PIPELINES

Covering Microware's Real-Time System Solutions

Volume 7 Number 1

Winter 1992

15

Y · E · A · R · S

microware®

PIPELINES
Winter 1992
Volume 7 Number 1

PIPELINES is published quarterly
by:
MICROWARE SYSTEMS CORPORATION
1900 NW 114th Street
Des Moines, Iowa 50325-7077
(515) 224-1929

Publication Coordinator:
David F. Davis

Editor:
Steve Simpson

Art Director:
Polly Beaver

Microware Contributors:
Jeff Ames, Warren Brown
Drew Crane, Kristin Doane
Kevin Erwin, Kristi Kramersmeier
Mike Maher, Nick Rainey
Ric Yeates
Stephen Montgomery

Photography:
David F. Davis

Celebrating 15 Years

MICROWARE IS CELEBRATING ITS FIFTEENTH anniversary in 1992. We're proud to be one of the oldest system software houses in the world. This issue of *PIPELINES* is dedicated to the past and the future.

Table of Contents

Looking Back, Looking Forward

An Interview with Ken Kaplan

15

YEARS

THIS YEAR, MICROWARE SYSTEMS Corporation celebrates 15 years of real-time system software development. To commemorate this milestone, PIPELINES talked with Ken Kaplan, president of Microware, about the past and the future of Microware.

PIPELINES: *How did Microware get started?*

KEN: A friend and I were attending Drake University here in Des Moines when we received a research grant from the National Science Foundation to work with first-generation microprocessors. The work involved writing software for experimental versions of Motorola's 6800. And this led to developing some of the first real-time embedded control applications.

We noticed that some of the software applications we wrote had common parts. It then occurred to us that we could develop tools and an operating system to handle those common parts. This led us to develop the first ROMed real-time oper-

"Pick up any trade journal and there's talk about a new 'standard.' The term 'standard' is applied to everything. Some are honest, independent standards. Others are politically-motivated by vendors to create interest in their products."

ating system for a microprocessor—RT/68. Though it was simple by today's standards, RT/68 allowed up to 16 tasks with rudimentary functionality for starting and stopping tasks, as well as handling interrupts.

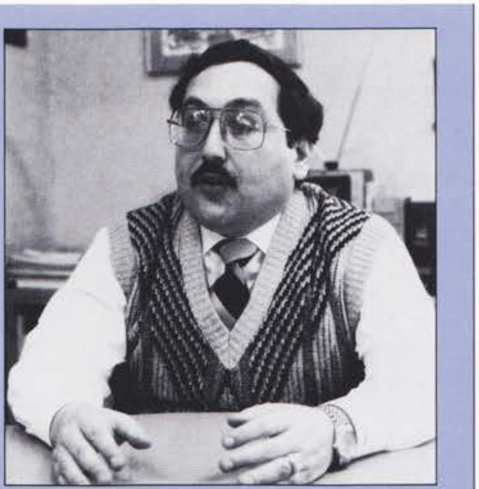
We put a small ad in *BYTE* magazine—*BYTE*'s second issue I think—to sell RT/68. We were working out of a spare bedroom and only had a P.O. box for a mailing address. To our amazement, orders started coming in. It seemed that there were a lot of people out there that needed an 'RT/68 solution'—and Microware was born! Later that year, we were able to move to a small office near Drake.

"After 15 years, I honestly am enjoying my job. The early years were rough. The last few have been good, exciting and growing years. I'm looking forward to what the future holds in store for Microware."

We also saw the need for a specialized real-time programming language. Most programmers in those days were writing code in assembler. We looked at BASIC as a good language and developed an industrial real-time version—A/BASIC.

PIPELINES: *After you got started in Des Moines, did you ever think about setting out for Silicon Valley?*

KEN: Not really. Our connection to Drake helped us get our start in Des Moines. We stayed here because we like the high quality of life and low cost of conducting business. Plus, we could tap into the resources at Drake, Iowa State University and the University of Iowa.



"We took a risk, and we began developing software for Motorola's 68000 chip on the VME bus. In the long run, this turned out to be a wise choice. It was the dawn of OS-9/68000."

From the first *BYTE* ad, we began doing business around the world. As business grew, we also saw business coming from Japan and Europe. We had dealers around the world that began offering our products. In some areas, it became apparent that direct presence was needed by Microware to provide local sales and, especially, support.

