K.S.R.COLLEGE OF ENGINEERING (AUTONOMOUS), TIRUCHENGODE-637215
DEPARTMENT OF ELECTRONICS COMMUNICATION AND ENGINEERING
COURSE / LESSION PLAN SCHEDULE

NAME : R.MAHENDRAN, P.MAHENDRAN
SUBJECT : 16EC613 / EMBEDDED SYSTEMS
CLASS : B.E - ECE
YEAR / SEM : III / VI

A.TEXT BOOKS:

1. Raj Kamal, "Embedded Systems Architecture, Programming and Design", Tata McGraw Hill, New Delhi, 2011.

B. REFERENCES

1. K.V.K.K.Prasad, "Embedded Real-Time Systems: Concepts, Design & Programming", Dream tech Press, 2009.
2. David E Simon "An Embedded Software Primer", Pearson Education, 2007.
3. Daniel .W Lewis, "Fundamentals of Embedded Software", Pearson Education, 2001.
4. Jean J. Labrosse, " MicroC/OS – II The Real Time Kernel", CMP Books, 2nd edition, 2002.
5. http://nptel.ac.in/courses/108102045.
6. http://nptel.ac.in/courses/Webcourse-contents/IIT%20Kharagpur/Embedded%20systems.

C. LEGEND:

L	-	Lecture	BB	-	Block Board
OHP	-	Over Head Projector	Tx	-	Text Book
pp	-	Pages	Rx	-	Reference Book
LCDP	-	Liquid Crystal Display Projector			

S.No	Lecture Hour	Topics to be covered	Teaching Aid Required	Book No /Page No
UNIT I - INTRODUCTION TO EMBEDDED SYSTEMS				
1	L1	Embedded system: Definition	BB	Tx1-pp(3-5) Rx1-pp(19-20)
2	L2	Categories	BB	Rx2-pp(22-25)
3	L3	Recent trends	BB	Rx2-pp(30-31)
4	L4	Design considerations	BB	Tx1-pp(37-43)
5	L5	Requirements	BB	Tx1-pp(37-43)
6	L6	Overview of architecture	BB	Rx1-pp(25-27)
7	L7	Purpose	BB	Tx1-pp(110-113)
8	L8	Major application area	BB	Rx1-pp(20-22)
9	L9	Design life cycle	BB,LCDP	https://www.globalspec.com/reference/28434/203279/chapter-1-the-embedded-design-life-cycle
UNIT II - EMBEDDED DEVICES AND BUSES				
10	L10	IO types and examples	BB	Tx1-pp(130-134)
11	L11	Serial communication devices	BB,LCDP	Tx1-pp(134-143)
12	L12	Parallel device ports - Sophisticated interfacing features in device ports	BB,LCDP	Tx1-pp(143-151)
13	L13	Wireless devices - Timer and counting devices	BB	Tx1-pp(151-157)
14	L14	Watchdog timer - Real time clock - Networked embedded systems	BB	Tx1-pp(157-160) Rx2-pp(71-72)
15	L15	Serial bus communication protocols	BB	Tx1-pp(160-165)
16	L16	Parallel bus device protocols	BB	Tx1-pp(166-170)

17	L17	Internet enabled systems - Wireless and mobile system protocols	BB	Tx1-pp(170-179)
18	L18	ISR concept - Interrupt sources - Interrupt service handling mechanism	BB	Tx1-pp(192-209) Rx2-pp(81-91)
UNIT III - EMBEDDED PROGRAMMING				
19	L19	Software programming in assembly and high level language	BB	Tx1-pp(235-237)
20	L20	Program elements: Macros and functions, data types	BB	Tx1-pp(239-242)
21	L21	Data structures, modifiers	BB	Tx1-pp(242-251)
22	L22	Statements, loops and pointers	BB	Tx1-pp(251-261)
23	L23	Object oriented programming - Embedded programming in C++	BB, LCDP	Tx1-pp(262-264)
24	L24	Embedded programming in JAVA - Program models - DFG models	BB, LCDP	Tx1-pp(264-269,274-282)
25	L25	State machine programming models for event controlled program flow	BB	Tx1-pp(282-288)
26	L26	Modeling of multiprocessor systems	BB	Tx1-pp(288-295)
27	L27	UML modeling.	BB	Tx1-pp(295-299)
UNIT IV - REAL TIME OPREATING SYSTEMS				
28	L28	Multiple processes, Threads in an application	BB	Tx1-pp(305-308)
29	L29	Tasks - Task states	BB	Tx1-pp(308-310) Rx2-pp(139-143)
30	L30	Semaphore	BB	Tx1-pp(314-326) Rx2-pp(153-167)
31	L31	Shared data - Interprocess communication - Signal function - Semaphore functions	BB,LCDP	Tx1-pp(326-335) Rx1-pp(191-194)
32	L32	Message queue functions - Mailbox functions - Pipe functions	BB,LCDP	Tx1-pp(335-341) Rx2-pp(173-184)
33	L33	Socket functions	BB,LCDP	Tx1-pp(341-345)
34	L34	OS services - Process management - Timer functions - Event functions	BB	Tx1-pp(351-359) Rx2-pp(184-195)
35	L35	Memory management - Device, File and IO subsystems management	BB	Tx1-pp(359-366) Rx2-pp(195-199)
36	L36	Interrupt routines in RTOS environment and handling of interrupt source calls.	BB	Tx1-pp(366-370) Rx2-pp(199-206)
UNIT V - CASE STUDIES				
37	L37	Types of RTOS - Introduction and features of MUCOS II	BB	Tx1-pp(408-410,410-453)
38	L38	VxWorks	BB	Tx1-pp(453-473)
39	L39	Windows CE	BB	Tx1-pp(478-494)
40	L40	OSEK	BB	Tx1-pp(494-496)
41	L41	Linux 2.6	BB	Tx1-pp(496-501)
42	L42	RTLinux	BB	Tx1-pp(501-505)
43	L43	Case studies: Automatic chocolate vending machine	BB, LCDP	Tx1-pp(512-522)
44	L44	Sending application layer byte streams on a TCP/IP network	BB, LCDP	Tx1-pp(537-544)
45	L45	Adaptive cruise control.	BB, LCDP	Tx1-pp(577-586)

K.S.R. COLLEGE OF ENGINEERING (Autonomous), TIRUCHENGODE – 637 215
DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
16EC613 – EMBEDDED SYSTEMS
QUESTION BANK

UNIT-I
INTRODUCTION TO EMBEDDED SYSTEMS

1. **What is an embedded system? (Remembering, CO1)**
   An embedded system is a system that has embedded software and computer hardware, which makes it a system dedicated for applications or specific part of an application or product or a part of a larger system.
2. **Mention the applications of embedded system. (Remembering, CO1)**
   Consumer appliances, industrial automation, medical electronics, computer networking, telecommunication, wireless technologies, instrumentation, security and finance.
3. **Classify the various categories of embedded system. (Understanding, CO1)**
   - Stand alone embedded system
   - Real time embedded system
   - Networked embedded system
   - Mobile embedded system
4. **Define stand alone embedded systems and give some examples. (Remembering, CO1)**
   It works in stand-alone mode. They take inputs, process them and produce the desired output.
   Examples: Digital camera, microwave oven, CD player, Air conditioner and TV, etc..,
5. **Define real time embedded systems and give some examples. (Remembering, CO1)**
   Embedded systems in which some specific work has to be done in a specific time period are called real-time systems.
   Examples: aircraft, missile launching, etc..,
6. **Classify the types of real time embedded systems. (Understanding, CO1)**
   Hard real-time system- Consider a system that has to open a valve within 30 milliseconds when the humidity crosses a particular threshold. If a valve is not opened within 30 milliseconds, a catastrophe may occur. Such systems with strict deadlines are called hard real-time systems.
   Soft real-time system- In some embedded systems, deadlines are imposed, but not adhering to them once in a while may not lead to a catastrophe. Consider a DVD player; you give a command to the DVD player from the remote control, and there is a delay of a few milliseconds in executing that command. But, this delay won't lead to a serious implication.
7. **Define networked embedded systems. (Remembering, CO1)**
   Embedded systems that are provided with network interfaces and accessed by networks such as Local Area Network or the Internet are called networked information appliances.
8. **Define mobile embedded systems. (Remembering, CO1)**
   Mobile devices such as mobile phones, Personal Digital Assistants (PDAs), smart phones etc., are a special category of embedded systems.
9. **Give some limitations of mobile embedded systems. (Understanding, CO1)**
   - Memory constraint
   - Small size
   - Lack of good user 9interfaces such as full-fledged keyboard and display
10. **List out the recent trends in embedded systems.(Analyzing, CO1)**
    Processor power, memory, operating systems, communication interfaces and networking capability, programming languages, development tools and programmable hardware.
11. **Mention some development tools in embedded systems. (Understanding, CO1)**
    - MATLAB and Simulink – used to model an embedded system as well as to generate code, substantially reducing the development time.

- BREW (Binary Runtime Environment for Wireless), java 2 Micro Edition (J2ME) development tools, wireless application Protocol (WAP) development tools facilitate easy development of applications for mobile devices.

