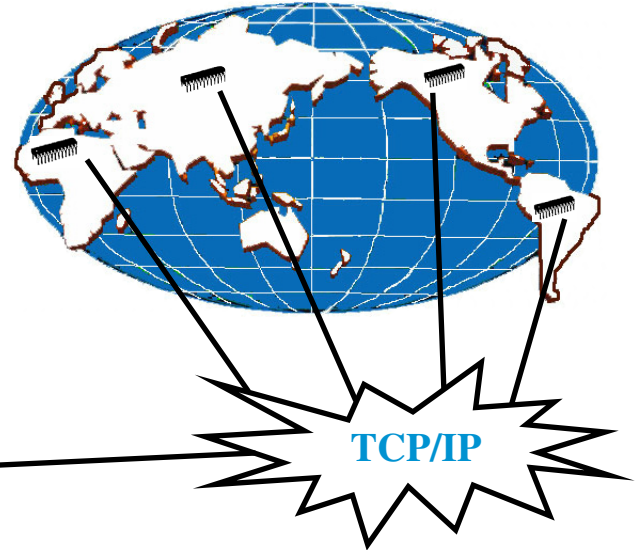
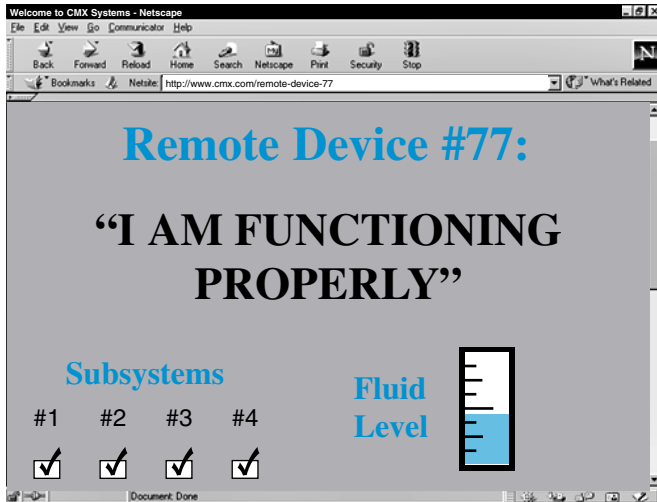




CMX-MicroNet™ for eZ80™

True TCP/IP Networking For ZiLOG eZ80 Acclaim Processors!



Finally, the *Right* Connectivity Solution for Your ZiLOG eZ80 Acclaim Processors (All Derivatives)

CMX-MicroNet has been developed by the company that is famous for providing complete, elegant solutions to the embedded community - CMX! Our developers have the expertise and hands-on experience to satisfy the most stringent real time demands that the 8-/16-bit community deals with every day. When we set about the task of creating the first true TCP/IP stack for these popular processors, we knew that it had to have:

Only Industry Standard Protocols. Of what benefit are closed, proprietary protocols that constrain and confuse your development team? CMX-MicroNet offers only industry standard protocols running right on your target processor and we provide full source code with every sale!

Use your Current Processor. Why should you have to upgrade your current processor, or, worst yet, add another processor just for TCP/IP? Those hardware costs can really add up! CMX-MicroNet allows you to work with your current design and still implement the networking connectivity you need.

An Affordable Pricing Structure. CMX-MicroNet offers a low, one-time fee and no royalties on deployed products. And you get the entire source code for free with every purchase!

Supported Protocols

- TCP
- PPP
- UDP
- SLIP
- IP
- HTTP Web Server
- DHCP
- FTP
- TFTP
- SMTP

Connectivity

- Ethernet
- Wireless Ethernet
- Dial Up
- Direct

Coming Soon!