Over the years, we continued to expand our presence around the world with offices in Silicon Valley, in Japan, in England and in France. Currently, 40 percent of Microware employees work outside Des Moines. Our business is still fairly equally spread between our three major market areas: North America, Japan and Europe. And Des Moines gives us a good central location to serve customers in all of these markets.

PIPELINES: *How did OS-9 come out of the early success of RT/68?*

KEN: OS-9 was an original—it wasn't the 'new version' of RT/68. Motorola was beginning to design the 6809 chip, a revolutionary 8/16-bit microprocessor. Motorola knew the 6809 would be well-suited for an advanced programming language. Because of our previous experience with Motorola, they

LOOKING BACK, FORWARD
Please turn to Page Ten

NEW PRODUCT

OS-9 Now Available for MVME 167

MOTOROLA'S MVME 167 SINGLE BOARD computer provides phenomenal power with on-board serial, SCSI and Ethernet hardware. The **OS-9/167 Development Pak** takes full advantage of the board's MC68040 microprocessor and the new peripherals.

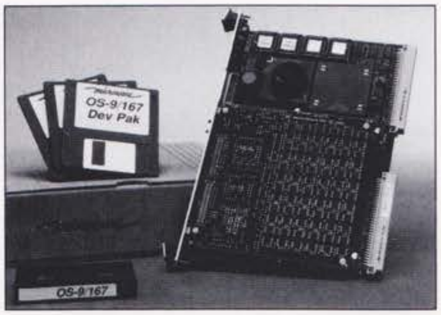
Load-and-Go Package

The OS-9/167 Development Pak is a pre-configured version of OS-9/68040 that is designed for easy installation, and quickly turns the MVME 167 into a multi-user, multi-tasking development system. The Development Pak installs in minutes and is easy to customize for unique application requirements.

In addition to the OS-9/68040 Real-Time Kernel, the Development Pak includes five field-proven I/O file managers and device drivers which support the advanced on-board peripherals.

Get Real-Time Power

The OS-9/167 Development Pak turns your MVME 167 into a powerful real-time development platform. To order a Development Pak or for more information, contact Microware or your authorized Microware representative. MSC



OS-9 includes support for all on-board devices of the MVME 167 single board computer.

From Missiles To TV Dinners

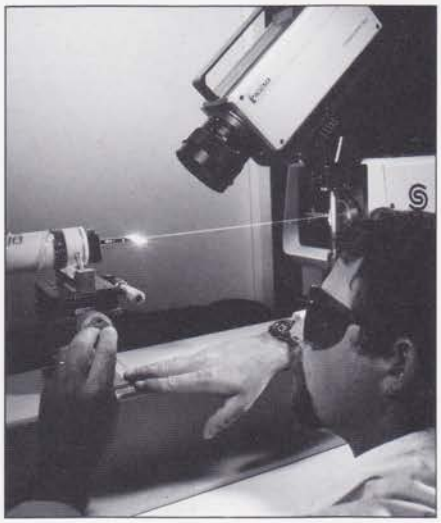
IR Imaging System Provides Expanded Real-Time Analysis Capabilities for Scientists and Engineers

by Gabriel von Friesendorff
AGEMA *Infrared Systems* AB

THE APPLICATIONS FOR INFRARED (IR) thermography, the imaging of the invisible IR energy radiated by all objects, are becoming more and more diverse and putting increasing demands on available technology. Although AGEMA (Danderyd, Sweden) introduced the first commercial IR scanners in the mid-60s, the real advances in system speed and sophistication, accuracy and analytical capabilities for this non-destructive thermal measurement technique have taken place in the last ten years or so, with the advent of relatively low-cost computing power.

And nowhere has the increase in IR imaging—whose early success was in electric utility, plant maintenance and medical diagnosis—been greater than in the area of research and development. Infrared systems can collect, measure and store more data more quickly and over a greater temperature range than other methodologies. In addition, because it is non-contact, IR can be used where it is impossible or impractical to use other techniques and can gather data without affecting the temperature of the object it is measuring.

Most recently, of course, infrared earned its stripes in the short-lived Gulf War, and was, in fact, itself largely responsible for the war's brevity. IR-equipped aircraft and tanks traced their Iraqi targets with unerring accuracy.



Engineer at Spectra-Physics Lasers uses Thermovision 900 to inspect surface of "Brewster Window," a critical component on a laser.