12. **What are the steps or levels of abstraction involved in the embedded system design process? (Remembering, CO1)**
    Requirement, specification, architecture, component and system integration
13. **List the challenges faced in embedded computing system design? (Analyzing, CO1)**
    - Amount and type of hardware needed
    - Optimizing power dissipation and consumption
14. **What are the different methods used to meet the design challenges? (Remembering, CO1)**
    Clock rate reduction, voltage reduction, wait, stop and cache disable instructions, process deadlines, flexibility and upgradability, reliability.
15. **What is meant by firmware? (Remembering, CO1)**
    Every embedded system consists of custom-built hardware built around a central processing unit. This hardware also contains memory chips onto which the software is loaded. The software residing o the memory chip is also called the firmware.
16. **List out the various building blocks of the hardware of an embedded system. (Analyzing, CO1)**
    - Central Processing Unit
    - Memory (read only memory and random access memory)
    - Input devices
    - Output devices
    - Communication interfaces
    - Application specific circuitry
17. **Mention some communication interfaces in embedded system. (Understanding, CO1)**
    RS232, RS422, RS485, USB, IEEE 1394 and Ethernet etc,

PART –B

1. Explain the different categories of embedded systems with suitable examples. **(Understanding, CO1)**
2. Discuss in detail about recent trends in embedded systems. **(Creating, CO1)**
3. Explain in detail about the steps or levels of abstraction involved in the embedded system design process? **(Understanding, CO1)**
4. Discuss briefly about overview of embedded system architecture with suitable diagram. **(Creating, CO1)**
5. Mention the various applications of an embedded system and explain in detail. **(Remembering, CO1)**
6. Explain in detail about design life cycle in embedded system. **(Understanding, CO1)**

UNIT-II
EMBEDDED DEVICES AND BUSES

1. **Define port. (Remembering, CO2)**
   A port is a device to receive the bytes from external peripheral(s) [or device(s) or processor(s) or controllers] for reading them later using instructions executed on the processor to send the bytes to external peripheral or device or processor using instructions executed on processor.
   A Port connects to the processor using address decoder and system buses.
2. **Compare serial port and parallel port.(Evaluating, CO2)**

Serial port	Parallel port
It is a port for serial communication.	It is a port for parallel communication.

It means that over a given line or channel one bit can communicate and the bits transmit at periodic intervals generated by a clock.	It means that multiple bits can communicate over a set of parallel lines at any given instance.
It suit for over short or long distance.	It suit for communicate within the same board, between ICs or wires over very short distances of at most less than a meter.

3. **Mention the different types of serial and parallel ports of I/O devices. (Remembering, CO2)**

Serial ports

- Synchronous Serial Input
- Synchronous Serial Output
- Asynchronous Serial UART input
- Asynchronous Serial UART output (both as input and as output, for example, modem.)

Parallel ports

- Parallel port one bit Input
- Parallel one bit output
- Parallel Port multi-bit Input
- Parallel Port multi-bit Output

4. **Define synchronous serial input. (Remembering, CO2)**

The sender along with the serial bits also sends the clock pulses SCLK (serial clock) to the receiver port pin. The port synchronizes the serial data input bits with clock bits. Each bit in each byte as well as each byte in synchronization

Synchronization means separation by a constant interval or phase difference. If clock period = T, then each byte at the port is received at input in period = 8T.

5. **Mention the different names for synchronous serial input. (Remembering, CO2)**

Master output slave input (MOSI)

MOSI when the SCLK is sent from the sender to the receiver and slave is forced to synchronize sent inputs from the master as per the inputs from master clock.

Master input slave output (MISO)

MISO when the SCLK is sent to the sender (slave) from the receiver (master) and slave is forced to synchronize for sending the inputs to master as per the master clock outputs.

6. **Define synchronous serial output. (Remembering, CO2)**

Each bit in each byte sent in synchronization with a clock. Bytes sent at constant rates. If clock period= T, then data transfer rate is (1/T) bps. Sender either sends the clock pulses at SCLK pin or sends the serial data output and clock pulse-input through same output line with clock pulses either suitably modulate or encode the serial output bits.

7. **How the port can be used in synchronous serial output? (Remembering, CO2)**

The processing element at the port (peripheral) sends the byte through a shift register at the port to where the microprocessor writes the byte.

8. **Define asynchronous serial input and output. (Remembering, CO2)**

<u>Asynchronous serial input</u>

It is denoted by RxD (Receive Data). Each RxD bit is received in each byte at fixed intervals but each received byte is not in synchronization. Bytes separate by the variable intervals or phase differences.

Asynchronous serial input also called UART input if serial input is according to UART protocol.

<u>Asynchronous serial output</u>

Asynchronous output serial port line TxD (transmit data). Each TxD bit in each byte transmits at fixed intervals but each output byte is not in synchronization (separates by a variable interval or phase difference). Minimum separation is 1 stop bit interval TxD.

Synchronous serial output is also called UART output if serial output is according to UART protocol.

9. Define parallel port. (Remembering, CO2)

A parallel port can have one or multibit input or output and can be bi-directional I/O.

- One bit input, output and I/O
- Eight or more bit input, output and I/O

10. Distinguish between half duplex and full duplex.(Analyzing, CO2)

Half duplex	Full duplex
Half duplex means that at any point communication can only be one way (input or output) on a bi-directional line.	Full duplex means that the communication can be both ways simultaneously.
An example of half-duplex mode — telephone communication.	An example of the full duplex asynchronous mode of communication is the communication between The modem and the computer though TxD and RxD lines

11. Classify the types of communication ports for I/O.(Understanding, CO2)

There are two types of communication ports for IOs:

- Serial line port communication and
- Parallel line port communication

12. Define serial communication devices. (Remembering, CO2)

Serial line port communication is synchronous when a clock of the master device controls the synchronization of the bits on the line.

It is asynchronous when clocks of the sender and receiver are independent and bytes are received, not necessarily at constant phase difference.

Serial communication can be full duplex, which means simultaneously communication both ways, or half duplex, which means one way communication.

13. Mention the examples of various types of IO devices. (Remembering, CO2)

IO device type	Examples
Serial Synchronous input	Inter processor data transfer, Reading from CD, Audio input, Video input, Dial tone, Network input, etc.,
Serial Synchronous output	Inter processor data transfer, Multiprocessor communication, Writing to CD, Audio output, Video output, Dialer output, Network output, etc.,
Serial asynchronous input	Keypad controller serial data in, Mice, Keyboard controller data in, Modem input, Character inputs on serial line
Serial asynchronous output	Output from modem, Output for printer, Output on a serial line
Parallel port one bit Input	Completion of a revolution of a wheel, Filling a liquid up to a fixed level, Achieving presser pressure in a boiler, Weight over a pan of an electronic balance, etc.,
Parallel one bit output	PWM output for a DAC-which controls liquid level, temperature, pressure, speed etc., and pulses to an external circuit.
Parallel Port multi-bit Input	ADC input from liquid level measuring sensor or temperature sensor or pressure sensor and encoder input for bits for angular position.
Parallel Port multi-bit Output	LCD controller for multiline LCD display matrix unit, Print controller, Stepper Motor

14. Mention the different ways of communication between the serial communication ports or devices. (Remembering, CO2)

- Synchronous communication
- Iso-synchronous communication

- Asynchronous communication

15. Define synchronous communication. (Remembering, CO2)

When a byte (character) or a frame (a collection of bytes) in of the data is received or transmitted at the constant time intervals with uniform phase differences, the communication is called as **synchronous communication.**

16. List out the characteristics of synchronous communication.(Analyzing, CO2)

i.) Bytes (or frames) maintain a constant phase difference, which means they are synchronous, i.e. in synchronization. There is no permission for sending either the bytes or the frames at random time intervals, this mode therefore does not provide for handshaking during the communication interval.

ii.) A clock ticking at a certain rate has always to be there for transmitting serially the bits of all the bytes (or frames) serially. Mostly, the clock is not always implicit to the synchronous data receiver. The transmitter generally transmits the clock rate information.

17. Mention the various synchronous device port bits. (Remembering, CO2)

- Sync code bit or bi-sync code bits or frame start and end signaling bits
- Data bits
- Clock bits

18. List out the common methods of encoding the clock information into a serial stream of bits in synchronous communication devices. (Analyzing, CO2)

- Frequency Modulation (FM)
- Mid Frequency Modulation (MFM)
- Manchester Coding
- Quadrature Amplitude Modulation (QAM)
- Bi-phase coding

19. Define asynchronous communication. (Remembering, CO2)

When a byte (characters) or frame (a collection of bytes) of data is received or transmitted at variable time intervals, communication is called **asynchronous.**

20. List out the characteristics of asynchronous communication. (Analyzing, CO2)

i.) Bytes (or frames) need not maintain a constant phase difference and are asynchronous i.e., not in synchronization. Bytes or frames can be sent at variable time intervals.

ii.) Though the *clock* must tick at a certain rate to transmit the bits of a single byte (or frame) serially, it is *always implicit* to the asynchronous data receiver.

21. Mention the some examples for asynchronous communication. (Remembering, CO2)

- RS232C communication between the UART devices
- UART communication for asynchronous data is used for the transfer of information between the keypad or keyboard and computer.

