

B.E. (Electronics & Communication / Telecommunication Engineering) Model Curriculum
Semester-VII
ET704M-2 / PEC-2 - Embedded Systems

P. Pages : 2

Time : Three Hours



GUG/W/24/14251

Max. Marks : 80

-
- Notes :
1. All questions carry marks as indicated.
 2. Assume suitable data wherever necessary.
 3. Illustrate your answers wherever necessary with the help of neat sketches.

1. a) What are the challenges faced in designing an embedded system? 8
b) Define design metrics in embedded systems. What are the different competing design metrics? What are the constraints of embedded system design? 8

OR

2. a) Write a note on classification of embedded systems. 8
b) Explain Digital camera as an embedded system. 8
3. a) Compare memory-mapped IO and IO mapped IDs. 8
b) Compare Harvard and Princeton memory organizations. 8

OR

4. a) Explain three-stage pipeline, superscalar processing and branch and data-dependency penalties. 8
b) Why should a program be divided into functions (routines or modules) and each placed in different memory blocks or segments? 8
5. a) Why do you break a program into header files, configurations files, modules and functions? 8
b) Explain the importance of the following declarations: Static, Volatile and Interrupt in embedded C. 8

OR

6. a) Why do you need a cross compiler? 8
b) What are the advantages of building ISR queues? 8
7. a) How does use of a counting semaphore differ from a mutex? How is a counting semaphore used? 8

- b) What are the states of a task? What is the entity controlling (scheduling) the transitions from one state to another in a task? **8**

OR

- 8.** a) What are the situations, which lead to priority inversion problems? How does an OS solve this problem by a Priority inheritance mechanism? **8**

- b) What is a mailbox? How does a mailbox pass a message during IPC? **8**

- 9.** a) Write a note on MUCOS features. **8**

- b) Write a note on IPC functions. **8**

OR

- 10.** a) With neat diagram explain Digital Camera hardware architecture. **8**

- b) Explain how the task for reading ports synchronizes with the port device driver. **8**