The accuracy of the missiles they deployed was due in large part to the data encoded on chips in the minicomputers which guided them. That data—a thermal profile or signature of their target—was gathered using infrared imagers.

Engineers and scientists are using state-of-the-art IR thermal imagers in a host of non-military applications, for example to:

- Design and evaluate semiconductors and advanced PCBs.
- Design and monitor the manufacture of laser optics.
- Evaluate the performance of device and system heat sinks in computer design.
- Evaluate and test new materials and composites, such as are used in critical aircraft and automotive parts.
- Design the electrical system on the proposed Space Station Freedom.
- Ensure the uniform microwave heating of frozen dinners.

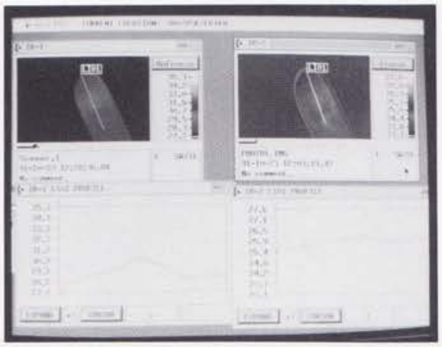
Relatively new as a technology or not, IR imaging long ago advanced from the "gee whiz, look at the pretty pictures" phase to the "how can I put all that information to good use" phase. In fact, many researchers are less interested in the IR thermal images themselves than the measurement data they "portray."

> *"OS-9 combines with X Windows to provide our research and development users with the industry's first multi-tasking workstation for infrared thermal data analysis."*

Thus, when designing our newest R&D system, the Thermovision 900, our objective was to create a system that achieved the highest possible image resolution and accuracy. We also wanted a system that captured and transmitted that data for immediate or later analysis with full data

AGEMA Infrared Systems' new Thermovision 900 thermal measurement and analysis system provides high image quality and data acquisition rates with sophisticated data manipulation and analytical capabilities, as well as ease of use. Photos courtesy of AGEMA.

Real-time analysis of the thermal image of a Brewster Window is facilitated by OS-9 and X Windows. In this photo, the operator compares a live image with a stored image of a component with an acceptable thermal profile.



Detail of screen from photo at left.

integrity and enabled the user to access and manipulate that data in real time if necessary.

Infrared systems usually rely on some sort of control/recording unit to digitize and store their scanner's signal. Until the Thermovision 900, however, these system controllers downloaded their live or stored information to a PC for limited image processing and analysis, very little of which could be done in real time due to the restrictions of the microprocessors and the PC's operating system.

With the Thermovision 900, analysis is done on the controller itself. The added responsibility was assigned to the controller because no currently available PC and few workstations can handle the high data rates of the system's new scanners, which pre-digitize their signal at 12 bits prior to transmission.

The dedicated system controller's two parallel channels allow simultaneous, real-time presentation and analysis of two images from either one scanner or two separate scanners on the same display, or direct comparison of current and stored image data.

To deal with the demanding real-time requirements imposed on the system—and in order to handle the scanners' full, 12-bit dynamic signal range—we selected OS-9 as our only option for the operating system and X Windows as our user interface. X Windows was ported to the OS-9 environment by our own team of system programmers.

For implementation, we rely on two Motorola MC68020 processors, both running under OS-9. One of the CPUs implements the X Window Server, and manages the presentation of live images (in color or black and white) and their associated measurement results on an SVGA display. The other processor handles data acquisition, thermal calculations and file system operations.

The combination of VME-bus architecture, OS-9 and X Windows provides us with the industry's first multi-tasking IR workstation, with all scanner operations, including remote focus and level setting, as well as measurement and analysis functions controlled from a mouse or keyboard. Built-in Ethernet allows two-way communication and control from a remote computer.

Gabriel von Friesendorff is Manager of Application Software at AGEMA Infrared Systems AB (Danderyd, Sweden) and was responsible for software development for the Thermovision 900 system. Mr. von Friesendorff, who holds a Master of Science degree in electronics from the Royal Institute of Technology in Stockholm, started as a hardware designer before concentrating on software development. When not developing software for AGEMA, he enjoys jogging in the summer and downhill and cross country skiing in the winter. Thermovision is a registered trademark of AGEMA. MSC

NEW PRODUCT

OSF/Motif Available for OS-9/X Window System

THE OPEN SOFTWARE FOUNDATION'S MOTIF graphical user interface package is one of the most popular user environments for X Window applications. OSF/Motif supplements the standard MIT X Window System development package with new X libraries and higher-level development tools. The Motif toolset simplifies the creation of X applications that feature 3-D screen appearance and standardized application behavior.