- POP3

Much more...Let us Know What You Need

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

Tel: 904.880.1840 • Fax: 904.880.1632 • Email: cmx@cmx.com • WWW: <http://www.cmx.com>

CMX-Micronet Configuration Manager

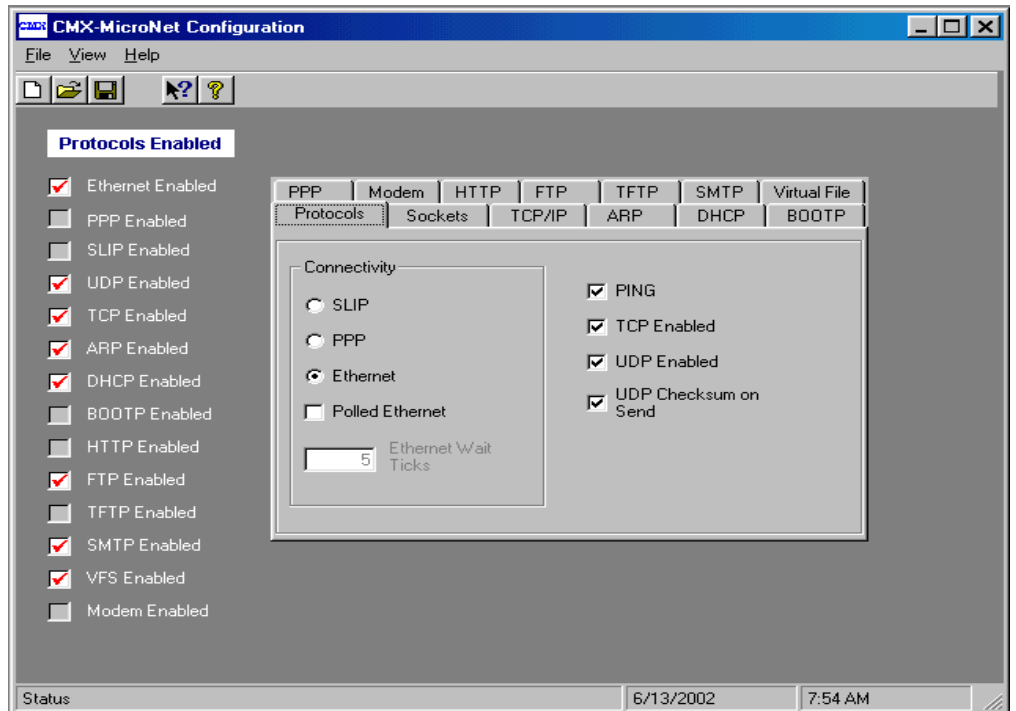
CMX-MicroNet Specifications for ZiLOG eZ80 Acclaim

ROM usage

UDP/IP + core	8960 bytes
TCP/IP + core	15297 bytes
UDP/TCP/IP + core	17107 bytes
PPP	9366 bytes
Modem	591 bytes
HTTP server	7180 bytes
Virtual file	1708 bytes
Ethernet	6503 bytes
DHCP Client	4862 bytes
FTP Server	6943 bytes
TFTP Client	1574 bytes
BOOTP	1611 bytes
SMTP	3069 bytes
Utility	2903 bytes

RAM (not including buffer sizes)

UDP/SLIP	71 bytes
TCP/HTTP/PPP	579 bytes
Ethernet	65 bytes



*CMX-MicroNet is easy to configure and integrate with your application.
Get your embedded processor networked FAST with CMX-MicroNet.*

Important Features of CMX-MicroNet

- **Tested and Proven with Hundreds of Design Wins Around the World**
- **Extremely Small ROM/RAM Requirements**
- **Supports All ZiLOG eZ80 Acclaim Processors**
- **Software Solution does not Require Additional Processor**
- **Web Pages May Contain CGI calls**
- **FTP Files, Including New Firmware**
- **Send Emails**
- **Can Serve up Java Applets**
- **No Proprietary Protocols**
- **Runs Stand Alone or with any RTOS**
- **Economical One Time Fee**
- **Full Source Code Provided**
- **No Royalties on Shipped Products**
- **Excellent Documentation and Support**

Do I Need an RTOS?

CMX-MicroNet has been specifically designed to run with or without an RTOS. If you choose to use CMX-MicroNet without an RTOS, as many customers do, it will run well in Polled Mode.

