



# CMX-RTX™ for MAXQ™

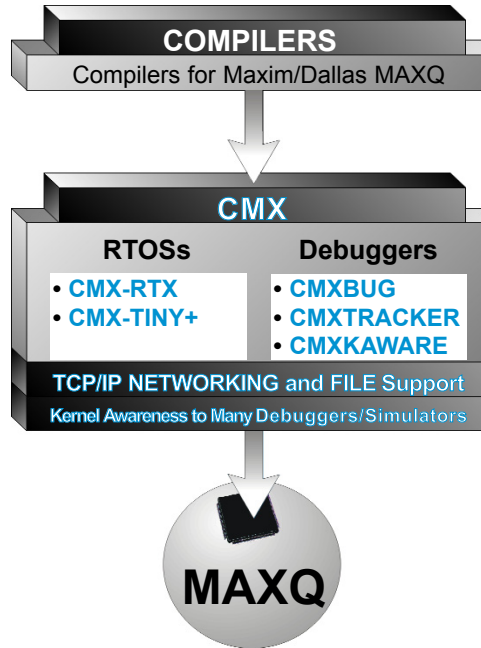
## IS YOUR PROCESSOR IN NEED OF A REAL-TIME MULTI-TASKING OPERATING SYSTEM?

- ◆ Does your processor control the way you program?
- ◆ Do you spend too much time figuring out how to make a section of code execute when it needs to?
- ◆ Do you constantly have to test flags (or go to routines that test) to see if you should execute a certain function?
- ◆ Do you spend too much time with interrupt routines while trying to write the code necessary to process the interrupt's event because the main code would not get to it in a timely manner?

If you answered YES to any of the above questions, you would certainly benefit from considering an RTOS for your application.

In some cases, well structured linear programming is sufficient for a product. In most cases, however, programmers appreciate not having to worry about structuring their code to perform all necessary tasks in a timely manner. This is where CMX-RTX can help. CMX-RTX allows tasks (pieces of code that do specific duties) to run quasi-concurrently. This means that tasks will seem to run all at the same time - doing many specific jobs simultaneously.

CMX-RTX takes the worry and headaches out of real time programming. Our software lets you concentrate on the overall application while taking care of the little details for you. Finish your projects faster and more efficiently with CMX-RTX in your programmer's toolbox!



*CMX-RTX offers a truly preemptive, multi-tasking operating system.*

## WHAT DOES TRULY PREEMPTIVE MEAN?

Some RTOS vendors offer only cooperative scheduling which means that the running task has to call the scheduler to perform a task switch. Others offer time slicing in which each task runs for a certain period of time at which point a task switch takes place no matter what. Other vendors claim to be fully preemptive, yet they do not allow any interrupt to cause a preemption. All of these models will fail you at one point or another.

CMX-RTX allows a task of higher priority that is able to run (whether starting or resuming) to preempt the lower priority running task. This will cause the scheduler to save the context of the running (lower priority) task and restore the context of the higher priority task so that it is now running. A truly preemptive RTOS allows interrupts to cause an immediate task switch. This means that the interrupts now have the added ability of using the RTOS's functions and causing an immediate context switch if needed.

## NOT ALL REAL TIME OPERATING SYSTEMS ARE CREATED EQUAL!

CMX-RTX is a powerful RTOS that uniquely provides:

- ◆ The smallest footprint
- ◆ The fastest context switch times
- ◆ The lowest interrupt latency times
- ◆ True preemption
- ◆ Scheduler and interrupt handler written in assembly for speed and optimization
- ◆ Optional co-operative and time-slicing scheduling
- ◆ Nested interrupts
- ◆ All functions contained in a library
- ◆ Interrupt callable functions
- ◆ Scalability
- ◆ Full source code provided

## NOT ALL COMPANIES ARE CREATED EQUAL

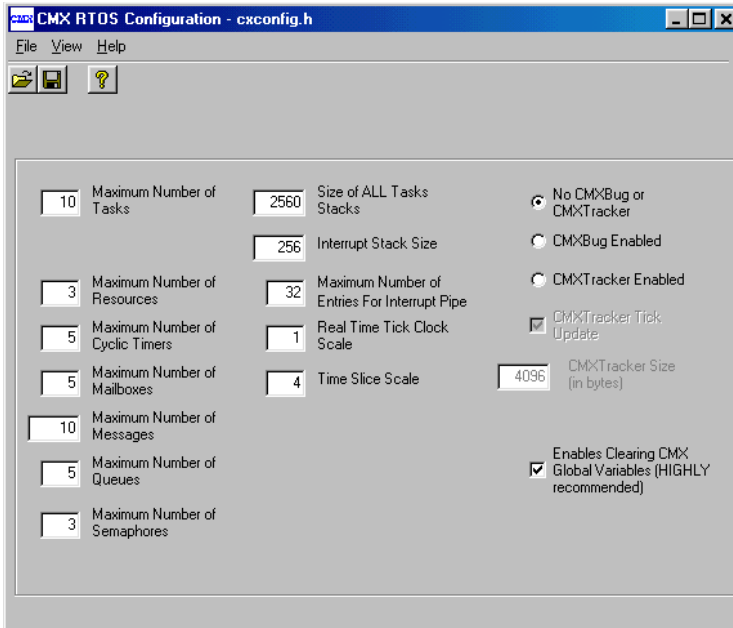
CMX technical support is renowned throughout the world. Average wait times for a tech support person is under one minute and over 95% of our calls are resolved over the phone! If a question cannot be resolved over the phone, the answer is usually found and relayed to the customer within hours.