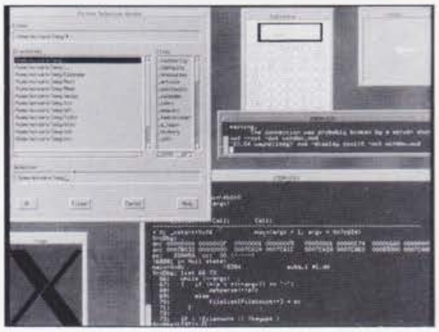
Two Packages Available

The **OS-9/Motif Development Pak** contains the Motif libraries and tools required to create and run Motif-based X applications. The resulting applications run under OS-9 and can display their output on any local or remote X Server.

The **OS-9/Motif Run-Time Pak** allows users to run the Motif Window Manager on an OS-9 system and display Motif-based applications. The package does not include the Motif application development tools.

Get the Power of X

OSF/Motif products are now available for the OS-9/X Window System. To order or for more information, contact Microware or your authorized Microware representative. MSC



Motif simplifies the creation of X applications with a 3-D appearance.

NEW PRODUCT

Introducing MShell: An Advanced Shell for OS-9

MSHELL IS AN ADVANCED COMMAND LINE interpreter and procedure file processor for OS-9. MShell is fully compatible with other Shells from Microware, yet offers new features to improve programmer efficiency.

New Features

MShell's improved command line interface includes history maintenance and recall, command line editing, and command and pathlist completion for more efficient interaction with OS-9. Repetitive typing is reduced by the use of command history recall.

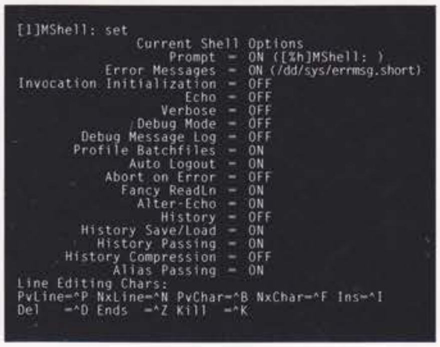
New command line options include the ability to pass parameters between MShells, as well as debug procedure files.

New UNIX-style commands are available (***pushd***, ***popd*** and ***dirs***) to simplify movement within directory structures.

MShell also features a powerful programming language that is a hybrid of C and BASIC. MShell's programming language makes it easy to build sophisticated procedure files to automate a variety of development activities.

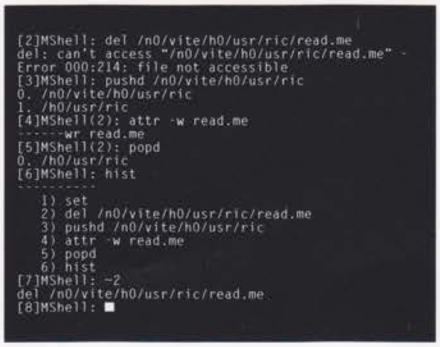
Now Available

MShell is now available for OS-9/680X0 systems. To order your copy, contact Microware or your authorized Microware representative. MSC



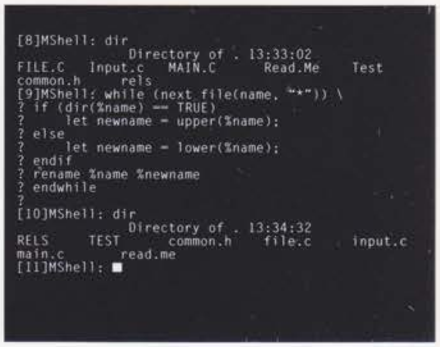
```
[1]MShell: set
               Current Shell Options
                   Prompt = ON ([%h]MShell: )
           Error Messages = ON (/dd/sys/errmsg.short)
Invocation Initialization = OFF
                     Echo = OFF
                  Verbose = OFF
               Debug Mode = OFF
        Debug Message Log = OFF
       Profile Batchfiles = ON
              Auto Logout = ON
           Abort on Error = OFF
              Fancy ReadLn = ON
               Alter-Echo = ON
                  History = OFF
        History Save/Load = ON
          History Passing = ON
      History Compression = OFF
            Alias Passing = ON
Line Editing Chars:
PvLine=^P NxLine=^N PvChar=^B NxChar=^F Ins=^I
Del   =^D Ends  =^Z Kill  =^K
```