22. List out the important protocols for synchronous or asynchronous transmission from a device port. (Analyzing, CO2)

RS232C, UART, HDLC, X.25, Frame relay, ATM, DSL and ADSL

23. List out the various types of internet enabled embedded systems for asynchronous communication. (Analyzing, CO2)

- HTTP (Hyper Text Transfer Protocol),
- HTTPS (Hyper Text Transfer Protocol Secure Socket Layer),
- SMTP (Simple Mail Transfer Protocol),
- POP3 (Post Office Protocol version 3),
- ESMTP (Extended SMTP),
- TELNET (Tele network),
- FTP (File Transfer Protocol),
- DNS (Domain Network Server),

- IMAP 4 (Internet Message Exchange Application Protocol) and
- Bootp (Bootstrap protocol)

24. Define RS232C and RS485 communication. (Remembering, CO2)

RS232C communication – It is interfacing signal standard between DTE (computer) (Data Terminal Equipment) port and DCE (modem) (Data Communication Equipment).

RS483 communication – It is a protocol for physical layer in case of two wires full or half duplex serial connection between multiple points.

25. Define HDLC protocol. (Remembering, CO2)

High Level Data Link Control (HDLC) is an international Standard protocol for a data link network. It is used for linking data from point to point and between multiple points.

It is a bit oriented protocol. The total number of bits is not necessarily an integer multiple of a byte or a 32-bit integer.

26. Mention the format of bits in Synchronous HDLC protocol based network device. (Remembering, CO2)

- Frame start and end signaling flag bit
- Address bits for destination
- Control bits case 1- information frame
- Control bits case 2 – supervisory frame
- Control bits case 3 – un-numbered frame
- Data bits
- FCS (Frame Check Sequence) bits
- Frame end flag bits

27. Expand and define SPI, SCI and SI port.(Understanding, CO2)

SPI – Synchronous Peripheral Interface

It is programmable for defining the occurrence of negative and positive edges within an interval of bits at serial data out or in.

SCI – Serial Connect Interface

It is a UART asynchronous mode port. Communication is in full duplex mode for the SCI transmissions and receiver.

SI – Serial Interface

It is a UART mode asynchronous port interface and also a function as USRT (Universal Synchronous Receiver and Transmitter).

28. What are the features of SPI? (Remembering, CO2)

- SPI has programmable clock rates
- Full-duplex mode
- Crystal clock frequency is 8MHz
- Open drain or totem pole output from master to slave

29. What are the different modes of operations in Serial Interface port? (Remembering, CO2)

There are two modes of operation

i.) Half duplex synchronous mode of operation (called mode 0)

ii.) Full duplex synchronous serial communication (called mode 1 or 2 or 3)

30. Define SDIO. (Remembering, CO2)

SDIO – Secure Digital Input Output

It is an SD card with programmable IO functionalities such that

(i) It can be used upto eight logical functions

(ii) It can provide additional memory storage in SD format

(iii) It can provide IOs using protocols in systems such as IrDA adapter, UART, GPS, etc.,

31. Define parallel device ports. (Remembering, CO2)

It can have parallel inputs, parallel outputs, bi-directional and quasi bi-directional IOs. A parallel device port can have handshaking pins. A parallel device port can also have control pins

for control-signal outputs to external circuit and status pins for inputs of status signals to external circuits.

32. List out the characteristics during interfacing a parallel device port.(Analyzing, CO2)

- Multi byte data input and output buffers
- Data Direction Register (DDR)
- LSTTL driving capability and port loading capability
- Quasi bi directional
- Multiple or alternate functionality in port pins
- Provision for multiplexed output to multiple system or units
- Provision for demultiplexed inputs from multiple systems or units

33. List out various applications of interfacing a parallel device port. (Analyzing, CO2)

- Parallel port interfacing with switches and keypad
- Parallel port interfacing with encoders
- Parallel port interfacing with stepper motor
- Parallel port interfacing with LCD controller
- Parallel port interfacing with touchscreen

34. What are the features for interfacing a parallel device port? (Remembering, CO2)

- Operation voltage level expected for logic state 1=5V
- When a device port is waiting for instruction, power management can be done at the gates of the device.
- Post interface can be used to be either open drain CMOS or TTL
- Device connects to a system bus and also an IO bus
- IO device may consist of multiple gigabit transceivers
- IO may integrate a SerDes (Serialization and de-serialization) subunit
- Multiple IO standards are developed

35. List out the various wireless devices. (Analyzing, CO2)

IrDA, Bluetooth, Wifi, 802.11 WLAN, ZigBee

36. Define timer and counting devices. (Remembering, CO2)

Timer device - Timer is a device, which counts the input at regular interval (δT) using clock pulses at its input.

Counting device – It is a device that counts the input for events that may occur at irregular or regular intervals.

37. What is meant by blind counting synchronization or free running counter? (Remembering, CO2)

A counting device may be a free running(blind counting) device with a prescaler for the clock input pulses and for comparing the counts with the ones preloaded in a compare register.

It is useful for action or initiating chain of actions and processor interrupts at the preset instances as well as noting the instances of occurrences of the events and processor interrupts for requesting the processor to use the capturing of counts on the events for future actions.

38. Define watchdog timer. (Remembering, CO2)

It is a timing device that can be set for a preset time interval, and an event must occur during that interval else the device will generate the timeout signal.

39. Mention the application of watchdog timer. (Remembering, CO2)

- In a mobile phone is that the display is turned off in case of no GUI interaction takes place within a specified time.
- In a mobile phone is that if a given menu is not selected by a click within a preset time interval, another menu can be presented or a beep can be generated to invite user's attention.

- In a temperature controller is that if a controller takes no action to switch off the current within the preset time, the current is switched off and a warning signal raised, indicating controller failure.

40. Define Real Time Clock (RTC). (Remembering, CO2)

It is a clock that causes occurrences of regular interval interrupts on its each tick (time out). An interrupt service routine executes on each timeout (overflow) of this clock. This timing device once started is generally never reset or never reloaded to another value.

41. List out the some uses of Real Time Clock.(Analyzing, CO2)

- Used in a system to save the current time and date.
- Used in a system to initiate return of control to the system (OS) after the preset system clock periods.

42. List out the various types of buses used in serial bus communication protocols. (Analyzing, CO2)

I^2C bus- ICs mutually network through a common synchronous serial bus

CAN bus –serial bus for interconnecting a central control network, it is used in automobiles.

USB bus–It is a bus between host system and number of interconnected peripheral devices.

FireWire-IEEE 1394 Bus – It is a high speed 800 Mbps serial bus interconnecting a system with multimedia streaming device and system.

43. What are the three I^2C bus standards? (Remembering, CO2)

- Industrial 100 kbps I^2C,
- 100 kbps SM I^2C,
- 400 kbps I^2C

44. Mention the disadvantages of I^2C bus. (Remembering, CO2)

- Time taken by algorithm in the hardware that analyzes the bits throughI^2C in case the slave hardware does not provide for the hardware that supports it.
- Certain ICs support the protocol and certain do not.
- Open collector drivers at the master need a pull-up resistance of 2.2 K on each line.

45. What are the features of USB Bus? (Remembering, CO2)

- Can be hot plugged (attached), configured and used, reset, reconfigured and used
- Bandwidth sharing with other devices: Host schedules the sharing of bandwidth among the attached devices at an instance.
- Can be detached (while others are in operation) and reattached.
- Attaching and detaching USB device or host without rebooting.

46. What are the three USB bus standards? (Remembering, CO2)

- USB 1.1 (a low speed1.5 Mbps 3 meter channel along with a high speed 12 Mbps 25 meter channel),
- USB 2.0 (high speed 480 Mbps 25meter channel), and
- Wireless USB (high speed 480 Mbps 3m)

47. How different types of pipes that supported the USB bus? (Remembering, CO2)

Stream - with no USB- defined protocol. It is used when the connection is already established and the data flow starts

Default Control - for providing access.

Message - for the control functions for of the device.

48. List out the various advanced serial high speed buses. (Analyzing, CO2)

- IEEE 802.3 – 2000 (1Gbps)
- IEE P802.3oe draft 4.1 (10 Gbps)
- IEE P802.3oe draft 4.1 (12.5 Gbps)

- XAUI (10 Giga bit)
- XSBI

49. What are the different types of parallel bus device protocols? (Remembering, CO2)

- **ISA Bus** – It connects only to an embedded device that has an 8086 or 80186 or 80286 processor, and in which the processor addressing and IBM PC architecture addressing limitations and interrupt vector address assignments are taken into account. There is no geographical addressing.
- **PCI and PCI/X Buses** – It is popular in distributed embedded devices. PCI and PCI/X buses are used for parallel bus communication and these are independent from the IBM architecture. It is an extension of PCI and supports 64/100MHz transfers. New version supports 132/528MB/s data transfer with synchronous throughputs.
- **ARM Bus –** It interfaces the memory, external DRAM and on chip IO devices, which connect to 32-bit data and 32-bit address line at high speed using AMBA (ARM Main Memory Bus Architecture) – AHB (ARM High Performance Bus).

50. Define EISA. (Remembering, CO2)

EISA – Extended ISA Bus

EISA bus is a 32-bit data and address version of ISA, and devices (system using this bus for IOs) are also supported. An EISA device driver first checks the availability on the hosting computer or IO addresses or systems. It supports the sharing of interrupt functions.

51. What are the different types of ARM bus? (Remembering, CO2)

- AMBA – AHB - connects to high speed memory
- AMBA – APB – connects the external peripherals to the system memory bus through a bridge.