Our philosophy also includes giving the engineer all of the source code to the product. This is not only an invaluable debugging tool, but dramatically reduces the learning curve associated with any RTOS. It also allows for smaller code size because only functions that are used are linked into the final output module.

CMX Systems, Inc. • 12276 San Jose Blvd #119 • Jacksonville, FL 32223

Tel: 904.880.1840 • Fax: 904.880.1632 • Email: [cmx@cmx.com](mailto:cmx@cmx.com) • WWW: <http://www.cmx.com>

# CMX RTOS Configuration Manager



CMX-RTX is easy to configure and integrate with your application using the CMX RTOS Configuration Manager.

## CMX-RTX Specifications for MAXQ

### FLASH

All CMX Functions:	5633 words
CMX Initialize Module:	842 words
CMX Assembly Module (scheduler):	448 words
RAM, Each Task Control Block:	28 bytes

### NOTES:

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

# A PARTIAL LISTING OF CMX-RTX FUNCTIONS

## TASK MANAGEMENT

Create a task.  
Remove a task.  
Start a task.  
Suspend a task, with time-out provision.  
Wake a suspended task.  
Forcefully wake a task.  
Change a task's priority.  
Terminate a task early.  
Do a cooperative rescheduling.  
Disable task scheduling.  
Enable task scheduling.

## MESSAGE MANAGEMENT

Get a message.  
Wait for a message, with time out provision.  
Send a message.  
Send a message, wait for reply.  
Wake task that sent message, if waiting on reply.  
Wait on Mailbox(s), with time out provision.

## QUEUE MANAGEMENT

Create a circular queue.  
Reset queue to empty.  
Add to top of queue.  
Add to bottom of queue.  
Remove from top of queue.  
Remove from bottom of queue.

## SYSTEM MANAGEMENT

Initialize CMX.  
Enter CMX.  
Enter interrupt.  
Exit interrupt.  
Enter power down mode.

## EVENT MANAGEMENT

Wait on event(s), with time-out provision.  
Set an event.  
Clear an event.

## MEMORY MANAGEMENT

Create a fixed block pool.  
Request free block from pool.  
Release block back to pool.

## RESOURCE MANAGEMENT

Get a resource.  
Reserve a resource, with time out provision.  
Release a resource.  
Automatic priority inversion.

## SEMAPHORE MANAGEMENT

Get semaphore.  
Pend for semaphore, with time out provision.  
Post to semaphore.  
Flush semaphore.

## TIMER MANAGEMENT

Create a cyclic timer.  
Change a cyclic timer event parameters.  
Start a cyclic timer.  
Restart a cyclic timer.  
Stop a cyclic timer.  
Restart a cyclic timer with new initial time period.  
Restart a cyclic timer with new cyclic time period.

MAXQ is a Trademark of Maxim Integrated Products

## A Few Customers of CMX...

- |                       |                    |                  |
|-----------------------|--------------------|------------------|
| ◆ AMD                 | ◆ ITT              | ◆ TRW            |
| ◆ IBM                 | ◆ AMP              | ◆ Enraf          |
| ◆ Sony                | ◆ Ford             | ◆ Xerox          |
| ◆ Baxter              | ◆ Boeing           | ◆ Siemens        |
| ◆ Philips             | ◆ U.S. Navy        | ◆ Rockwell       |
| ◆ TV/COM              | ◆ U.S. Robotics    | ◆ Kenwood        |
| ◆ Analog Devices      | ◆ AT&T Wireless    | ◆ Honeywell      |
| ◆ Fujitsu Telecom     | ◆ Temic Telefunken | ◆ ABB Power      |
| ◆ Ericsson Mobile     | ◆ Hewlett Packard  | ◆ Benefon OY     |
| ◆ Nokia Telecomm      | ◆ Oak Technologies | ◆ Allied Signal  |
| ◆ Lucent Technologies | ◆ Bose Corporation | ◆ Hughes Network |