P16CSE2A -Embedded System

2-Mark Questions with answerws

1. Define Embedded System?

An Embedded System is one that has computer hardware with software embedded in it as one of its most important Components.

2. What are the Characteristics of Embedded System?

- ➢ Real time: system must be fast enough to react to physics
  - ➢ Critical: safety-critical and/or business critical
  - ➢ Limited resources: memory, processor, energy, space

3. What are the User function calls used embedded system in c?

- ✓ Declaring a function
- ✓ Defining the statements in the function
- ✓ Call to a function

4. What are programming advantages of C++?

C++ is an object oriented Program (OOP) language, which in addition, supports the procedure oriented codes of C. Program coding in C++ codes provides the advantage of objected oriented programming as well as the advantage of C and in-line assembly. Programming concepts for embedded programming in C++ are as follows:

(i) A class binds all the member functions together for creating objects. The objects will have memory allocation as well as default assignments to its variables that are not declared static. Let us assume that each software timer that gets the count input from a real time clock is an object. Now consider the codes for a C++ class RTCSWT. A number of software timer objects can be created as the instances of RTCSWT.

(ii) A class can derive (inherit) from another class also. Creating a child class from RTCSWT as a parent class creates a new application of the RTCSWT.

(iii) Methods (C functions) can have same name in the inherited class. This is called method overloading. Methods can have the same name as well as the same number and type of arguments in the inherited class. This is called method overriding. These are the two significant features that are extremely useful in a large program.

5. What are then the disadvantages of C++ ?

Program codes become lengthy, particularly when certain features of the standard C++ are used. Examples of these features are as follows:

(a) Template.

(b) Multiple Inheritance (Deriving a class from many parents).

(c) Exceptional handling.

(d) Virtual base classes.

(e) Classes for IO Streams. [Two library functions are cin (for character (s) in) and cout (for character (s) out). The I/O stream class library provides for the input and output streams of characters (bytes.

6. What is Device Driver in Embedded System?

A **device driver** is a computer program that operates or controls a particular type of device that is attached to a computer. A driver provides a software interface to hardware devices, enabling operating systems and other computer programs to access hardware functions without needing to know precise details about the hardware being used.

7. What is interrupt service Mechanism?

An **interrupt** is a signal to the processor emitted by hardware or software indicating an event that needs immediate attention. Whenever an **interrupt** occurs, the controller completes the execution of the current instruction and starts the execution of an **Interrupt Service** Routine (**ISR**) or **Interrupt** Handler.

8. What is context switching in embedded system?

Context Switching involves storing the **context** or state of a process so that it can be reloaded when required and execution can be resumed from the same point as earlier.

9. What is meant by real time programming in embedded system?

Real-time systems are computer **systems** that monitor, respond to, or control an external environment. This environment is connected to the computer **system** through sensors, actuators, and other input-output interfaces.

10. What is multitasking in embedded systems?

Multitasking is a method by which multiple tasks, also known as processes, share common processing resources, such as CPU.

11. What are the Java has advantages for embedded programming ?

1. Java is completely an OOP language.

2. Java has in-built support for creating multiple threads. [For the definition of thread and its similarity in certain respects to task refer to Section 8.1.] It obviates the need for an operating system (OS) based scheduler [Section I.5.6] for handling the tasks.

3. Java is the language for most Web applications and allows machines of different types to communicate on the Web.

4. There is a huge class library on the network that makes program development quick.

5. Platform independence in hosting the compiled codes on the network is because Java generates the byte codes. These are executed on an installed JVM (Java Virtual Machine) on a machine. [Virtual machines (VM) in embedded systems are stored at the ROM.] Platform independence gives portability with respect to the processor used. 6. Java does not permit pointer manipulation instructions. So it is robust in the sense that memory leaks and memory related errors do not occur. A memory leak occurs, for example, when attempting to write to the end of a bounded array.

7. Java byte codes that are generated need a larger memory when a method has more than 3 or 4 local variables. 8. Java being platform independent is expected to run on a machine with an RISC like instruction execution with few addressing modes only.

12. Define Real time Operation System.

A **real-time operating system** (**RTOS**) is an operating system (OS) intended to serve real-time applications that process data as it comes in, typically without buffer delays. Processing time requirements (including any OS delay) are measured in tenths of seconds or shorter increments of time. A real-time system is a time-bound system which has well-defined, fixed time constraints.

13. Define Preprocessor directives.

Program statements and directives for the compiler before the main function to define global variable, global macro (section of code), new data type and global constant

14. What is interrupt latency?

Interrupt latency refers primarily to the software interrupt handling latencies. In other words, the amount of time that elapses from the time that an external **interrupt arrives** at the processor until the time that the **interrupt processing begins**.