52. List out the various advanced parallel high speed buses.(Analyzing, CO2)

- GMII (Gigabit Ethernet MAC Interchange Interface)
- XGMI (10 Gigabit Ethernet MAC Interchange Interface)
- CSIX-1 6.6 Gbps
- Rapid IO interconnect specification v1.1 at 8 Gbps

53. Mention the types of internet enabled systems or network protocols. (Remembering, CO2)

- HTTP (Hyper Text Transfer Protocol)
- TCP (Transport Control Protocol)
- UDP (User Datagram Protocol)
- IP (Internet Protocol)
- Ethernet

54. What are the features of HTTP? (Remembering, CO2)

- Requesting for a URL address
- Stateless protocol,
- File transfer like protocol for HTML files,
- Very light and speed compared to existing protocols,
- Flexibility

55. Compare TCP and UDP.(Understanding, CO2)

TCP	UDP
Message is virtual connection oriented	Message is connection less and stateless.
It does not permit multicasting but point to point connection.	It supports the broadcast networking mode

56. What are the features of Ethernet LAN? (Remembering, CO2)

- Used for local network of computers, workstation and devices.
- It supports for sharing local computers, systems and local resources such as printer, hard disk, etc.,

- Speed is 10Mbps, 100Mbps for unshielded or shielded wires and 4 Gbps for twisted pair mode
- Broadcast in medium

57. Categorize the various types of wireless and mobile system protocols.(Analyzing, CO2)

- IrDA (Infrared Data Association)
- Bluetooth
- Wireless LAN Uses IEEE standard 802.11
- ZigBee

58. What is the use of interrupt service routines or device drivers? (Remembering, CO2)

It is used for the declaration of functions and data types, type def and executes named set of codes.

ISR must be small (short), reentrant or must have solution for shared data problem.

59. Define hardware interrupt. (Remembering, CO2)

When a device or port is ready, a device or port generates an interrupt or when it completes the assigned action, it generates an interrupt. This interrupt is called as hardware interrupt.

60. Define software interrupt or trap or exception. (Remembering, CO2)

When software run-time exception condition is detected, either processor hardware or a software instruction generates an interrupt. This interrupt is called as software interrupt.

61. What are the features of ISR call? (Remembering, CO2)

- An ISR call due to interrupt executes an event. Event can occur at any moment and event occurrences are asynchronous.
- ISR call is event based diversion from the current sequence of instructions to another sequence of instructions. This sequence of instructions executes till the return instruction.
- Interrupt service mechanism exists in a system to call ISR from multiple sources.
- Event signaling by software interrupt instruction SWI used in device drivers functions create(), open(), etc.,

62. Define device drivers and its generic functions. (Remembering, CO2)

- It is a function used by a high level language programmer and does the interaction with the device hardware and communicates data to the device, sends control commands to the device and runs the code for reading the device data.
- The generic functions used for the commands is create(),open(),connect(), bind(),read(), write(), ioctl(), delete(), close().

63. Mention the various interrupt sources.(Analyzing, CO2)

- There are number of **hardware interrupt sources** which can interrupt an ongoing program. These are processor or microcontroller or internal device hardware specific.
- There can be **external hardware interrupt source** for interrupting an ongoing program that also provides the ISR address or vector address or interrupt type information through the data bus.

64. Define interrupt vector. (Remembering, CO2)

It is a memory address to which the processor vectors. The processor transfers the program counter to the interrupt vector new address on an interrupt, using this address, the processor services that interrupt by executing corresponding ISR.

65. Classify the types of interrupts.(Understanding, CO2)

There are two types of interrupt. They are,

- **Non Maskable interrupt**–Examples are RAM parity error in a PC and error interrupts like division by zero.
- **Maskable interrupt** – Maskable interrupts are those for which the service may be temporarily disabled to let higher priority ISRs be executed first uninterruptedly.
- **Non-Maskable only when defined so within few clock cycles after reset**

PART –B

1. Explain in detail about various Input and output types with example. **(Understanding, CO2)**
2. Explain the synchronous and asynchronous communications from serial devices. **(Understanding, CO2)**
3. Explain the various parallel device ports in detail. **(Understanding, CO2)**
4. Explain the sophisticated interfacing features in device ports. **(Understanding, CO2)**
5. Write short notes on wireless devices. **(Remembering, CO2)**
6. Explain in detail about various timing and counting devices. **(Understanding, CO2)**
7. Describe watch dog timer and real time clock. **(Remembering, CO2)**
8. Explain in detail about various functions in networked embedded systems.**(Understanding, CO2)**
9. Discuss briefly about serial bus communication protocols. **(Creating, CO2)**
10. Explain serial communication using I^2C, CAN and USB bus in detail. **(Understanding, CO2)**
11. Explain various types of parallel bus device protocols in detail. **(Understanding, CO2)**
12. Explain in detail about various network protocols in internet enabled systems. **(Understanding, CO2)**
13. Discuss in briefly about wireless and mobile system protocols with suitable examples.**(Creating, CO2)**
14. Describe about ISR concept in detail. **(Remembering, CO2)**
15. Explain in detail about various sources in ISR. **(Understanding, CO2)**
16. Explain various types of mechanisms in interrupt servicing handling process in detail. **(Understanding, CO2)**

UNIT III
EMBEDDED PROGRAMMING
PART -A

1. **What are the advantages of Assembly language? (Remembering, CO3)**
    - It gives the precise control of the processor internal devices and full use of processor specific features in its instruction sets and addressing modes.
    - The machine codes are compact, which requires only small memory.
    - Device drivers need only few assembly instructions.
    - Bottom-up-design approach
2. **What are advantages of high level languages? (Remembering, CO3)**
    - High level program development cycle is short for complex systems
    - Facilitates Data type declaration
    - Facilitates Type checking making the program less prone to error
    - Facilitates use of Control structures
    - Portability of non-processor specific codes
3. **Define In -line assembly. (Remembering, CO3)**

    Inserting an assembly code in between is said to be in-line assembly.
4. **Mention the elements of C program. (Understanding, CO3)**
    - Files:
        - Header files
        - Source files
        - Configuration files
        - Preprocessor directives
    - Functions:
        - Macro function

- Main function
- Interrupt service routines or device drivers

- Others:
  - Data types
  - Data structures
  - Modifiers
  - Statements
  - Loops and pointers

5. **What is meant by preprocessor macros? (Remembering, CO3)**

   A macro is a collection of codes that is defined in a program as preprocessor directive.

   Differs from a function in the sense that once a macro is defined by a name, the compiler puts the corresponding codes at the macro at every place where that macro-name appears.

6. **Difference between macros and functions.(Understanding, CO3)**

   Macros:

   - The code for macro are compiled in every function wherever that macro name is used, as a compiler, before compilation, puts the codes at the places wherever the macro is used.
   - Macros are used for short codes only.

   Functions:

   - The codes for a function compiled once only
   - On calling that function, the processor has to save the context, and on return restore the context.
   - When a function call is used instead of macro, the overheads (context saving and return) will take a time, $T_{overheads}$ that is the same order of magnitude as the time, T_{exec} for execution of short codes within a function.
   - Use the function when the $T_{overheads} << T_{exec}$ and macro when $T_{overheads} \sim= \text{ or } > T_{exec}$.

7. **What are the data types available in C language? (Remembering, CO3)**

   Char - 8 bit for characters; byte - 8 bit; short - 16 bit;
   unsigned short - 16 bit; unsigned int - 32 bit; int - 32 bit;
   long double - 64 bit; float - 32 bit; double – 64

8. **List out the various actions of modifiers.(Analyzing, CO3)**

   (i)Auto (ii)unsigned (iii)static (iv)const (v)register
   (vi)interrupt (vii)extern (viii)volatile (ix)volatile static

9. **What is the use of loop? (Remembering, CO3)**

   Loops are used when executing a set of statements repeatedly. A loop starts from an initial value or condition and executes till the limiting condition is fulfilled.

   For example, **for (i = 0; i<= 100; i++)**

10. **Mention the use of infinite loops. (Understanding, CO3)**

    Never desired in usual programming. Because, the program will never end and never exit or proceed further to the codes after the loop.

    Infinite loop is a feature in embedded system programming.

    Example: A telephone is never switching off.

    The system software in the telephone has to be always in awaiting loop that finds the ring on the line. An exit from the loop will make the system hardware redundant.

11. **What is meant by pointer and null pointer and its syntax? (Remembering, CO3)**

    Pointer – It is a powerful tool when used correctly and according to certain basic principles. Pointer is a reference to a starting memory address. A pointer can refer to a variable, data structure or function. Before a pointer, in C language symbol * is used.

    Example: unsigned char *0x1000 -------character of 8 bits at address 0x1000

    Null Pointer - Free the memory spaces allotted to a data structure.

Example:#define NULL (void*) 0x0000----we can assign any address instead of 0x0000 that is not in use in a given hardware.

12. Mention the data structures available in C language.(Understanding, CO3)

- Queue
- Stack
- Array (1-dimentional and multi-dimensional)
- List
- Table and Hash table
- Pipe

13. Define stack. (Remembering, CO3)

A stack means an allotted memory block data from which a data element is read in a LIFO (Last In First Out) way and an element is popped or pushed from an address pointed by a pointer, called SP (Stack Pointer) or S_{top} and SP changes on each push or pop such that it points to the top of stack.

