Operations, Reset Sequence, CORTEX M3 Instruction Sets: Assembly Basics, Instruction List, Instruction Descriptions.

#### PART – A

Q.No	Questions	CO	BT Level	Competence
1	Define ARM architecture.	CO1	BTL2	Understand
2	List the different versions of ARM architecture.	CO1	BTL2	Understand
3	State the purpose of Instruction Set Architecture (ISA).	CO1	BTL2	Understand
4	Identify the core used in ARM Cortex-M3.	CO1	BTL2	Understand
5	Name the general-purpose registers in Cortex-M3.	CO1	BTL1	Remember
6	List the special registers available in Cortex-M3.	CO1	BTL2	Understand
7	State the function of the Stack Pointer (SP).	CO1	BTL1	Remember
8	State the role of the Link Register (LR).	CO1	BTL2	Understand
9	Identify the function of the Program Counter (PC).	CO1	BTL1	Remember
10	List the types of exceptions supported by Cortex-M3.	CO1	BTL1	Remember
11	Name the components of the vector table.	CO1	BTL2	Understand
12	List any four Thumb-2 instructions.	CO1	BTL2	Understand
13	Explain the evolution of ARM instruction sets.	CO1	BTL2	Understand
14	Mention the significance of Thumb-2 technology.	CO1	BTL2	Understand
15	How the Stack Pointer operates in Cortex-M3.	CO1	BTL1	Remember
16	Write the function of special registers in Cortex-M3.	CO1	BTL2	Understand
17	List the operation modes of Cortex-M3.	CO1	BTL1	Remember
18	Describe the role of vector tables in interrupt handling.	CO1	BTL2	Understand
19	How exceptions differ from interrupts in Cortex-M3.	CO1	BTL2	Understand
20	Describe the reset sequence of the Cortex-M3 processor.	CO1	BTL2	Understand
21	Explain stack memory operations during subroutine calls.	CO1	BTL2	Understand
22	Narrate the basic features of Cortex-M3 assembly language.	CO1	BTL2	Understand
23	Give the importance of instruction descriptions in ARM assembly programming.	CO1	BTL2	Understand
24	How Thumb-2 instructions improve code density.	CO1	BTL1	Remember

#### PART – B

1	Demonstrate the use of Thumb-2 instructions for arithmetic operations with a suitable example.	CO1	BTL3	Apply
2	Illustrate the function call mechanism in Cortex-M3 using stack operations.	CO1	BTL3	Apply
3	Apply the reset sequence steps of Cortex-M3 in a typical embedded system startup.	CO1	BTL4	Analyze
4	Show how exceptions are handled in Cortex-M3 with a neat flow diagram.	CO1	BTL3	Apply
5	Demonstrate data transfer between registers using Cortex-M3 assembly instructions.	CO1	BTL3	Apply
6	Illustrate the usage of Stack Pointer during nested subroutine calls.	CO1	BTL3	Apply
7	Apply Thumb-2 branch instructions in program control flow.	CO1	BTL4	Analyze
8	Demonstrate interrupt servicing using vector table entries in Cortex-M3.	CO1	BTL3	Apply
9	Show the role of Link Register during subroutine execution with an example.	CO1	BTL4	Analyze
10	Analyze the differences between ARM and Thumb-2 instruction sets in terms of performance and code size.	CO1	BTL3	Apply
11	Analyze the impact of exceptions and interrupts on program execution in Cortex-M3.	CO1	BTL3	Apply
12	Differentiate the functions of Program Counter and Link Register during instruction execution.	CO1	BTL4	Analyze
13	Analyze the structure and significance of the vector table in Cortex-M3.	CO1	BTL4	Analyze
14	Compare and analyze stack memory operations during normal execution and interrupt handling.	CO1	BTL3	Apply
15	Analyze how operation modes affect exception handling in Cortex-M3.	CO1	BTL4	Analyze
16	Differentiate general-purpose registers and special registers with respect to their usage.	CO1	BTL4	Analyze
17	Analyze the role of Thumb-2 instruction set in improving embedded system efficiency.	CO1	BTL3	Apply

## UNIT - II: CORTEX EXCEPTION HANDLING AND INTERRUPTS

Exception Types, Priority, Vector Tables, Interrupt Inputs and Pending behaviour, Fault Exceptions, Supervisor Call and Pendable Service Call, NVIC: Nested Vector Interrupt Controller, Basic Interrupts, SYSTICK Time, Interrupt Behaviour and Interrupt/Exception Sequences, Exception Exits, Nested Interrupts, Tail – Chaining Interrupts, Late Arrivals and Interrupt Latency.