15. What is response time?

The **response time** for a job is the **time** between when it becomes active (e.g. an external event or timer triggers an interrupt) and the **time** it completes

16. Which software is used for Embedded System?

The **software** and operating **system** requirements of an **embedded system** are also different from a traditional computer based **system**. Typically **embedded systems** use basic **embedded system software** such as C, C++, ADA, etc. Some specialized **embedded systems** may use OS such as Windows CE, LINUX, TreadX, Nucleus RTOS, OSE, etc.

17. What are the disadvantages of embedded systems?

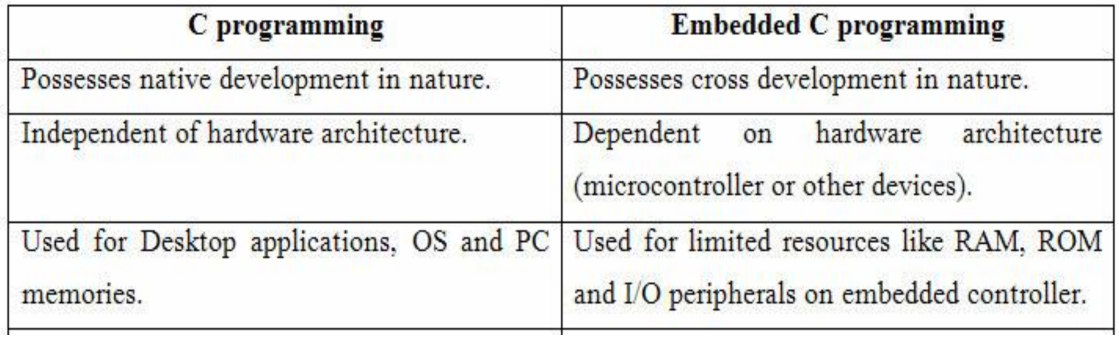
- Difficult to upgrade.
- If any problem occurs then you need to reset settings.
- Nearly not scalable.
- Hardware is limited.
- Troubleshooting is difficult.
- Difficult to transfer data from one system to other.

18. Define Design Cycle in Embedded system.

- Product specification
- Partitioning of the design into its software and hardware components
- Iteration and refinement of the partitioning
- Independent hardware and software design tasks
- Integration of the hardware and software components
- Product testing and release
- On-going maintenance and upgrading

19. What is difference between C and Embedded C?

Differences between C and Embedded C



C programming	Embedded C programming
Possesses native development in nature.	Possesses cross development in nature.
Independent of hardware architecture.	Dependent on hardware architecture (microcontroller or other devices).
Used for Desktop applications, OS and PC memories.	Used for limited resources like RAM, ROM and I/O peripherals on embedded controller.

20. Give the advantages of Embedded C Program.

- its takes less time to develop application program.
- It reduces complexity of the program.
- It is easy to verify and understand.
- It is portable in nature from one controller to another.

21. Enumerate the Characteristics of Embedded System:

✓ Requires Less power
✓ Low Cost
✓ Task Specific
✓ Time Specific
✓ Minimal user Interface
✓ High Efficiency
✓ High Reliability

- ✓ High Stable

22. Define Optimisation of Memory.

Certain steps changed to reduce the need for memory and having a compact code. It reduces the total size of the memory needed. It also reduces the total number of CPU cycles, and thus, the total energy requirements.

5 (or) 10 Marks Questions

1. Briefly Explain about I/O Ports, Buses and Interrupt handlers that are embedded in a system.

2. Briefly Explain about Various types and uses of RAM and ROM for designing embedded systems.

3. Explain how optimization of codes in embedded C++ can be done

4. Briefly explain how control data flow graph is used to model a program.

5. Explain how graphs can be used to model partitioning and scheduling of multiprocessors systems.

6. Briefly explain the typical components of an I/O Sub system.

7. Explain round robin time slicing scheduling in RTOS with an example.

8. Explain the role of target system in the design process of an Embedded system.

9. Describe various issues in hardware software design and co-design.

10. Explain various structural units of a processor in an Embedded system with a neat diagram.

11. Discuss in detail about Interrupt Servicing Mechanism.

12. Explain the state machine programming models for event controlled program flow.

13. Explain in detail the interrupt handling in RTOS environment.

14. Explain the basic system of an automatic chocolate vending systems.

15. Discuss in detail about Data structures in Embedded Systems.

16. Write short note on Embedded Programming in Java.

17. Explain about Embedded programming in C++.

18. List out the data types, statements, Loops in C.

19. Discuss about Performance metrics in schedule model.

20. Explain about the automatic vender machine in Embedded System.

21. What is Macro? Explain it.

22. Explain about programming model in Embedded System.

23. Write short note on the following:
    Header files.
    Preprocessor directive
    Functions

24. Write short note an Operating System Services in Embedded system.

25. Explain about Task scheduling model in RTOS .

26. Explain use of scopes and logical analyzer for system hardware test.

27. Explain about Sharing data in multiple task and functions.

28. Discuss in detail about design issues in Embedded System.