Of course, CMX-MicroNet has been tightly integrated with the CMX-RTX and CMX-Tiny+ RTOSes, should your networking application require the services of an operating system. But CMX-MicroNet also has an RTOS porting layer in the code in case you wish to use a different RTOS than one provided by CMX.

CMX believes that our users need this unusual development flexibility to ensure that they are developing the best networked products for their markets.



CMX-RTX™ for eZ80™

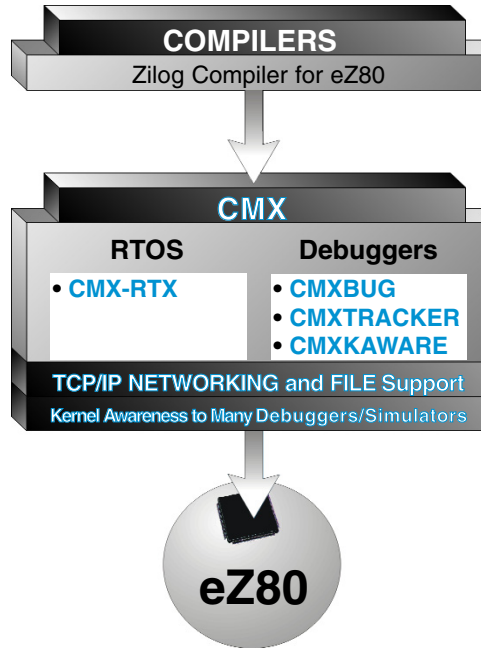
IS YOUR PROCESSOR IN NEED OF A REAL-TIME MULTI-TASKING OPERATING SYSTEM?

- ◆ 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:

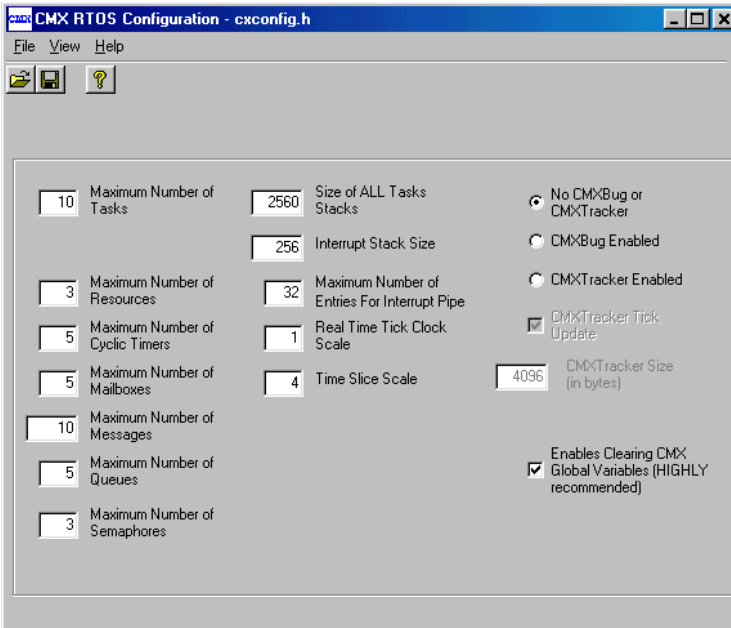
- ◆ The smallest footprint
- ◆ The fastest context switch times
- ◆ The lowest interrupt latency times
- ◆ True preemption
- ◆ Scheduler and interrupt handler written in assembly for speed and optimization
- ◆ Optional co-operative and time-slicing scheduling
- ◆ Nested interrupts
- ◆ All functions contained 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 RTOS Configuration Manager



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

CMX-RTX Specifications for eZ80

ROM

All CMX Functions: 7907 bytes
 CMX Initialize Module: 1747 bytes
 CMX Assembly Module (scheduler): 485 bytes

RAM, Each Task Control Block: 37 bytes

Min. Context Switch: 425 cycles (starting a task)
 445 cycles (resuming a task)

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.

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 |