The ***set*** command lists the current MShell options and command line editing characters. All options and editing keys can be changed for particular needs.



```
[2]MShell: del /n0/vite/h0/usr/ric/read.me
del: can't access "/n0/vite/h0/usr/ric/read.me" -
Error 000:214: file not accessible
[3]MShell: pushd /n0/vite/h0/usr/ric
0. /n0/vite/h0/usr/ric
1. /h0/usr/ric
[4]MShell(2): attr -w read.me
------wr read.me
[5]MShell(2): popd
0. /h0/usr/ric
[6]MShell: hist
----------
   1) set
   2) del /n0/vite/h0/usr/ric/read.me
   3) pushd /n0/vite/h0/usr/ric
   4) attr -w read.me
   5) popd
   6) hist
[7]MShell: ~2
del /n0/vite/h0/usr/ric/read.me
[8]MShell: ■
```

An error is reported when trying to delete a file over OS-9/NET. ***Pushd*** pushes the directory onto the stack allowing local change of the file attributes. ***Popd*** returns to the original directory. ***Hist*** lists the current command line history. By typing tilde (~) and a history number, that command line is executed. In this case, the original deletion is re-executed.



```
[8]MShell: dir
                 Directory of . 13:33:02
FILE.C   Input.c   MAIN.C       Read.Me     Test
common.h       rels
[9]MShell: while (next file(name, "*")) \
? if (dir(%name) == TRUE)
?      let newname = upper(%name);
? else
?      let newname = lower(%name);
? endif
? rename %name %newname
? endwhile
?
[10]MShell: dir
                 Directory of . 13:34:32
RELS     TEST      common.h    file.c      input.c
main.c       read.me
[11]MShell: ■
```

In this case, the programmer created a temporary MShell ***batchfile***. By typing a backslash (\) at the end of a command, MShell begins a batchfile editor—lines that start with a question mark (***?***). MShell's command line editing keys are available while creating a batchfile. Type a carriage return on a blank line to execute the batchfile.

Microware Co-Sponsors First U.S. CD-I Conference

by Jeff Ames
Microware Systems Corporation

ON OCTOBER 31, 1991, MICROWARE helped write an exciting new chapter in consumer electronics history as multimedia professionals from around the world gathered in Los Angeles for CD-I One. The two-day conference, which was the first ever held in the United States, was sponsored by Microware, along with Philips Consumer Electronics and Sony Corporation. CD-I One brought together over 700 compact disc-interactive (CD-I) developers, producers and publishers from the U.S., Europe and Japan.

Microware has been instrumental in the development of CD-I from its earliest beginnings. In 1985, Microware's unique operating system, OS-9, was selected by Philips and Sony as the exclusive operating system for CD-I players. That's why today you'll find Microware's CD-RTOS (Compact Disc Real-Time Operating System) at the heart of every player produced.

One of the highlights of the conference was Microware President Ken Kaplan's keynote speech. It offered attendees some exciting and perceptive insights into the future of CD-I. Among his predictions:

- Within the year, portable hand-held CD-I players will begin to appear in stores all over the United States.
- By the end of 1992 the price of CD-I players will be down to about $600.
- As the cost of CD-I players comes down, standard CD players will begin to disappear from the market (CD-I players can also play regular CD audio discs).
- Full Motion Video will be available in players at the beginning of 1993.

Microware's booth at CD-I One showcased CD-I development tools, our CD-I studio and MicroMall.

The conference also marked the official unveiling of Microware's new CD-I Publishing Group. The Group, with offices and studios located at Microware Corporate Headquarters in Des Moines, has already started to design, develop and produce consumer CD-I titles, or discs.

The studio will be working on games, music, children's programs, self-help programs and how-to materials this year. Since we developed CD-RTOS, Microware literally knows CD-I from the inside out. We plan to use that knowledge and expertise to produce exceptional consumer titles designed to motivate, educate and "interactive-ate" users.

Jeff Ames is the director of Microware's CD-I Publishing Group. MSC

Meredith and Microware Sign CD-I Co-Publishing Agreement

Meredith Corporation (Des Moines, Iowa) and Microware recently signed an agreement to co-publish compact disc-interactive (CD-I) titles. Under the agreement, Meredith Multimedia Interactive (MMI) will provide editorial and print material, video and other information from Meredith books, magazines and related products to Microware's CD-I Publishing Group.

