MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

MUST KNOW CONCEPTS

MKC

ECE

2021-22

Subject		19ECC15 / Embedded Systems and RTOS		
S. No.	**Term**	**Notation Symbol)**	**Concept/Definition/Meaning/Units/Equation/Expression**	**Units**
UNIT I INTRODUCTION				
1.	System		It is an arrangement in which all its units assemble and work together according to the plan or program.	
2.	Embedded System		It is a combination of hardware and software, either fixed in capability or programmable, designed for a specific function or functions within a larger system.	
3.	Components of an Embedded System		• Hardware, Application Software, • Real Time Operating Systems (RTOS)	
4.	Constraints		• Available System Memory, • Available Processor Speed, • limit power dissipation	
5.	Classifications of Embedded Systems		• Small Scale Embedded System, • Medium Scale Embedded System, • Sophisticated Embedded System	
6.	Small Scale Embedded Systems		An entry-level system in which 8-bit or 16-bit processor is used. The processor has very limited resources like RAM, ROM and processing speed.	
7.	Medium Scale Embedded Systems		These systems are usually designed with a single or few 16-bit or 32-bit microcontrollers or Digital Signal processor.	
8.	Sophisticated Embedded Systems		The embedded system which can do large-scale works with multiple 32 - 64-bit chips.	
9.	Applications of Embedded System		Consumer electronics, Consumer products, Automobiles, Industrial process controllers & avionics/defense	
10.	General Purpose Processor		A Programmable digital system intended to solve computation tasks in a large variety of applications.	
11.	DSP Processor		A specialized microprocessor (or a SIP block) chip, with its architecture optimized for the operational needs of digital signal processing.	

12.	Microprocessor		It is a central processing unit on a single integrated circuit chip containing millions of very small components.	
13.	Embedded microcontroller		It is particularly suited for embedded applications to perform dedicated task or operation.	
14.	SOC		System on chip is an embedded systems are being designed on a single silicon chip.	
15.	Important considerations when selecting a processor		• Instruction set, • Maximum bits in an operand, • Clock frequency • Processor ability	
16.	Requirements of embedded system		• Reliability, • Low power consumption, • Cost effectiveness, • Efficient use of processing power	
17.	FPGA		Field-Programmable Gate Array is an integrated circuit designed to be configured by a customer or a designer after manufacturing	
18.	Quality Attributes of Embedded system		• Operational quality attributes • Non-operational quality attributes	
19.	Operational Quality attributes		• Response • Throughput • Reliability • Maintainability • Security • Safety	
20.	Non-operational quality attributes		• Testability and Debug-ability • Evolvability • Portability • Time to prototype and market • Per unit and total cost	
21.	Characteristics of Embedded System		• Application and domain specific, • Tightly constrained, • Reactive and real time • Operation in harsh environment • Distributed • Compact in size and light weight	
22.	Design metrics		• Power, • Size, • NRE cost, • Performance	
23.	Challenges Of Embedded Systems		• Hardware needed, • Meeting the deadlines, • Minimizing the power consumption, • Design for upgradeability	

24.	Steps in Embedded System Design		• Requirements, • Specifications, • Architecture, • Components • System integration	
25.	Major Application Areas of Embedded Systems		• Consumer Electronics • Household Appliances • Home Automation and Security Systems • Automotive Industry • Telecom • Computer Peripherals • Computer Networking Systems • Health Care • Measurement & Instrumentation • Banking & Retail • Card Readers	
UNIT II TYPICAL EMEDDED SYSTEM				
26.	ASSP		Application Specific System Processor is a processing unit for specific tasks like image compression and it provides a faster solution.	
27.	common types of memories used in embedded systems for control algorithm storage		OTP, PROM, UVEPROM, EEPROM and FLASH	
28.	Core of the Embedded Systems		• General Purpose and Domain Specific Processors ✓ Microprocessors ✓ Microcontrollers ✓ Digital Signal Processors • Programmable Logic Devices (PLDs) • Application Specific Integrated Circuits (ASICs) • Commercial off the shelf Components (COTS)	
29.	Microcontroller		A highly integrated silicon chip containing a CPU, scratch pad RAM, Special and General-purpose Register Arrays, On Chip ROM/FLASH memory for program storage, Timer and Interrupt control units and dedicated I/O ports	
30.	General Purpose Processor		General Purpose Processor or GPP is a processor designed for general computational tasks	
31.	ASIP		Application Specific Instruction Set processors (ASIPs) are processors with architecture and instruction set optimized to specific domain/application requirements like Network processing, Automotive, Telecom, media applications, digital signal processing, control applications etc.	
32.	Examples of Application		• Microcontrollers (like Automotive AVR, USB AVR from Atmel),	