### PART – A

Q.No.	Questions	CO	BT Level	Competence
1	Define an exception in ARM Cortex-M processors.	CO2	BTL1	Remember
2	List the different types of exceptions supported by Cortex-M.	CO2	BTL2	Understand
3	State the purpose of exception priority levels.	CO2	BTL1	Remember
4	Identify the role of the vector table in interrupt handling.	CO2	BTL2	Understand
5	Name the basic interrupt sources in Cortex-M processors.	CO2	BTL2	Understand
6	Define NVIC (Nested Vector Interrupt Controller).	CO2	BTL1	Remember
7	List any two fault exceptions in Cortex-M.	CO2	BTL1	Remember
8	State the function of Supervisor Call (SVC).	CO2	BTL2	Understand
9	Identify the purpose of Pendable Service Call (PendSV).	CO2	BTL2	Understand
10	Name the system timer used in Cortex-M.	CO2	BTL1	Remember
11	State the use of SysTick timer.	CO2	BTL2	Understand
12	List the stages involved in an interrupt sequence.	CO2	BTL2	Understand
13	Explain how exception priority affects interrupt execution.	CO2	BTL2	Understand
14	Describe the pending behavior of interrupt inputs.	CO2	BTL1	Remember
15	Explain the significance of fault exceptions in system reliability.	CO2	BTL2	Understand
16	Describe the working of NVIC in handling nested interrupts.	CO2	BTL2	Understand

17	Explain the role of SysTick in real-time applications.	CO2	BTL2	Understand
18	Describe the interrupt behavior during exception entry.	CO2	BTL2	Understand
19	Explain the concept of nested interrupts with an example.	CO2	BTL1	Remember
20	Describe the exception exit process in Cortex-M.	CO2	BTL2	Understand
21	Explain tail-chaining in interrupt processing.	CO2	BTL1	Remember
22	Describe the concept of late arrival of interrupts.	CO2	BTL1	Remember
23	Explain how interrupt latency is minimized in Cortex-M.	CO2	BTL1	Remember
24	Describe the relationship between interrupts and exception sequences.	CO2	BTL2	Understand

**PART – B**

1	Illustrate the exception entry sequence in an ARM Cortex-M processor with a neat diagram.	CO2	BTL4	Analyze
2	Demonstrate how NVIC prioritizes multiple interrupt requests using an example.	CO2	BTL4	Analyze
3	Apply the vector table mechanism to handle an external interrupt in Cortex-M.	CO2	BTL4	Analyze
4	Show how pending interrupts are managed by the NVIC.	CO2	BTL3	Apply
5	Demonstrate the working of SysTick timer for generating periodic interrupts.	CO2	BTL3	Apply
6	Apply the Supervisor Call (SVC) mechanism in a user-level application.	CO2	BTL3	Apply
7	Illustrate the use of PendSV for context switching in a real-time system.	CO2	BTL4	Analyze
8	Demonstrate the handling of fault exceptions with a suitable example.	CO2	BTL3	Apply
9	Apply the exception exit sequence during interrupt completion.	CO2	BTL4	Analyze
10	Analyze the impact of exception priority on nested interrupt execution.	CO2	BTL3	Apply
11	Analyze the role of NVIC in minimizing interrupt latency.	CO2	BTL3	Apply
12	Differentiate and analyze fault exceptions and system exceptions.	CO2	BTL3	Apply
13	Analyze the behavior of nested interrupts in Cortex-M processors.	CO2	BTL3	Apply
14	Analyze tail-chaining and its effect on interrupt response time.	CO2	BTL3	Apply
15	Analyze the concept of late arrival of interrupts and its influence on system performance.	CO2	BTL4	Analyze
16	Analyze the complete interrupt/exception sequence from request to return.	CO2	BTL4	Analyze
17	Analyze the factors contributing to interrupt latency in Cortex-M.	CO2	BTL3	Apply

**UNIT - III: CORTEX M3/M4 PROGRAMMING**

Cortex M3/M4 Programming: Typical Development Flow, CMSIS Using ALP, Exception Programming Using Interrupts, Exception/Interrupt Handlers, Software Interrupts, Vector Table Relocation, Memory Protection Unit, MPU Registers, Setting up the MPU, Power Management, Multiprocessor Configuration.

