UNIT V

EMBEDDED SYSTEM APPLICATION DEVELOPMENT

5.2 µ **C/OS – II System level and task Functions**

- Void *OSI t(void)* At the beginning prior to the OS Start( )
- void *OS Start (void)* After OSI nit () and task- creating function(s)
- void *OS Tick Init (void)*In first task function that executes once. Initializes the system timer ticks (RTC interrupts)

Interrupt Service Task (ISR) Start and End

- *OS int Enter ()and OS Int Exit( )*
- Function void OS Int Enter (void)─ used at the start of ISR For sending a message to RTOS kernel for taking control ─ compulsory to let OS kernel control the nesting of the ISRs in case of occurrences of multiple interrupts of varying priorities.
- Function void OS Int Exit (void)─ used just before the return from the running ISR─ For sending a message to RTOS kernel for quitting control of presently running ISR

Critical Section Start and End

- OS_ENTER_CRITICAL
  - Macro to disable interrupts before a critical section
  - Used at the start of a ISR or task - for sending a message to RTOS kernel and disabling the interrupts
  - Use compulsory when the OS kernel is to take note of and disable the interrupts of the system
- OS_EXIT_CRITICAL─ Macro to enable interrupts. [ENTER and EXIT functions form a pair in the critical section]
  - used at the end of critical section
  - for sending a message to RTOS kernel and enabling the interrupts
  - Use is compulsory to OS kernel for taking note of and enables the disabled interrupts.

Function void *O Stick In it (void)*
— is used to initiate the system clock ticks and interrupts at regular intervals as per OS _ TICKS _ PER _SEC predefined when defining configuration of