	Specific Instruction Set Processors (ASIPs)		• System on Chips, • Digital Signal Processors etc	
33.	Key units in Digital Signal Processor		• Program Memory • Key memory • Computational Engine • I/O unit	
34.	Types of processor architecture design		• Harvard architecture • Von-Neumann architecture	
35.	Von-Neumann architecture		Von-Neumann architecture shares a single common bus for fetching both instructions and data. Program instructions and data are stored in a common main memory	
36.	Harvard architecture		Microprocessors/controllers based on the Harvard architecture will have separate data bus and instruction bus. This allows the data transfer and program fetching to occur simultaneously on both buses	
37.	Endianness		Endianness specifies the order in which the data is stored in the memory by processor operations in a multi byte system	
38.	Little-endian		Little-endian means the lower-order byte of the data is stored in memory at the lowest address, and the higher-order byte at the highest address	
39.	Big-endian		Big-endian means the higher-order byte of the data is stored in memory at the lowest address, and the lower-order byte at the highest address. (The big end comes first.)	
40.	Programmable logic devices (PLDs)		Devices that can be re-configured to perform any number of functions at any time	
41.	Types of PLDs		• Field Programmable Gate Arrays (FPGAs) and • Complex Programmable Logic Devices (CPLDs)	
42.	FPGA		FPGA is an IC designed to be configured by a designer after manufacturing	
43.	FPGA applications		• data processing and storage, • instrumentation, • telecommunications, and • digital signal processing	
44.	CPLD		A complex programmable logic device (CPLD) is a programmable logic device with complexity between that of PALs and FPGAs, and architectural features of both.	
45.	Commercial off the Shelf Component (COTS)		COTS products are designed in such a way to provide easy integration and interoperability with existing system components	
46.	Program Storage Memory		Stores the program instructions	
47.	Masked ROM (MROM)		One-time programmable memory	

48.	Programmable Read Only Memory (PROM) / (OTP)		OTP is widely used for commercial production of embedded systems whose proto-typed versions are proven and the code is finalized.	
49.	EPROM		Erasable Programmable Read Only (EPROM) memory gives the flexibility to re-program the same chip.	
50.	Advantage of EPROM		Erasable Programmable Read Only (EPROM) memory gives the flexibility to re-program the same chip using electrical signals	
UNIT-III: EMBEDDED FIRMWARE				
51.	Firmware		• A specific class of computer software that provides the low-level control for the device's specific hardware. • Simply, it is the software substituted for hardware and stored in ROM.	
52.	Real-time clock		Much like a digital wristwatch, a *real-time clock* (RTC) keeps the time and date in an embedded system.	
53.	Common examples of fireware		• Typical examples of devices containing firmware are embedded systems (running embedded software), home and personal-use appliances, computers, and computer peripherals. • Firmware is held in non-volatile memory devices such as ROM, EPROM, EEPROM, and Flash memory.	
54.	Brownout reset circuit		A brownout reset is a circuit that causes a computer processor to reset (or reboot) in the event of a brownout, which is a significant drop in the power supply output voltage.	
55.	Reset circuit		A power on reset circuit ensures the system power supply stabilizes at the correct levels, the clocks of the processors settle accurately, and that the loading of the internal registers is complete before the device actually starts working or gets powered up.	
56.	Brown-out condition		A brownout, sometimes also called a 'sag', is a "dip" in the voltage level of the electrical line. When a brownout occurs, the voltage drops from its normal level to a lower voltage and then returns.	
57.	PICs that do not support the Brown-Out-Reset (BOR) feature		• PIC 16C61 • PIC 16C71	
58.	Timer		A *timer* is a device that generates a signal pulse at specified time intervals.	
59.	Counter		*Counter* is nearly identical to a timer, except that instead of counting clock cycles (pulses on the clock signal), a counter counts pulses on some other input signal.	
60.	Watchdog Timer		• A watchdog timer (WDT) is a timer that monitors microcontroller (MCU) programs to see if they are	