Joe Ward, president of the Meredith Book group and MMI said, "Meredith products already touch the lives of millions every day. Now CD-I technology will enable families to 'experience' information and entertainment in a whole new way."

The first joint project will center around Meredith's *Max the Dragon* character with an outer space theme. Subsequent projects will include Max, as well as production assets from *WOOD* magazine and other Meredith publications.

Max the Dragon is a trademark of Meredith Corporation.

OS-9 IC Test System for Philips Semiconductor

by Philip Gilks
Origin IT

PHILIPS SEMICONDUCTORS (SOUTHAMPTON, England) was recently faced with an interesting challenge. To meet demand for testing popular integrated circuit (IC) products, Philips decided to design their own IC functional test systems. Origin IT (Redhill, England) had just completed a complex OS-9 control system for a large international drinks and confectionery company, and was chosen to undertake the software design.

Since it was Philips' intention to use this new semi-dedicated test system for testing unrelated families of products, such as teletext chips and digital-to-analog converters, it was important to choose a hardware design that was flexible and to design a common core of reusable control software.

For these reasons, it was decided to use VMEbus hardware for the wide variety of high quality cards available. OS-9 was chosen as the software platform due to its performance, its exceptional development environment and its comprehensive disk and I/O facilities. With time at a premium, the "off-the-shelf" availability of OS-9 drivers for many VME cards would not only shorten development time but also reduce development cost.

The test system is fully automated with integrated circuits fed to a test head continuously from a robotic handler. After testing, the product is sorted into bins depending on the results of the test. In addition to its normal mode of operation, quality engineers are permitted to perform additional checks.

The heart of the hardware is a Motorola MVME 147 single board computer with 4M RAM and running at 25 MHz. Other equipment in the rack includes IEEE-488 and RS-232 interface cards, A/D and D/A convertors, digital I/O, a digital signal generator and a waveform analyzer for fast fourier analysis of the analog output of the product.

Origin's integrated circuit (IC) functional test system for Philips Semiconductor.

The software is comprised of a multi-tasking test executive responsible for scheduling sequences of tests on the product. Results from these tests are stored on hard disk and sent to a host computer via a TCP/IP Ethernet network.

The type of tests, the pass/fail limits, the choice of chip stimuli and other test parameters are designed to be user-definable through an ASCII text file. This means that a certain degree of flexibility is given to the various users of the system.

The user interface was designed to be easily changed to allow for product differences. This is important so that users can easily transfer between variants of the test system.

Analog characteristics such as total harmonic distortion, digital noise and channel matching are best performed by a waveform analyzer. This presented a problem as it was desirable to use the same analyzer that was used for bench testing of these devices. However, the analyzer is driven from a PC by an operator who configures and selects tests from a menu. In addition, the analyzer software only accepts direct keyboard input.

A VME-based PC from XYCOM provided the solution with manipulation of a DOS keyboard buffer coming from the MVME 147 OS-9 system through the VME backplane.

The analyzer software allows the spawning of DOS shells. A DOS program was designed that copies the results to the PC's extended memory and then generates a VME interrupt to instruct the OS-9 system to fetch and process the data. A simple OS-9 device driver was written to handle the incoming interrupt and to signal the test process which reads the results from the shared memory on the PC.

The success of this project has meant that Philips has not only been able to increase the amount of product tested, but also reallocate expensive flexible analog test equipment to products with shorter production runs.

Philip Gilks is a senior designer with Origin IT (Redhill, England). Acknowledgments to Mike Smith and his team at Philips Semiconductors, David Knowles for his work on the OS-9/PC interface, John Davey, Julian McEntegart and Peter Ba Han. MSC

Powerful Tool for PLC Code Development

ISAGRAF FROM CJ INTERNATIONAL (Grenoble, France) provides a powerful, standards-based tool for the development of GRAFCET and Relay Ladder Logic (RLL) programmable logic controller (PLC) applications.

ISaGRAF gives designers the ability to develop PLC code on a personal computer, compile the code for a specific PLC, then download the application to an OS-9 or OS-9000 PLC. ISaGRAF allows PLC applications to use procedures or tasks written in C to enhance GRAFCET capabilities.