**PART – A**

Q.No.	Questions	CO	BT Level	Competence
1	Define CMSIS in ARM Cortex-M programming.	CO3	BTL1	Remember
2	List the stages involved in a typical Cortex-M3/M4 development flow.	CO3	BTL2	Understand
3	State the purpose of Application Layer Programming (ALP).	CO3	BTL1	Remember
4	Identify the role of interrupt handlers in exception programming.	CO3	BTL1	Remember
5	Name the function used to generate a software interrupt in Cortex-M.	CO3	BTL2	Understand
6	List any two features of CMSIS.	CO3	BTL2	Understand
7	State the purpose of vector table relocation.	CO3	BTL1	Remember
8	Define Memory Protection Unit (MPU).	CO3	BTL2	Understand
9	Name the key registers used in MPU configuration.	CO3	BTL1	Remember
10	State the function of MPU region registers.	CO3	BTL2	Understand

11	List the power modes supported by Cortex-M processors.	CO3	BTL2	Understand
12	Identify the role of power management in embedded systems.	CO3	BTL2	Understand
13	Explain the importance of CMSIS in Cortex-M programming.	CO3	BTL1	Remember
14	Describe the typical development flow of a Cortex-M3/M4 application.	CO3	BTL2	Understand
15	Explain how exceptions are programmed using interrupts.	CO3	BTL1	Remember
16	Describe the working of an interrupt/exception handler.	CO3	BTL2	Understand
17	Explain the concept of software interrupts with an example.	CO3	BTL2	Understand
18	Describe the need for vector table relocation.	CO3	BTL2	Understand
19	Explain how the MPU enhances system security.	CO3	BTL1	Remember
20	Describe the process of setting up the MPU.	CO3	BTL2	Understand
21	Explain the role of MPU registers in memory protection.	CO3	BTL1	Remember
22	Describe the importance of power management in Cortex-M systems.	CO3	BTL2	Understand
23	Explain how low-power modes affect program execution.	CO3	BTL2	Understand
24	Describe the concept of multiprocessor configuration in Cortex-M systems.	CO3	BTL2	Understand

### PART – B

1	Illustrate the typical development flow of a Cortex-M3/M4 based embedded system.	CO3	BTL3	Apply
2	Demonstrate the use of CMSIS functions in Application Layer Programming (ALP).	CO3	BTL4	Analyze
3	Apply exception programming using interrupts in a Cortex-M processor.	CO3	BTL3	Apply
4	Show how an interrupt handler is written and linked to the vector table.	CO3	BTL3	Apply
5	Demonstrate the generation of a software interrupt using SVC instruction.	CO3	BTL3	Apply
6	Illustrate the steps involved in vector table relocation with a suitable example.	CO3	BTL3	Apply
7	Apply the configuration steps required to enable the Memory Protection Unit (MPU).	CO3	BTL4	Analyze
8	Demonstrate the use of MPU registers to protect a memory region.	CO3	BTL3	Apply
9	Apply power management techniques to place the Cortex-M processor into low-power mode.	CO3	BTL4	Analyze
10	Analyze the role of CMSIS in simplifying Cortex-M3/M4 application development.	CO3	BTL3	Apply
11	Analyze the behavior of exceptions and interrupts during program execution.	CO3	BTL4	Analyze
12	Analyze the effectiveness of software interrupts in system control.	CO3	BTL3	Apply
13	Differentiate and analyze the default and relocated vector tables.	CO3	BTL3	Apply
14	Analyze how MPU configuration improves system reliability and security.	CO3	BTL3	Apply
15	Analyze the impact of power management modes on system performance.	CO3	BTL4	Analyze
16	Analyze the interaction between exception handlers and application code.	CO3	BTL4	Analyze
17	Analyze the challenges involved in multiprocessor configuration in Cortex-M systems.	CO3	BTL3	Apply

### UNIT - IV: ARMCORTEX M3/M4 MICROCONTROLLER AND DEBUGGING TOOLS

STM32L15XXX ARM CORTEX M3/M4 Microcontroller: Memory and Bus Architecture, Power Control, Reset and Clock Control, STM32L15XXX Peripherals: GPIOs, System Configuration Controller, NVIC, ADC, Comparators, GP Timers, USART Development and Debugging Tools: Software and Hardware tools – Cross Assembler Compiler, Debugger, Simulator, In-Circuit Emulator, Logic Analyser

### PART – A

Q.No.	Questions	CO	BT Level	Competence
1	What is the role of the Harvard architecture in ARM Cortex-M microcontrollers?	CO4	BTL1	Remember
2	List the different memory types available in STM32L15xxx microcontrollers.	CO4	BTL1	Remember