			out of control or have stopped operating. • It acts as a "watchdog" watching over MCU operation.	
61.	Types of firmware		• Low-level firmware, • High-level firmware, and • Subsystem	
62.	Function of watchdog timer		Its main function is to protect the system against malfunctions.	
63.	Embedded firmware design approaches		• Super loop-based approach, • Operating system based approach	
64.	System components of embedded firmware		Firmware is also known in the industry as "embedded software" or "low-level software." Major firmware components optionally include • An operating system (OS), • Kernel, • Device drivers, and • Application code.	
65.	Difference between firmware and embedded software		• Embedded software typically implements higher-level features and functions of the device. • Firmware takes care of low-level tasks such as converting analog sensor signals to digital data and managing communications protocols.	
66.	Functions of firmware		• Firmware assumes an intermediary role between the hardware and software – including potential future upgrades of the software. • Some firmware (such as the BIOS on a PC) does the job of booting up a computer by initialising the hardware components and loading the operating system.	
67.	Difference between hardware and firmware		• A hardware has a physical entity and can undergo physical damage, unlike a firmware. • A hardware needs a program to run. A firmware is a program itself. • A hardware cannot operate without a firmware. A firmware operates on a hardware.	
68.	Embedded firmware Development Languages/Options		• Assembly Language • High Level Language • Mix of Assembly & High-level Language	
69.	High Level Language for Embedded firmware development		• Subset of C (Embedded C) o Subset of C++ (Embedded C++) • Any other high-level language with supported Cross-compiler	
70.	Mix of Assembly & High level Language		• Mixing High Level Language (Like C) with Assembly Code • Mixing Assembly code with High Level Language	

			(Like C) • Inline Assembly	
71.	Advantage of High Level Language for Embedded firmware development		• Reduced Development time • Developer independency • Portability	
72.	Cross-compiler		A software utility that converts the high level language to target processor specific machine code	
73.	Assembly language program		Program written in assembly code is saved as .asm (Assembly file) file or a .src (source) file or a format supported by the assemble	
74.	Assembler		The software that performs the translation of assembly code to machine code	
75.	Advantages of assembly language programming		• Efficient Code Memory & Data Memory Usage (Memory Optimization) • High Performance • Low level hardware access • Code reverse engineering	
UNIT IV: REAL TIME OPERATING SYSTEMS				
76.	RTOS		Real Time Operating System is an OS for embedded systems, as these have real time programming issues to solve.	
77.	Processes		A computational unit that processes on a CPU and whose state changes under the control of kernel of an OS.	
78.	PCB		Process Control Block is a data structure having the information using which the OS controls the process state.	
79.	Thread		A process or sub process within a process that has its own PC, its own SP and stack, its own priority parameter for its scheduling.	
80.	Task		Task is a set of computations or actions that processes on a CPU under control of a scheduling kernel.	
81.	Task states		• Idle state, • Ready state, • Running State, • Blocked State, • Deleted State	
82.	Characteristics of task		A task is a function, which executes on scheduling. A task can wait as well as post the events or signals or messages.	
83.	Inter process communication		An output from one task passed to another task through the scheduler and use of signals, exception, semaphore, queues, mailbox, pipes, sockets, and RPC.	
84.	Semaphore		Semaphore provides a mechanism to let a task wait till another finishes. It is a way of synchronizing concurrent processing operations.	
85.	Mutex		Mutex is a semaphore that gives at an instance two tasks mutually exclusive access to resources.	

86.	Priority inversion		A problem in which a low priority task inadvertently does not release the process for a higher priority task	
87.	Deadlock situation		A set of processes or threads is deadlocked when each process or thread is waiting for a resource to be freed which is controlled by another process.	
88.	Socket		It provides the logical link using a protocol between the tasks in a client server or peer to peer environment.	
89.	Remote Procedure Call		A method used for connecting two remotely placed methods by using a protocol.	
90.	Goals of RTOS		• Facilitating easy sharing of resources, • Facilitating easy implantation of the application software, • Maximizing system performance	
91.	Kernel		It is a computer program at the core of a computer's operating system with complete control over everything in the system	
92.	Functions of a kernel		• Process management, • Process creation to deletion, • Processing resource requests Scheduling, • IPC, • Memory management, • I/O management, • Device management	
93.	Non-preemptive scheduling		Once the CPU has been allocated to a process, the process keeps the CPU until it releases the CPU either by terminating or switching to the waiting state.	
94.	Preemptive scheduling		Preemptive scheduling can preempt a process which is utilizing the CPU in between its execution and give the CPU to another process.	
95.	Two important RTOS		• µCOS • VxWorks	
96.	Vxworks		Vxworks is a popular Real-time multi- tasking operating system for embedded microprocessors and systems.	
97.	Features of µC/OS II		• Preemptive, • Portable, • Scalable, • Multitasking	
98.	Application for the VxWorks RTOS		• Automobiles, • Avionics, • Consumer electronics • Medical devices, • Military, • Aerospace, • Networking	
99.	Features of VxWorks		• High performance, • Host and target-based development approach, • Supports advanced processor architecture.	