ISaGRAF features two levels of systems development: the project manager level and the user level. The significant feature at the project manager level is library management that provides function block definitions, I/O drivers and user procedures. The user level focuses on GRAFCET and RLL programming, compiling and debugging. MSC

CD-I Shown At Japanese Electronics Show

by Kristi Kramersmeier
Microware Systems Corporation

CD-I titles were demonstrated at the CD-I Consortium Japan exhibit.

The 30th Annual Electronics Show was held October 1-5, 1991, in Tokyo, Japan. Leading consumer electronics companies from around the world displayed their latest product developments to nearly 350,000 visitors.

Among the most revolutionary products at the Electronics Show were those relating to CD-I. Stand-alone CD-I systems could be found at numerous exhibits including those of Sony, Matsushita, Philips, Pioneer and Sanyo, to name only a few.

Over a dozen CD-I players and numerous CD-I application titles were brought together under one roof at a booth sponsored by the CD-I Consortium Japan. The Consortium was formed last year to coordinate the development of the CD-I industry in Japan and currently has over 700 members from 216 companies.

Microware Systems K.K., Microware's Japanese subsidiary, participated in the Electronics Show with CD-I title demonstrations at the Consortium's CD-I Plaza Stage. The Microware K.K. CD-I team used the extensive title development support libraries of CD-RTOS/RAVE to create an entertaining tour guide called *I Love the Earth*. This title caught the attention of many passers-by with its clever combination of 3-D graphics, still-frame and full-motion pictures.

The CD-I Consortium Japan booth included the latest in CD-I technology.

The Electronics Show was yet another glimpse into the important role CD-I will play in the future of consumer electronics.

Kristi Kramersmeier is Microware's Asian Marketing Coordinator and spends most of her time at Microware K.K. in Tokyo. MSC

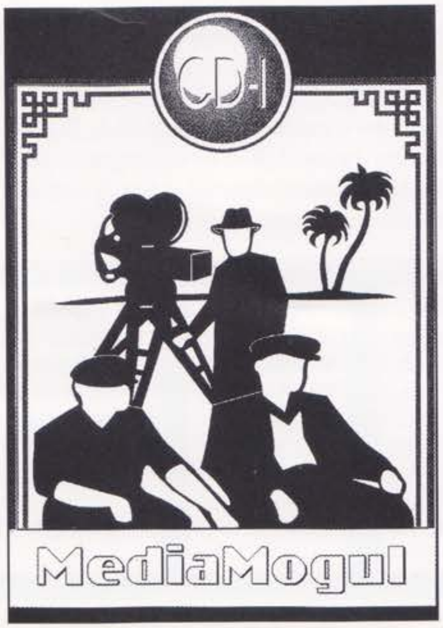
OptImage to Release New Version of MediaMogul

by Craig Rispin
OptImage Interactive Services Co.

OptImage (Des Moines, Iowa) is about to release an update to **CD-I MediaMogul**, the CD-I authoring tool designed for non-programmers. This upgrade adds significant new features such as movies in a window and touch screen support. Previously, these and other new features required custom programming on the part of the user. Now, OptImage's customers can access even more of the multimedia capabilities of CD-I without programming.

CD-I MediaMogul is the most widely used "no-programming" CD-I authoring system in the world. Typical customers are using MediaMogul for point-of-sale kiosks, title prototyping, multimedia presentations and simple training applications.

Other new features to CD-I MediaMogul include:



- Play Sync—synchronizes video files to audio tracks from user's mouse clicks.
- Color Cycling—adds "weather map" cycling effects to graphics.
- Improved Interface—allows for faster editing.

In addition to new features for non-technical users, OptImage has updated MediaMogul's features for programmers with support for Balboa subroutines and standard real-time files.

Training Available

OptImage conducts CD-I MediaMogul training classes on the last Monday of every month in Des Moines. Classes are also periodically scheduled in major cities across the United States.

For more information about CD-I MediaMogul and training, call OptImage at 1-800-CDI-5484 (toll-free, U.S. only) or (515) 225-7000. OptImage's new address is Suite 100, 1501 50th Street, West Des Moines, Iowa 50265-5961.

Craig Rispin in the Channel Manager for OptImage. MSC

LOOKING BACK, FORWARD

Continued from Page Three

came to Microware and asked us to develop that language. As we began developing what would eventually become BASIC09, we realized the need for a more powerful operating system. We wanted to incorporate sophisticated I/O, multi-user and multi-tasking capabilities.