3	What is the function of the AHB and APB buses in STM32 architecture?	CO4	BTL1	Remember
4	Explain how memory mapping is organized in STM32L15xxx devices.	CO4	BTL2	Understand
5	Why is Flash memory preferred for program storage in microcontrollers?	CO4	BTL2	Understand
6	What are the main power modes supported by STM32L15xxx microcontrollers?	CO4	BTL1	Remember
7	Define brown-out reset (BOR) in STM32 microcontrollers.	CO4	BTL1	Remember
8	What is the function of the Reset and Clock Control (RCC) unit?	CO4	BTL1	Remember
9	Differentiate between HSI and HSE clock sources.	CO4	BTL2	Understand
10	Explain the importance of PLL in clock generation.	CO4	BTL2	Understand
11	What is the function of the Nested Vectored Interrupt Controller (NVIC)?	CO4	BTL1	Remember
12	List any two advantages of vectored interrupts in Cortex-M processors.	CO4	BTL1	Remember
13	Explain how interrupt priority is handled in NVIC.	CO4	BTL2	Understand
14	What is the role of the System Configuration Controller (SYSCFG)?	CO4	BTL1	Remember
15	What are the different modes of GPIO operation in STM32 microcontrollers?	CO4	BTL1	Remember
16	Why are alternate function pins required in GPIO configuration?	CO4	BTL2	Understand
17	State the purpose of the ADC in STM32L15xxx microcontrollers.	CO4	BTL1	Remember
18	Explain the need for sample-and-hold circuits in ADCs.	CO4	BTL2	Understand
19	What is the function of analog comparators in STM32 devices?	CO4	BTL1	Remember
20	List the applications of general-purpose timers in microcontrollers.	CO4	BTL1	Remember
21	Explain how timers are used for PWM generation.	CO4	BTL2	Understand
22	What is USART and mention its two modes of operation.	CO4	BTL1	Remember
23	Differentiate between synchronous and asynchronous communication in USART.	CO4	BTL2	Understand
24	Explain the importance of an in-circuit emulator (ICE) during debugging.	CO4	BTL2	Understand

### PART – B

1	Explain the memory organization and bus architecture of STM32L15xxx ARM Cortex-M microcontrollers. Discuss the roles of Flash, SRAM, peripheral memory map, AHB, and APB buses with a neat diagram.	CO4	BTL4	Analyze
2	Describe the Harvard architecture used in ARM Cortex-M processors and explain how it improves instruction execution efficiency in STM32L15xxx devices.	CO4	BTL3	Apply
3	Discuss the power control architecture of STM32L15xxx microcontrollers. Compare the various low-power modes and explain their suitability for battery-powered applications.	CO4	BTL4	Analyze
4	Explain the Reset and Clock Control (RCC) system of STM32L15xxx and describe the procedure for system clock generation using HSE and PLL.	CO4	BTL3	Apply
5	Examine the different reset sources available in STM32L15xxx microcontrollers and justify their importance in ensuring system reliability.	CO4	BTL4	Analyze
6	Explain the architecture and functioning of the Nested Vectored Interrupt Controller (NVIC) and discuss its role in interrupt prioritization and latency reduction.	CO4	BTL4	Analyze
7	Describe how multiple interrupts are handled in STM32 microcontrollers using NVIC, with suitable examples from real-time embedded systems.	CO4	BTL3	Apply
8	Explain the configuration of GPIO ports in STM32L15xxx microcontrollers and describe how they are used for interfacing external devices.	CO4	BTL3	Apply
9	Discuss the role of the System Configuration Controller (SYSCFG) in STM32L15xxx and explain its significance in external interrupt and alternate function configuration.	CO4	BTL4	Analyze
10	Explain the working principle of the ADC in STM32L15xxx microcontrollers, detailing sampling, conversion process, and digital data representation.	CO4	BTL3	Apply
11	Discuss the factors affecting ADC performance in STM32 microcontrollers and explain how accuracy and resolution are influenced.	CO4	BTL4	Analyze
12	Explain the operation of analog comparators in STM32L15xxx and describe their use in voltage monitoring applications.	CO4	BTL3	Apply
13	Explain the operation of general-purpose timers in STM32L15xxx and	CO4	BTL3	Apply

	describe how PWM signals are generated for control applications.			
14	Compare different timer modes available in STM32 microcontrollers and discuss their applications in embedded systems.	CO4	BTL4	Analyze
15	Explain the USART architecture of STM32L15xxx and describe the steps involved in serial communication setup	CO4	BTL4	Analyze
16	Discuss the embedded software development flow and explain the roles of cross-assembler, compiler, linker, debugger, and simulator.	CO4	BTL4	Analyze
17	Examine the importance of hardware debugging tools such as in-circuit emulator and logic analyzer in troubleshooting STM32-based systems.	CO4	BTL4	Analyze

### UNIT - V: INTRODUCTION TO SYSTEM – ON – CHIP

System Architecture: Components of the System Processors, Memories and Interconnects, Processor Architectures, Memory and Addressing, System Level Interconnection – An Approach for SOC Design – Chip basics – Cycle Time – Die Area – Power and Cost – Tradeoff in Processor Design – Reliability and Configurability – SOC Design Approach.