100.	Basic functions of µCOS		• System level, Task service function, • Task delay, • Memory allocation and partitioning, • IPCs, • mailbox and queues	
UNIT V: RTOS-BASED EMBEDDED SYSTEM DESIGN				
101.	Digital camera		Cameras that we use today are smart and have a lot of features that were not present in early cameras all because of embedded system used in them	
102.	RT Linux		Hard real-time real time operating system microkernel that runs the entire linux operating system as a fully preemptive process.	
103.	Washing Machine		Allows the washing machine to adjust the water and cut the power automatically.	
104.	Home Security Systems		Collects the information of the parameter such as temperature, gas, fire, human presence etc and sends the information to the microcontroller.	
105.	Cruise control		A system that takes charge of controlling the throttle from the driver and cruising the vehicle at preset and constant speed.	
106.	Adaptive control		An embedded system is adaptive if it is able to adjust its.	
107.	Adaptive control algorithm		An embedded system algorithm is adaptive if it is able to adjust its.	
108.	Application area		• Digital electronics, • Telecommunications, • Computing network, • Smart cards, • Satellite systems, • Military defense system equipment, • Research system equipment, and so on	
109.	Soft embedded systems		A real-time task is said to be soft if generating the output after its deadline has still some utility for the system, although causing a performance degradation.	
110.	Hard embedded systems		Hard real-time embedded systems include medical systems such as heart pacemakers and industrial process controllers.	
111.	Medial application		• MRI and CT Scanner, • Sonography, • Digital Flow Sensors, • Defibrillator, • Blood Pressure and Glucose Test Set, • Fetal Heart Monitoring Machine, • Wearable Device.	
112.	Military application		• Surveillance and Reconnaissance UAVs,	

			• Communication, • Computing, • Cyber Security. • Vehicle Electronics	
113.	Automation		Embedded systems are designed to help control power efficiency, maximize performance, and control processes while operating in demanding environments	
114.	Consumer application		MP3 players, mobile phones, video game consoles, digital cameras, DVD players, and GPS.	
115.	Handshaking		Handshaking is the exchange of information between two modems and the resulting agreement about which protocol to use that precedes each telephone connection.	
116.	Aurdino		Open-source electronic prototyping platform enabling users to create interactive electronic objects.	
117.	Debug		Debugging is the process of finding and resolving defects or problems within a computer program that prevent correct operation of computer software or a system.	
118.	Office automation		Embedded systems are designed to help control power efficiency, maximize performance, and control processes while operating in demanding environments	
119.	Security		Embedded system security is a strategic approach to protecting software running on embedded systems from attack.	
120.	Tele communication		Telecommunications systems employ numerous embedded systems from telephone switches for the network to cell phones	
121.	Instrumentation		Equipment and appliances irrespective of circuit	
122.	Entertainment		Video games, mp3, mind storm, smart toy, etc	
123.	Banking and finance		Embedded banking allows businesses to have their own software to perform financial operations — weaving the bank's capabilities into the company's life.	
124.	Software Application		That permanently resides in an industrial or consumer device.	
125.	Android		It is a subset of Embedded Linux.	
PLACEMENT QUESTIONS				
126.	Microprocessor		Microprocessor is managers of the resources (I/O, memory) which lie outside of its architecture.	
127.	Microcontroller		The microcontroller is a self-contained system with peripherals, memory and a processor that can be used as embedded system.	
128.	Pre processor		Preprocessor is a program that processes its input data to produce output that is used as input to another program.	