14. Classify the various pointers in multiple stacks.(Analyzing, CO3)

- RIP (Return Instruction Pointer)
- SP (Stack Pointer)
- FP (Data Frame Pointer)
- PFP (Previous Program Frame Pointer)

15. Define array. (Remembering, CO3)

An array has multiple data elements, each identifiable by an index or by a set of indices and indices are unsigned integers. An n-dimensional array is a special data structure at the memory.

16. Define queue and its operations. (Remembering, CO3)

A queue means an allotted memory block from which a data element retrieved in the FIFO (First In First Out) mode. Using the queues, the bytes are sent onto a memory buffer or network or printer.

- **Insertion** – In queue, it is written to an address next to the address from where the queue element was last written. This writing is called **insertion.**
- **Deletion** –In queue, each element is read from an address next to the address from where the queue element was last read. This reading is called **deletion.**

17. Define circular queue. (Remembering, CO3)

A queue is called circular queue when a pointer on reaching a limit Q_{limit}, return to its starting value $*Q_{start}$.

A circular queue is a queue in which tail and head pointers cannot increment beyond then memory block (buffer) and reset to the starting value on insertion beyond the boundary.

18. Define pipe. (Remembering, CO3)

A pipe is a device, which uses device driver functions and in which insertions are from the source end and deletions are at sink – end.

19. Define table. (Remembering, CO3)

A table is a two-dimensional structure (matrix) and is an important data set that is allocated a memory block. There is always a base pointer for a table. There are two indices, one for a column and other for a row.

20. What is meant by lookup table? (Remembering, CO3)

A lookup table is a two-dimensional structure (matrix) and is an important data set. It has only rows and each row has a key and on reading the key, the addressed data is traced.

21. Define hash table. (Remembering, CO3)

A hash table is a data set that is a collection of pairs of key and corresponding value. A hash table has a key or name in one column. The corresponding value or object is at the second column. The keys may be at non-consecutive memory addresses.

22. Define list. (Remembering, CO3)

A list is a data structure with a number of memory blocks, one for each element. A list has a top (head) pointer for the memory address from where it starts. Each list element at the memory also stores the pointer to the next element. The last element points to null.

23. What is Object oriented programming? (Remembering, CO3)

An object-oriented programming language is used when there is a need for re-usability of defined objects or a set of objects that are common for many applications.

24. What are the advantages of OOPs? (Remembering, CO3)

- Data encapsulation
- Reusable software components
- Inheritance
- Creation of multiple instances of the defined object

25. What are the characteristics of OOPs? (Remembering, CO3)

- An identity - reference to a memory block that holds its state and behavior.
- A state - data, field and attributes
- A behavior - methods to manipulate the state of the object.

26. Define class and object. (Remembering, CO3)

Class - A class declaration defines a new type that links code and data. It is then used to declare objects of that class. Thus a class is a logical abstraction but an object has physical existence.

Object – Object is functional. Each object can interact with other objects to process the users' data. An object is an instance of a class.

27. What are the disadvantages of embedded programming in C++?(Remembering, CO3)

Lengthier Code when using Template, Multiple Inheritance (Deriving a class from many parents), Exceptional handling, Virtual base classes and classes for IO Streams.

28. List out the various elements in a Java program.(Analyzing, CO3)

- Local variable
- Class
- Inheritance
- Interface
- Instance method
- Data types
- Exception

29. What are the advantages of Java Programming? (Remembering, CO3)

- Java is completely an OOP language. Java starts with classes.
- There is a huge class library on the network that makes program development quick.
- Java has extensibility.
- Java is easier to learn by a c++ programmer.

30. What are the disadvantages of Java Programming? (Remembering, CO3)

- As java codes are first interpreted by the JVM, it runs comparatively slowly.
- Java byte codes that are generated need a larger memory.

31. List out the features of KVM. (Analyzing, CO3)

- Use of following data types is optional
- Errors are handled by the program classes, which inherit only a few needed error handling classes from the java.
- There is no verification of the classes.
- There is no object finalization.
- Thread groups are not available.

32. Define J2ME. (Remembering, CO3)

J2ME –Java 2 Micro Edition

- It is used to helps in reducing the code size to 8kb for the usual application.

33. Define MIDP. (Remembering, CO3)

MIDP – Mobile Information Device Profiler

- It is profiler class for mobile devices. A profile defines the support of java to a device family. It is a layer between the application and the configuration.

34. What are the features of MIDP? (Remembering, CO3)

- Touch screen or keypad
- Minimum of 96 X 54 pixel colour or monochrome display
- Wireless networking
- Minimum of 32kb RAM, 8kb EEPROM or flash for data and 128kb ROM
- MIDP uses as in PDAs, mobile phones and pagers.

35. List out the various types of models in program model.(Analyzing, CO3)

- **Polling for events model**-cyclic loop for the event, state variable, messages and signals using switch case statement
- **Sequential program model** – there are sequential multiple function calls within a function.
- **Dataflow model** – consists of DFG and control data flow graph (CDFG).
- **State machine model** - there are different states and the model considers a system as a machine, which is producing the states.
- **Concurrent processes and interprocess communication model** – there are several concurrent tasks and each task has the sequential codes in infinite loop.

36. How does data flow in a program? (Remembering, CO3)

- Data that is input after the operations in the program becomes data that is output after a data flow. A diagram called the DFG represents this graphically.

37. What is meant by DFG? (Remembering, CO3)

- A diagram called the DFG represents this graphically. A DFG does not have any conditions within it so that the program has one data entry point and one data output point. There is only one independent path for the program flow when the program is executed.

38. Give some examples for non-acyclic data input.(Understanding, CO3)

- Event
- Status flag setting in a device
- Input as per output condition of the previous process.

39. What is meant by CDFG model? (Remembering, CO3)

CDFG – Control Data Flow Graph

- The program determines all program execution steps and the flow of a program. It is a diagram which graphically represents the conditions and the program flow along a condition dependent path.

40. What is meant by SDFG? (Remembering, CO3)

SDFG – Synchronous Data Flow Graph

- When there are number of tokens (inputs) required for a computation to generate more tokens (outputs) in a single firing.

41. Define state machine and state transition. (Remembering, CO3)

State machine – It is a model in which it is assumed that there are states and state transition functions, which produces the states.

State transition – It is a function which changes a state to its next state.

42. Define FSM model. (Remembering, CO3)

FSM – Finite State Machine

- FSM model states that there is finite number of possible states in a system and a system can only exist in one of these states at an instance.

43. Define FSM state table and mention its components. (Remembering, CO3)

FSM state table – State table can be designed for representation of every state in its rows.

Components: Present state, Action, events, outputs, Next state, Expected time interval.

44. Define multiprocessor system. (Remembering, CO3)

A large complex program can be partitioned into the tasks or sets of instruction (or processes or threads) and the ISRs. The tasks and ISRs can run concurrently on different processors and by some mechanism the tasks can communicate with each other.

45. Define UML language. (Remembering, CO3)

UML – Unified Modeling Language

- It is a language for modeling and any general system for which object oriented analysis and design are feasible and which can be abstracted by models.

46. Mention the different elements used in UML language.(Understanding, CO3)

- Class, Package, Stereotype, Object, Anonymous object and State.

47. Mention the different types of UML diagrams. (Understanding, CO3)

State diagram – shows a model of a structure for its state, end in between associations through the transitions and shows event labels with associated transitions.

Class diagram–shows how the classes and objects of a class relate hierarchical associations and object interactions between the classes and objects.

Object diagram– Defines the static configuration of the systems. It also gives the relationship among the consequent objects.

Sequence diagram– shows visualizes the interactions between the objects. Sequence diagrams also specify the sequences of states.

Collaboration diagram – It visualizes the concurrent sequences of states or object interactions.

PART –B

1. Explain in detail about advantages of Software programming in assembly and high level language. **(Understanding, CO3)**
2. Explain briefly about various program elements. **(Understanding, CO3)**
   (i) Macros
   (ii) Functions
3. Explain in detail about use of data types and data structures in software programming. **(Understanding, CO3)**
4. What are the actions taken uses of modifiers for software programming? Explain in detail. **(Remembering, CO3)**
5. Explain briefly about use of loops and pointers with suitable examples. **(Understanding, CO3)**
6. Write short notes on object oriented programming? **(Remembering, CO3)**
7. Explain in detail about advantages and disadvantages of embedded programming in C++. How it can be overcome? **(Understanding, CO3)**
8. Explain briefly about embedded programming in JAVA. Mention its advantages and disadvantages. **(Understanding, CO3)**
9. Write short notes on J2ME, JavaCard and Embedded Java? **(Remembering, CO3)**
10. What are the various types of program models and explain briefly. **(Remembering, CO3)**
11. Explain in detail about DFG model with suitable example. **(Understanding, CO3)**.
12. Explain briefly about State machine programming models for event controlled program flow. **(Understanding, CO3)**
13. Discuss in detail about Modeling of multiprocessor systems. **(Creating, CO3)**
14. Describe in detail about UML Modeling with suitable diagrams. **(Remembering, CO3)**

UNIT – IV
REAL TIME OPREATING SYSTEMS
PART –A

1. **Define process. (Remembering, CO4)**

   Process is defines as a computational unit that processes on a CPU and whose state changes under the control of kernel of an OS. It has state, which at an instance defines by the process status.

