



CMX-SchedulerTM for dsPICTM

CMX-Scheduler Preemptive Kernel is *FREE* for the dsPIC!

CMX-Scheduler is the result of a special collaboration between CMX and Microchip. Available in object code only, CMX-Scheduler is available for FREE to embedded systems designers using the dsPIC series of processors. CMX-Scheduler has been specially designed for use by developers whose designs do not require a full-blown RTOS and/or who are wondering if a kernel might help their application. The perfect entry level kernel, CMX-Scheduler is intuitive to use and easy to implement.

CMX-Scheduler also offers many growth paths for future designs. User Applications developed with the CMX-Scheduler kernel are upwardly compatible with the popular CMX-Tiny+TM or CMX-RTXTM RTOSes. CMX-Scheduler also has been tightly integrated with the unique CMX-MicroNetTM TCP/IP stack for those applications that require networking connectivity. CMX-Scheduler software and documentation is delivered in electronic format and is freely licensed for unlimited product usage on the dsPIC.

Download it now: <http://www.cmx.com/microchip/>

CMX-Scheduler Specifications for the dsPIC:

All CMX Functions:	972 bytes
CMX Initialize Module:	153 bytes
CMX Assembly Module:	567 bytes

RAM, Each Task Control Block:	11 bytes
FLASH, Each Task Control Block:	5 bytes

Min. Context Switch: 81 cycles (starting a task)
102 cycles (resuming a task)

NOTE:

CMX Functions are contained in a library, thus reducing code size, if not referenced.

CMX-Scheduler Features

- ◆ **FREE for Use on Any dsPIC!**
- ◆ **Easy to Learn and Use**
- ◆ **Truly Preemptive Kernel**
- ◆ **Supports Up to Five Tasks**
- ◆ **Fast Performance**
- ◆ **Free Bug Fixes and Updates**
- ◆ **No Royalties on Shipped Products**
- ◆ **Compatible with CMX-Tiny+ & CMX-RTX**
- ◆ **Complete Electronic Documentation**
- ◆ **Integrated with CMX-MicroNet for Optional Networking Connectivity**

CMX-Scheduler Functionality

K_Task_Create - creates a task
K_Task_Start - starts a task
K_Task_Wake - wake a task
K_Task_Wait - have a task wait with/without a timeout
K_Task_Kill - Delete a task

K_Task_Coop_Sched - perform a cooperative task switch
K_Event_Wait - wait on an event
K_Event_Signal - signal an event from a task
K_Event_Signal - signal an event from an interrupt
K_Event_Reset - reset an event for a particular task