#### PART – A

Q.No.	Questions	CO	BT Level	Competence
1	Define system architecture.	CO5	BTL1	Remember
2	List the major components of a system-on-chip (SoC).	CO5	BTL1	Remember
3	What is the role of a processor in a computing system?	CO5	BTL1	Remember
4	Define interconnect in a system architecture.	CO5	BTL1	Remember
5	Explain the relationship between processors, memories, and interconnects in a system.	CO5	BTL2	Understand
6	What is meant by processor architecture?	CO5	BTL1	Remember
7	List any two types of processor architectures.	CO5	BTL1	Remember
8	Differentiate between RISC and CISC architectures.	CO5	BTL2	Understand
9	Explain the significance of pipelining in processor design.	CO5	BTL2	Understand
10	What is the function of a control unit in a processor?	CO5	BTL1	Remember
11	List different types of memory used in SoC design.	CO5	BTL1	Remember
12	Define memory addressing.	CO5	BTL1	Remember
13	Explain the difference between byte addressing and word addressing.	CO5	BTL2	Understand
14	What is the purpose of cache memory in a system?	CO5	BTL1	Remember
15	Explain how memory hierarchy improves system performance.	CO5	BTL2	Understand
16	What is system-level interconnection?	CO5	BTL1	Remember
17	List any two types of interconnect structures used in SoC.	CO5	BTL1	Remember
18	Explain the importance of bus arbitration in shared interconnects.	CO5	BTL2	Understand
19	Differentiate between bus-based and network-on-chip (NoC) interconnects.	CO5	BTL2	Understand
20	Define cycle time in digital system design.	CO5	BTL1	Remember
21	What is meant by die area?	CO5	BTL1	Remember
22	Explain how power consumption affects SoC cost.	CO5	BTL2	Understand
23	State the trade-offs involved in processor design.	CO5	BTL1	Remember
24	Explain the importance of reliability in SoC design.	CO5	BTL2	Understand

#### PART – B

1	Explain the overall system architecture of a computing system and discuss the interaction between processors, memories, and interconnects with a suitable block diagram.	CO5	BTL4	Analyze
2	Describe the role of processors in system architecture and explain how processor selection influences overall system performance.	CO5	BTL3	Apply
3	Discuss different processor architectures used in modern SoC designs and compare their suitability for performance-oriented and low-power applications.	CO5	BTL4	Analyze

4	Compare RISC and CISC processor architectures and justify their relevance in contemporary embedded and SoC-based systems.	CO5	BTL4	Analyze
5	Explain the concept of pipelining in processor architecture and illustrate its effect on instruction throughput and execution time.	CO5	BTL3	Apply
6	Discuss the major trade-offs involved in processor design with respect to performance, power consumption, area, and complexity.	CO5	BTL4	Analyze
7	Explain the organization of memory in a system architecture and describe how different memory types are integrated into an SoC.	CO5	BTL3	Apply
8	Discuss memory addressing schemes and compare byte-addressable and word-addressable architectures with suitable examples.	CO5	BTL4	Analyze
9	Examine the concept of memory hierarchy and explain how it contributes to improved system performance and efficiency.	CO5	BTL4	Analyze
10	Explain system-level interconnection in SoC design and describe the function of buses and interconnect networks.	CO5	BTL3	Apply
11	Compare bus-based interconnects and Network-on-Chip (NoC) architectures and discuss their advantages and limitations.	CO5	BTL4	Analyze
12	Explain chip-level design basics and describe the significance of cycle time in determining system performance.	CO5	BTL3	Apply
13	Discuss the impact of die area on chip cost and manufacturing yield in SoC design.	CO5	BTL4	Analyze
14	Explain the relationship between power consumption and cost in SoC design and discuss techniques to balance both.	CO5	BTL4	Analyze
15	Discuss the importance of reliability in SoC design and explain how design choices influence system robustness.	CO5	BTL4	Analyze
16	Explain configurability in SoC design and describe how it enhances flexibility and reuse across applications.	CO5	BTL3	Apply
17	Elaborate on a systematic SoC design approach, explaining the major stages involved from system specification to implementation.	CO5	BTL4	Analyze