2. **Define PCB. (Remembering, CO4)**

   PCB – Process Control Block

   It is a data structure having the information using which the OS controls the process state. The PCB stores in the protected memory addresses at kernel.

3. **Define thread. (Remembering, CO4)**

   A thread defines a minimum unit of a multithreaded process that an OS schedules onto the CPU and allocates other system resources.

4. **What is meant by multiple threads? (Remembering, CO4)**

   A multiprocessing OS runs more than one processes. When a process consists of multiple threads, it is called multithreaded process.

5. **How does a task differ from a thread? (Remembering, CO4)**
   - Thread is a concept used in Java or UNIX.A thread can either be a sub-process within a process or a process within an application program. To schedule the multiple processes, there is the concept of forming thread groups and thread libraries.
   - A task is a process and the OS does the multitasking. Task is a kernel-controlled entity while thread is a process-controlled entity.

6. **Define RTOS. (Remembering, CO4)**

   A real-time operating system (RTOS) is an operating system that has been developed for real time applications. It is typically used for embedded applications, such as mobile telephones, industrial robots, or scientific research equipment.

7. **Define task and task states. (Remembering, CO4)**

   Task –It is defined as embedded program computational unit that runs on a CPU under the state control of kernel of an OS. It has a state, which at an instance defines by status (running, blocked or finished).

   Task state – It includes its status at a given instance in the system.

8. **List out the various types of task states. (Analyzing, CO4)**
   - Idle (created) state
   - Ready (active) state
   - Running state
   - Blocked (waiting) state
   - Deleted (finished) state

9. **Define Task Control Block (TCB). (Remembering, CO4)**

   TCB is a memory block. It is a data structure having the information using which the OS controls the task state. The TCB stores in the protected memory area of the kernel. The TCB consists of the following information about the task.

10. **What is meant by context and context switching? (Remembering, CO4)**
    - **Context** –Each task has a context CPU registers and parameters, which includes registers for the task PC and pointer to the called function stack top. This reflects the CPU state just before the OS blocks one task and initiates another task into the running state. The context thus continuously updates during the running of a task, and the context is saved before switching occurs to another task.

- **Context switching** – Only after saving these registers and pointers does the CPU control switch to any other process or task. The context must retrieve on transfer of program control to the CPU back for running the same task again, on the OS unblocking its state and allowing it to enter the running state again. The context switching action must happen each time the scheduler blocks one task and runs another task.

11. Define Semaphore. (Remembering, CO4)

Semaphore provides a mechanism to let a task wait till another finishes. It is a way of synchronizing concurrent processing operations. When a semaphore is taken by a task then that task has access to the necessary resources. When given the resources unlock. Semaphore can be used as an event flag or as a resource key.

12. Differentiate counting semaphore and binary semaphore. (Understanding, CO4)

- **Binary semaphore -** When the value of binary semaphore is one it is assumed that no task has taken it and that it has been released. When the value is 0 it is assumed that it has been taken.
- **Counting semaphore -** Counting semaphore is a semaphore which can be taken and given number of times. Counting semaphores are unsigned integers.

13. Define Mutex. (Remembering, CO4)

When a binary semaphore is used to at beginning and end of critical sections in two or more tasks such that at any instance only one section code can run, then the semaphore is called Mutex.

14. Define multiple semaphores. (Remembering, CO4)

Multiple semaphores are used and different set of semaphores can share among different set of tasks. It can be used in multitasking system.

15. List out the types of synchronous mechanism in semaphore. (Analyzing, CO4)

- **P semaphores** – P for wait operation. P semaphore function signals that the task requires a resource and if not available waits for it.
- **V semaphores** – V for signal notifying operation. V semaphore function signals from the task to the OS that the resource is now free for the other users.

16. What is shared data problem? (Remembering, CO4)

If a variable is used in two different processes and another task if interrupts before the operation on that data is completed then the value of the variable may differ from the one expected if the earlier operation had been completed .This ids known as shared data problem.

17. What is Priority inversion? (Remembering, CO4)

A problem in which a low priority task inadvertently does not release the process for a higher priority task.

18. What is Deadlock situation? (Remembering, CO4)

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.

19. Define Inter process communication. (Remembering, CO4)

IPC means that a process (scheduler or task or ISR) generates some information by setting or resetting a token or value, or generates an output so that it lets another process take note or uses it under the control of OS.

20. List out the various functions provide the software programmer in IPC. (Analyzing, CO4)

- Signals
- Semaphores as token or mutex or counting semaphores for the inter task communication between tasks sharing a common buffer or operations.
- Queues and mailboxes
- Pipes and sockets
- Remote procedure calls (RPCs) for distributed processes.

21. Define Remote Procedure Call. (Remembering, CO4)

A method used for connecting two remotely placed methods by using a protocol. Both systems work in the peer to peer communication mode and not in the client server mode.

22. Define signal handler. (Remembering, CO4)

A signal provides the shortest communication. The signal () sends an output n for a process, which enables the OS to unmask a signal mask of a process or task as per the n.

23. Define signal mask. (Remembering, CO4)

A signal mask is the software equivalent of the flag at a register that sets on masking hardware interrupt.

24. List out various signal functions provided by RTOS.(Analyzing, CO4)

- Sighandler ( )
- Connect
- Signal ( )
- Mask
- Unmask
- Ignore

25. List out various semaphore functions provided by RTOS. (Analyzing, CO4)

- OSSemCreate ( )
- OSSemPost ( )
- OSSemPend ( )
- OSSemAccept ( )
- OSSemQuery ( )

26. Define lock function. (Remembering, CO4)

There is a function in kernel is called lock ( ). I t locks a process to the resources till that process executes unlock.

27. Define spin lock. (Remembering, CO4)

A spin lock does not let a running task to be blocked instantly, but first successively tries with or without decreasing the trial periods before finally blocking a task. A spin lock obviates need of context switching by pre emption and use of Mutex function calls to OS.

28. Define Message Queue. (Remembering, CO4)

A task sending the multiple FIFO or priority messages into a queue for use by another task using queue messages as an input.

29. What are different features of message queue functions? (Remembering, CO4)

- An OS provides for inserting and deleting the message pointers or messages.
- Each queue for the message or message pointers needs initialization before using functions in kernel for the queue.
- Each created queue has an ID.
- Each queue has a user definable size.

30. List out various message queue functions provided by OS.(Analyzing, CO4)

- OSQCreate
- OSQPost
- OSQPend
- OSQAccept
- OSQFlush
- OSQQuery
- OSQPostFront

31. Define mailbox. (Remembering, CO4)

A message mailbox is for an IPC message that can be used only by a single destined task. The mailbox message pointer or can be a message.

32. List out various mailbox functions provided by OS. (Analyzing, CO4)

- OSMBoxCreate
- OSMBoxPost
- OSMBoxWait
- OSMBoxAccept
- OSMBoxQuery

33. Define pipe functions. (Remembering, CO4)

A message pipe is a device for inserting (writing) and deleting (reading) from that between two given interconnected tasks or two sets of tasks.

34. List out various pipe functions provided by OS.(Analyzing, CO4)

- PipeDevCreate ( )
- Open ( )
- Connect ( )
- Write ( )
- Read ( )
- Close ( )

35. Define Socket. (Remembering, CO4)

A socket provides a bi directional transfer of messages and may also send protocol header. The transfer is between two or between the multiple clients and server process.

36. List out the types of protocols used in socket functions. (Analyzing, CO4)

- Connectionless oriented protocol – UDP requires a UDP header
- Connection oriented protocol – TCP

37. List out various socket functions provided by OS. (Analyzing, CO4)

- Socket ( )
- Inlink ( )
- Bind ( )
- Accept ( )
- Recv ( )
- Send ( )
- Close ( )

38. What is RTOS? (Remembering, CO4)

An RTOS is an OS for response time controlled and event controlled processes. RTOS is an OS for embedded systems, as these have real time programming issues to solve.

39. What are the goals of RTOS? (Remembering, CO4)

- Facilitating easy sharing of resources
- Facilitating easy implantation of the application software
- Maximizing system performance
- Providing management functions for the processes, memory, and I/Os and for other functions for which it is designed.
- Providing management and organization functions for the devices and files and file like devices.
- Portability
- Interoperability
- Providing common set of interfaces.

40. What is meant by user mode and supervisor mode? (Remembering, CO4)

- **User mode** – The user process is permitted to run and use only a subset of functions and instructions in OS. This is done in the user mode either by sending a message to a waiting process associated with the OS kernel or by initiating a system call.

- **Supervisor mode** – The OS runs the privileged functions and instructions in the protected mode and the OS only accesses the hardware resources and the protected area memory.

41. List the functions of a kernel. (Analyzing, CO4)

- Process management
- Process creation to deletion
- Processing resource requests
- Scheduling
- IPC
- Memory management
- I/O management
- Device management

42. Define process creation. (Remembering, CO4)

Creation of a process means specifying the resources for the process and address spaces (memory blocks) for the created process, stack, data and heap and placing the process initial information at a PCB.

43. Define process manager. (Remembering, CO4)

Process manager is a unit of the OS that is the entity responsible for controlling a process execution. Process management enables process creation, activation, running, blocking, resumption, deactivation and deletion.