129.	Integrated Circuit		Microchip is an electronic circuit etched onto a silicon chip.	
130.	Firmware		A specific class of computer software that provides the low-level control for the device's specific hardware	
131.	ARM		An ARM processor is one of a family of CPUs based on the RISC	
132.	Semaphore		The semaphore is an abstract data store that is used to control resource accesses across the various threads of execution or across different processes.	
133.	Dead lock		A group of threads are waiting for resources held by others in the group.	
134.	Processor		A computational unit that processes on a CPU and whose state changes under the control of kernel of an OS.	
135.	PCB		PCB is process control block and it is a data structure having the information using which the OS controls the process state.	
136.	RTOS		Real-time operating system-an operating system (OS) intended to serve real-time applications.	
137.	Embedded operating system		A specialized operating system designed to perform a specific task for a device.	
138.	Memory leak		Memory leak is nothing but the accumulation of memory which is not cleared.	
139.	RAM		Random-access memory is a form of computer memory and volatile.	
140.	ROM		Read-only memory is a type of non-volatile memory used in computer.	
141.	Number System		**If the number 481*673 is completely divisible by 9, what is the the smallest whole number in place of *?** **Soln:** => 4+8+1+x+6+7+3) is divisible by 9 => (29+x) is divisible by 9 x should be the smallest whole number. Hence, (29+x)=36 =>x=36−29=7 **Ans : 7**	
142.	Calendar		**Today is Monday. After 61 days, it will be** A. Sunday B. Saturday C. Monday D.Thursday **Soln** 61 days = 8 weeks 5 days = 5 odd days Hence if today is Monday, After 61 days, it will be = (Monday + 5 odd days) = Saturday **Ans :** C.Saturday	
143.	Profit and Loss		**Ramesh bought a chair for Rs. 1540 and sold it to Suresh. If Ramesh earned a profit of 25%, find the selling price of chair.** A. Rs.1875 B. Rs.1900 C.Rs.1925 D.Rs.1950 Soln:	

			C.P. of the chair = Rs. 1540 S.P. of the chair =? Profit earned = 25% Selling Price =(100+Profit%) *C.P. 100 Therefore, S.P. =(100+25) *1540 100 =125/100 *1540=1925 Ans : C Rs.1925	
144.	Time and Distance		**A car running at a speed of 140 km/hr reached its destination in 2 hours. If the car wants to reach at its destination in 1 hour, at what speed it needs to travel?** A. 300 km/hr B. 280 km/hr C. 250 km/hr D. 240 km/hr **Soln:** Distance to be covered = Speed x Time = 140 * 2 = 280 km Time = 1 hour Required Speed = $^{280}/_{1}$= 280 km/hr **Ans: B.280 km/hr**	
145.	Pipes and Cisterns		**Two pipes can fill a tank in 6 hours and 8 hours respectively. A third pipe can empty the same tank in 12 hours. If all the pipes start working together, how long it will take to fill the tank?** A. 4 hours B.4.5 hours C.4.8 hours D.5.2 hours **Soln:** Part of the tank filled by two pipes in one hour = 1/6 + 1/8 Part of the tank emptied by the third pipe in one hour =1/12 Net part of the tank filled in one hour = 1/6+1/8-1/12= 5/24 5/24 Part of tank can be filled in one hour ∴ The whole tank will be filled in 24/5 = 4.8 hours **Ans: C. 4.8 hours**	
146.	Races & Games		**A can run 22.5 m while B runs 25 m. In a one kilometer race, B beats A by** A. 100 m B.111 1/9 m C.25 m D.50 m Soln: When B runs 25 m, A runs 45/2 m When B runs 1000 m, A runs (45/2 * 1/25 * 1000)=900m ∴ B beats A by 100 m. Ans: A. 100 m	
147.	Alligation and Mixture		**A 20 liter mixture contains 30% alcohol and 70% water. If 5 liters of water is added to the mixture, what will be the percentage of alcohol in the new mixture?** A. 22% B.23% C.24% D.25%	

			Soln: Initially, the mixture contains 30% alcohol = 30/100 ∗20=6 liters of alchol And, 70% of water = 70/100 ∗20=14 liters of water After adding 5 liters of water, the mixture contains (14+5) 19 liters of water and 6 liters of alcohol. ∴Percentage of alcohol = 6/25∗100=24% **Ans: C. 24%**	
148.	Logical Reasoning		**SCD, TEF, UGH, ____, WKL** A.CMN B.UJI C.VIJ D. IJT **Soln:** There are two alphabetical series here. The first series is with the first letters only: STUVW. The second series involves the remaining letters: CD, EF, GH, IJ, KL. **Ans: C.** VIJ	
149.	Logical Reasoning		**Which word does NOT belong with the others?** A. Tulip B. Rose C. Bud D.Daisy Tulip, rose, and daisy are all types of flowers. A bud is not. **Ans: C. Bud**	
150.	Logical Reasoning		**Find the next number in the sequence:** **3, 6, 9, 30, 117.....** A. 192 B. 352 C. 388 D. 588 Soln: 3 * 1 + 3 = 6 6 * 2 – 3 = 9 9 * 3 + 3 = 30 30 * 4 – 3 = 117 117 * 5 + 3 = 588 Ans. D 588	

Faculty Team Prepared **Signatures**

1. Dr.R.Praveena, ASP/ECE
2. Dr.U.Saravanakumar, HOD/ECE,
3. Mrs.A.Nivetha, AP/ECE
4. Dr.P.Padmaloshani, ASP/ECE

HOD