EPSON

E0C33 Family ROS33 Middleware

## Realtime OS middleware

● Realtime OS for E0C33 Family
● Suport µITRON 3.0 level S
● Optimize for E0C33 Family

■ DESCRIPTION

• The ROS33 is a realtime operating system for the E0C33 Family of single-chip microcomputers based on µITRON 3.0. System calls up to Level S (Standard) are supported.

• Programs can be developed in C and assembly language.

• Compact and high-speed kernel optimized for use in the E0C33 Family.

• The ROS33 is released with a 3.5-inch FD including all source code, library and sample programs. So you can customize the ROS33 for your system.

• Multiple tasks can share a common stack area (when not processed in parallel). You can minimize the amount of RAM used in your system by your application.

• Using ROS33 in your design enables you to quickly and efficiently develop embedded applications for printers, PDAs, FA products and various types of control equipment.

■ FEATURES

● System Diagram
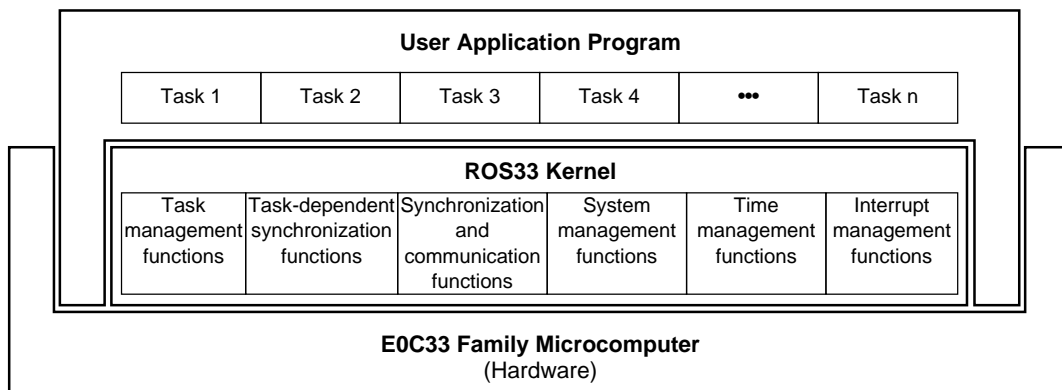
● Resources

Number of tasks : 1 to 255	Number of semaphores : 1 to 255
Number of priority levels : 1 to 9	Number of mailboxes : 1 to 255
Number of event flags : 1 to 255	

● Kernel Size

µITRON 3.0 Level R (Required) support : 1,748 bytes
µITRON 3.0 Level S (Standard) support : 2,632 bytes
Maximum configuration (including debug kernel) : 3,804 bytes

● Speed of ROS33

E0C33 system configuration	Dispatch time	Interrupt time
20MHz, internal ROM, internal RAM	12.9µS	7.2µS
20MHz, external ROM (2 Wait), internal RAM	23.6µS	14.8µS
33MHz, internal ROM, internal RAM	7.84µS	4.3µS
33MHz, external ROM (2 Wait), internal RAM	14.3µS	9.0µS

These values were evaluated using the ICE33 when tasks of the same priority were switched over by a rot_rdq system call.

E0C33 Family ROS33 Middleware

● **System Call List**

Task Management Functions			
ena_dsp()	Enable Dispatch	**dis_dsp()**	Disable Dispatch
ext_tsk()	Exit Issuing Task	**sta_tsk()**	Start Task
chg_pri()	Change Task Priority	**ter_tsk()**	Terminate Other Task
rel_wai()	Release Wait of Other Task	**rot_rdq()**	Rotate Tasks on the Ready Queue
Task-Dependent Synchronization Functions			
wup_tsk()	Wakeup Other Task	**slp_tsk()**	Sleep Task
sus_tsk()	Suspend Other Task	**rsm_tsk()**	Resume Suspended Task
can_wup()	Cancel Wakeup Request		
Synchronization and Communication Functions			
preq_sem()	Poll and Request Semaphore	**sig_sem()**	Signal Semaphore
wai_sem()	Wait on Semaphore	**rcv_msg()**	Receive Message from Mailbox
prcv_msg()	Poll and Receive Message from Mailbox	**snd_msg()**	Send Message to Mailbox
wai_flg()	Wait on Eventflag	**pol_flg()**	Wait for Eventflag (Polling)
set_flg()	Set Eventflag	**clr_flg()**	Clear Eventflag
System Management Functions			
get_ver()	Get Version Information		
Time Management Functions			
set_tim()	Set System Clock	**get_tim()**	Get System Clock
dly_tsk()	Delay Task		
Interrupt Management Functions			
loc_cpu()	Lock CPU	**unl_cpu()**	Unlock CPU