44. List out various timer functions provided by OS. (Analyzing, CO4)

- OSTickInit ( )
- OSTimeDelay ( )
- OSTimeSet ( )
- OSTimeGet ( )
- OSSemPend ( )
- OSTimeDelayResume ( )

45. List out various event functions provided by OS. (Analyzing, CO4)

- OSEventCreate ( )
- OSEventQuery ( )
- OSEventDelete ( )

46. What is meant by memory allocation? (Remembering, CO4)

When a process is created, the memory manager allocates the memory addresses (blocks) to it by mapping the process address space. Threads of a process share the memory space of the process.

47. What are advantages of memory manager in OS? (Remembering, CO4)

- Secure,
- Robust and
- Well protected

48. List out the various memory managing strategy for a system. (Analyzing, CO4)

- Fixed-blocks allocation
- Dynamic -blocks Allocation
- Dynamic Page-Allocation
- Dynamic Data memory Allocation
- Dynamic address-relocation
- Multiprocessor Memory Allocation
- Memory Protection to OS functions

49. What are the functions of memory managers? (Remembering, CO4)

- Use of memory address space by a process,
- Specific mechanisms to share the memory space and

- Specific mechanisms to restrict sharing of a given memory space
- Optimization of the access periods of a memory by using a hierarchy of memory (caches, primary and external secondary magnetic and optical memories).

50. Define device manager. (Remembering, CO4)

It is defined as the software that manages these for all. It polls the requests at the devices and the actions occur as per their priorities. It manages the IO interrupt queues. It creates an appropriate kernel interface and API, and that activates the control resister specific actions of the devices.

51. What are the functions of device manager? (Remembering, CO4)

- Device detection and addition
- Device deletion
- Device allocation and registration
- Detaching and deregistration
- Device sharing

52. List the set of OS command functions for a device management. (Analyzing, CO4)

- Create and open
- Write
- Read
- Close and delete

53. List the set of command functions of POSIX file system. (Analyzing, CO4)

- Open
- Write
- Read
- lSeek
- Close

54. Mention the data structure of file descriptor in a file system.(Understanding, CO4)

Identity, creator or owner state, state, locks and protection fields, file info, sharing permission, count and storing media details.

55. What is the role of file manager? (Remembering, CO4)

A file manager creates, opens, reads, seeks a record, writes and closes a file. A file has a file descriptor.

56. Mention the types of file system. (Understanding, CO4)

- **Block file system** – Its application generates records to be saved into the memory
- **Byte Stream file system**–Its application generates record streams.

57. Classify the operations of I/O subsystems. (Understanding, CO4)

Synchronous IO operations – fixed data transfer rates.

Asynchronous IO operations – variable data transfer rates. It provisions for a process of high priority not blocked during the IOs.

58. What are the functions of ISR? (Remembering, CO4)

- ISR have higher priority over the OS functions and the application tasks.
- An ISR does not also wait for mutex else it has to wait for other critical section code to finish before the critical codes in the ISR can run.

PART –B

1. Explain in detail about multiple processes and threads with an application. **(Understanding, CO4)**
2. What is meant by tasks and show the various states present in the tasking process? **(Remembering, CO4)**
3. Define semaphore and explain detail about the types of semaphores. **(Understanding, CO4)**

4. What is meant by shared data? Explain the various problems present in shared data. How it can overcome? **(Understanding, CO4)**
5. Discuss in detail about inter process communication. **(Creating, CO4)**
6. Explain in detail about following function types: **(Understanding, CO4)**
    (i) Signal function
    (ii) Semaphore function
    (iii) Message queue function
7. Explain in detail about following function types: **(Understanding, CO4)**
    (i) Mail box function
    (ii) Pipe function
    (iii) Socket function
8. Explain the concept of goals and various services of OS in detail. **(Understanding, CO4)**
9. Discuss briefly about process management. **(Creating, CO4)**
10. Write short notes on timer and event functions? **(Remembering, CO4)**
11. Explain in detail about memory management and its managing strategy. **(Understanding, CO4)**
12. Illustrate in detail about the Device, File and IO subsystems management. **(Understanding, CO4)**
13. Elaborate the Interrupt routines in RTOS environment and handling of interrupt source calls. **(Creating, CO4)**

UNIT V
CASE STUDIES
PART –A

1. **List out the various basic functions of RTOS. (Analyzing, CO5)**
    - Basic kernel functions and scheduling: pre-emptive or pre-emptive plus time slicing.
    - Priorities definitions for the tasks and IST.
    - Priority inheritance feature with option of priority ceiling feature.
    - Limit for number of tasks.
    - Task synchronization and IPC function.
    - Device imaging tool and device drivers.
2. **What are the different development approaches in RTOS? (Remembering, CO5)**
    Host target approach– It is used in real time or non real time application development.
    Self host development approach – The system with full RTOS is used for development on which the application runs.
3. **Mention the different types of RTOS. (Understanding, CO5)**
    In house developed RTOS – The codes are written for the specific need, and application or product and customizes.
    Broad based commercial RTOS – A readily available broad band commercial RTOS package.
    General purpose OS with RTOS – It can be used in combination with the RTOS function.
    Special focus RTOS – It is used with specific processors like ARM or 8051 or DSP.
4. **Name any two important RTOS. (Understanding, CO4)**
    - MUCOS
    - VxWorks
5. **What are the different types of source files used in MUCOS? (Remembering, CO5)**
    Processor dependent source files – Two header files are considered, processor definitions header file and kernel building configuration file.
    Processor independent source files – Two header files are considered, MUCOS header file and C file.

6. **List out the various features or functions present in MUCOS-II. (Analyzing, CO5)**
    - Scalable
    - Multitasking
    - System level function
    - Task service function
    - Task delay function
    - IPC function
7. **What is MICRO C/OS II? (Remembering, CO5)**
    - It stands for micro-controller operating system(UC/OS II).
    - It is a real time kernel
    - The other names of MICROC/OS II are MUCOS and UCOS.
    - The codes are in 'C' and Assembly language.
8. **What are the real time system level functions in UC/OS II? State some? (Remembering, CO5)**
    - Initiating the OS before starting the use of the RTOS functions.
    - Starting the use of RTOS multi-tasking functions and running the states.
    - Starting the use of RTOS system clock.
    - Sending message to RTOS kernel for taking control at the start of an ISR.
    - Sending a message to RTOS kernel for quitting the control at return from an ISR.
9. **What are the system level functions supported by MUCOS? (Remembering, CO5)**
    - Void OSInit (void)
    - Void OSStart(void)
    - VoidOSTickInit(void)
    - Void OSIntEnter(void)
    - Void OSIntExit(void)
10. **What are the task service and time functions supported by MUCOS? (Remembering, CO5)**
    - OSTaskCreate(void)
    - OSTaskSuspend (void)
    - OSTaskResume (void)
    - OSTimeSet (unsigned int count)
    - OSTimeGet (void)
11. **What are the time delay functions supported by MUCOS? (Remembering, CO5)**
    - OSTimeDly (unsigned short delay Count)
    - OSTimeDlyResume(unsigned byte task priority)
    - OSTimeDlyHMSM(Unsigned byte hr, unsigned byte mm, unsigned byte sec , unsigned short ms)
12. **What are the memory allocations related functions supported by MUCOS? (Remembering, CO5)**
    - OSMemCreate (void)
    - OSMemGet (void)
    - OSMemQuery (void)
    - OSMemPut (void)
13. **What are the semaphores related functions supported by MUCOS? (Remembering, CO5)**
    - OS_EventOSSemCreate(unsigned short semval)
    - Void OSSemPend(OS_Event *eventPointer, unsigned short timeout,unsigned byte*SemErrPointer)
    - unsigned short OSSemAccept(OS_Event*eventPointer)
    - unsigned short OSSemPost(OS_Event*eventPointer)
14. **What are the mail boxes related functions supported by MUCOS? (Remembering, CO5)**

- OSMboxCreate(void*msg)
- OSMboxAccept(OS_EVENT *mboxPointer)
- OSMboxPend(Unsigned byte)
- OSMboxPost(void)
- OSMboxQuery(OS_EVENT *mboxPointer)

15. What is the queue related functions supported by MUCOS? (Remembering, CO5)

- OSQCreate(void)
- OSQPend(OS_EVENT *QmsgPointer)
- OSQFlush(OS_EVENT *QmsgPointer)
- OSQPost(OS_EVENT *QmsgPointer)
- OSQQuery(OS_EVENT *QmsgPointer)

16. Write short notes on Vxworks? (Remembering, CO5)

- Vxworks is a popular Real-time multi-tasking operating system for embeddedmicroprocessors and systems.
- Vxworks can run on many target processors.
- It is a UNIX like Real time operating system.

17. How is Vx Works TCB helpful for tasks? (Remembering, CO5)

- Provide control information for the OS that includes priority, stack size, state and options.
- CPU context of the task that includes PC, SP, CPU registers and task variables.

18. What are the various features of Vx Works? (Remembering, CO5)

- VxWorks is a scalable OS
- RTOS hierarchy includes timers, signals, TCP/IP sockets, queuing functions library, Berkeley ports and sockets, pipes, UNIX compatible loader, language interpreter, shell, debugging tools, linking loader for UNIX.
- Multitasking
- System level function

19. What is an active task in the context of Vx Works? (Remembering, CO5)

Active task means that it is in one of the three states, ready, running, or waiting.

20. What are the task service functions supported by Vx Works? (Remembering, CO5)

- taskSpawn()
- taskResume()
- taskSuspand()
- taskDelay()
- taskSuspand()
- taskInit()
- exit()
- taskDelete()

21. Name any four interrupt service functions supported by Vx Works?(Understanding, CO5)

- intLock()
- intVectSet()
- intVectGet()
- intContext()

22. Name some of the inter process communication function in Vx Works. (Understanding, CO5)

- semBCreate()
- semMCreate()
- semCCreate()
- semTake()
- semDelete()

23. Name some of the inter process communication function used for messaging in Vx Works. (Understanding, CO5)

- msgQCreate()
- msgQDelete()
- msgQSend()
- msgQReceive()

24. What are VxWorks pipes? (Remembering, CO5)

VxWorks pipes are the queues that can be opened and closed like a pipe. Pipes are like virtual IO devices that store the messages as FIFO.

25. What are the different types of scheduling supported by Vx Works? (Remembering, CO5)

- Preemptive priority
- Time slicing

26. Define Window CE. (Remembering, CO5)

It is an RTOS for handheld computers and mobile systems, CE stand for the compact, connectable, compatible, companion and efficient. It can also be perceived as windows for consumer electronics systems, however applications do not limit to consumer electronics systems.

27. List out the various features of Windows CE. (Analyzing, CO5)

- Provides a windows platform for the systems
- It is an open, scalable and small footprint 32 bit OS
- Enables running of PocketPC applications
- Multitasking and multithreaded OS
- Low interrupt latency

28. What are the memory management types in Windows CE? (Remembering, CO5)

Program memory (system heap) – It is allocated to the program from a pool of unused memory area.

Object store – It is a virtual RAM disk for permanent store, which is protected from turning off power.

29. Define static allocation and stack in Windows CE. (Remembering, CO5)

Static allocation – WCE allocates two allocations, one for read only and other for read/write data.

Stack – Stack stores the temporary variables and processor registers for the application and OS functions.

30. What are different properties created by files in Windows CE? (Remembering, CO5)

- Long pointer for character string
- 32 bit desired access parameter
- 32 bit shared mode specification
- Long pointer for security attributes
- 32 bit to specify creation and distribution

31. Define registry and its functions in Windows CE. (Remembering, CO5)

- WCE registry uses standard registry API. Registry API is an API for system database. It uses standard file registry functions.
- The functions are RegCreateKeyEx, RegOpenKeyEx, RegSetValueEx, RegQueryValueEx, RegDeleteKeyEx, RegDeleteValueEx and RegCloseKey.

32. List out the various feature or properties of Windows CE databases. (Analyzing, CO5)

- Database format
- Maximum number of records and record size
- Database record properties data types
- CeMountDBVol Windows CE functions

- CeCreateDatabaseEx Windows CE function
- CloseHandle( )

33. What are the properties and windows CE functions for Thread and processes? (Remembering, CO5)

- Thread properties
- Thread priorities
- Maximum number of processes and memory size
- Create Process WCE function
- Terminate Process Windows CE function
- Open Process Windows CE function

34. What are the Windows CE functions for exceptions, notification, Events, Semaphores, Mutex and Message Queues? (Remembering, CO5)

- Signalling (exception) functions
- Signalling (notification) functions
- Critical section functions
- Semaphore functions (for thread of a process)
- Message queue functions
- Event functions

35. Define windows control and Windows Menus. (Remembering, CO5)

Windows control– Each window uses a number of classes, called controls. A control has a number of user interface elements.

Windows Menus – WCE menus are at the menu bar or command bar control.

36. What are the functions of Windows CE serial communication? (Remembering, CO5)

- CreateFile WCE functions
- ReadFile
- WriteFile
- TransmitCommChar
- Set CommMask
- Get CommMask
- WaitCommEvent

37. List out the various WNet API network connection functions. (Analyzing, CO5)

- WNetAdd-Connection
- WNetConnection-Dialog
- WNetCancel-Connection
- WNetDis-connectDialog
- WNetGet-Connection
- WNetGetUser

38. What are the functions of Winsock API subset in WCE for device – device socket communication? (Remembering, CO5)

- Socket function
- Bind function
- Accept function
- Connect function
- Listen function
- Send function
- Recv function

39. Define Win32 API programming. (Remembering, CO5)

It is defined as the most important part of application development in a computer or embedded system or handheld system which has a screen or touch screen for interaction with a

user. It has large number of APIs in a PC. However, only a subset is required for handheld devices and small screen size systems.

40. Mention the different standards in OSEK/VDX.(Understanding, CO5)

- **Real time execution of ECU** (Electronic Control Units) and MODISTARC (Methods and tools for the validation of OSEK/VDX based distributed architectures.
- **Communications stack** for data exchange
- **Network management protocol** for automotive embedded systems configuration determination and monitoring.

41. List out the various standards in OSEK. (Analyzing, CO5)

- **OSEK-OS** has greater reliability
- **OSEK-NM** for network management
- **OSEK-COM** for IPC between the same CPU control unit tasks and between the different CPU control unit tasks.

42. What are the importances of Linux 2.6.x? (Remembering, CO5)

- It is used in embedded systems and real time enhancements
- It provides preemptive scheduling, high resolution timers and preemptive interrupt service threads.

43. Mention the functions of embedded Linux application program.(Understanding, CO5)

- Process management
- Memory management
- File system
- Shared memory
- Networking system functions
- Device control functions
- Graphic editors
- Soft real time scheduling

44. List out the features of Linux. (Analyzing, CO5)

- It is multi user OS and supports user groups
- It has root directory
- It has large number of editor, file, directory, IO commands.
- It has number of interfaces for user.
- It uses POSIX processes and threads.

45. List the registering and de-registering functions in Linux modules. (Analyzing, CO5)

init, insmod, rmmod, cleanup, register_capability, inregister_capability, and register_symtab.

46. What the Linux functions for the signal, multithreaded, semaphore and message queue interprocess communication? (Remembering, CO5)

Thread properties, Signal function, Thread function, Semaphore function, Mutex function, Message queue function, Time function and Shared memory function.

47. What is meant by RTLinux? (Remembering, CO5)

An extension of Linux version in which earlier there was no real time support is a POSIX hard real time environment using a real time core. The core is called RTLinuxFree, and RTLinuxPro, freeware and commercial software respectively.

48. List out the configuration for hard real time performance. (Analyzing, CO5)

- Run the primitive tasks with only statically allocated memory.
- Run the real time task with no address space protection.
- Run with disabling of interrupts so that other interrupts do not introduce the unpredictability.
- Run a simple fixed priority scheduler.

49. List the registering, de-registering, priority and scheduling of real time thread and FIFO related function in RTLinux. (Analyzing, CO5)

Initinsmod, cleanup, rmmod, RT Linux functions, thread creation and getting thread ID, Semaphore and mutexfucntions, thread wait and thread priority.

50. What the various units present in ACVM? (Remembering, CO5)

Alphanumeric keypad, three line LCD display, coin insertion slot, delivery slot and interrupt connection port.

51. List out the various commands used in Tasks functions and IPC. (Analyzing, CO5)

- Task_ReadPorts
- Task_Collect
- Task_Deliver
- Task_Refund
- Task_ExcessRefund
- Task_Display

52. What are major role of Adaptive Cruise Control (ACC)? (Remembering, CO5)

The system is commonly used in aviation electronics and defense aircrafts for cruising since along. Use in the automotives is of recent origin.

53. Classify the different ports and their functions in ACC. (Understanding, CO5)

- Port_Align – It is a stepper motor port.
- Port_ReadRange – It is a front end car range measuring port.
- Port_Speed – The port control function routine enables free running counter overflow interrupt.
- Port_Brake –Port device applies the brakes or emergency brakes on an interrupt signal.

54. List out the hardware standards in ACC architecture. (Analyzing, CO5)

- TTP (Time Triggered Protocol)
- CAN (Controller Area Network)
- MOST (Media Oriented System Transport)
- IEE (Institute of Electrical Engineers)

55. List out the hardware units in ACC architecture. (Analyzing, CO5)

Microcontroller, Separate processor with RAM and ROM, Speedometer, Stepper motor based alignment unit, Transceiver, LCD and port devices.

PART-B

1. Describe in detail about the basic functions and types of RTOS. **(Remembering, CO5)**
2. Discuss in detail about various functions of MUCOS-II in RTOS. **(Creating, CO5)**
3. What are features of VxWorks and explain detail about different functions used in VxWorks RTOS? **(Remembering, CO5)**
4. Explain in detail about Windows CE in RTOS. **(Understanding, CO5)**
5. Explain in briefly about communication and networking in Windows CE RTOS. **(Understanding, CO5)**
6. Explain in detail about OSEK in RTOS. **(Understanding, CO5)**
7. Explain briefly about Linux 2.6 and Real time Linux. **(Understanding, CO5)**
8. Explain briefly about various design process in Automatic Chocolate Vending Machine. **(Understanding, CO5)**
9. Explain briefly about Sending application layer byte streams on a TCP/IP network. **(Understanding, CO5)**
10. Explain briefly about various design process in Adaptive cruise control in a car system. **(Understanding, CO5)**