OS-9 involved revolutionary concepts and designs not previously found in operating system software. UNIX was recognized in research circles as powerful system software and we began studying the underlying concepts of UNIX. We took the best concepts of UNIX and applied them to real-time design. We developed a revolutionary modular design and, after a lot of work, OS-9/6809 was born.

PIPELINES: *What were some of the early bus platforms OS-9/6809 could be found on?*

KEN: Some people may remember that the SS-50 bus and ExorBus were very successful for 6800 and 6809 real-time control systems. The SS-50 started as part of a kit computer, before PCs were prevalent. Home users started to find ways to use the bus for industrial control systems. However, as microprocessors moved beyond 8-bit technology, it became apparent that the SS-50 bus and ExorBus would not make the transition. Microware started looking at 16- and 32-bit technology and the upcoming VMEbus design. We took a risk and we began developing software for Motorola's 68000 chip on the VMEbus. In the long run, this turned out to be a wise choice. It was the dawn of OS-9/68000 for 16- and 32-bit microprocessors and microcontrollers.

PIPELINES: *Tandy began offering OS-9 on the Color Computer III. How did this come about?*

KEN: Microsoft offered a ROMed BASIC and DOS for the CoCo III. But the CoCo III begged for more sophisticated system software to use the power of this 6809-based computer. After being introduced to Tandy by Motorola, Microware was contracted to port OS-9 to the CoCo III. We invested about a year of development in the project and we were very satisfied with the result. We're proud of the fact that OS-9 became a sort of "cult classic" with CoCo users. A lot of third-party OS-9/6809 software was developed as a result of the CoCo III project, along with a very active user's group.

PIPELINES: *Ken, you mentioned the emergence of the VMEbus. How has the growth of VME influenced Microware?*

KEN: The VMEbus has achieved strong growth in North America, Japan and Europe for real-time systems. Almost every VME manufacturer sells OS-9/68000. We're pleased with the past growth of VME and anticipate continued use into the foreseeable future.

PIPELINES: *Microware later formed a partnership with Philips and Sony to develop compact disc-interactive or CD-I. How did this partnership come about, and how has it affected Microware?*

KEN: Philips had been a customer of Microware's for several years. We had worked with them on several projects, including intelligent products for the home that were predecessors to CD-I. When Philips began CD-I development, they evaluated more than 60 other operating systems and kernels.

Microware won out for several reasons. First, Philips recognized that the audio/video features of CD-I required real-time processing power. Multimedia applications had to deal with real-time processing; much more than any PC could ever provide. Second, OS-9 offered price/performance and size/performance advantages that other system software manufacturers couldn't match. Finally, Microware had built a suite of off-the-shelf development tools like compilers, editors and networking software. Philips knew that emerging CD-I studios would need powerful development tools to create sophisticated applications for consumers.

But here is the key to our success with CD-I. Early on, we separated CD-I development from our core real-time business. We recognized that advanced 32-bit real-time system software was our primary business and we didn't want resources taken from this core area. Just the opposite is happening, in fact.

PIPELINES: *Has CD-I had other effects on Microware's business?*

KEN: It certainly has. CD-I has also provided some additional products to our core real-time market. The first derivative product was RAVE [Real-Time Audio/Video Environment], an industrial graphical user interface for non-technical users. RAVE was a direct result of our work in CD-I. Less direct descendents include improved development tools and support for optical disks.

In addition to our development work on CD-I, Microware has created three separate business units to meet specific needs in this emerging new media area. Microware's CD-I Publishing Group provides publishers with multimedia studio services for the development of consumer CD-I

Timeline showing 15 years of real-time innovation at Microware. Major product releases and market developments are shown.

1977	1978	1979	1980	1981	1982	1983
Microware Incorporated RT/68 Released	A/BASIC Compiler and LISP Interpreter Released Motorola Basic Development License Signed	OS-9/6809 Development Begins	OS-9/6809 and BASIC09 Released Foreign Distributors Established	Pascal Compiler Released C Compiler Development Begins	OS-9/68000 Development Begins Major PC Deals Signed with Tandy, Fujitsu and Hitachi	OS-9/68000 Released 6809 C Compiler Released Significant Third-Party Software Made Available

titles. MicroMall, Inc. is a separate entity that is applying CD-I technology to point-of-information and point-of-sale delivery. OptImage Interactive Services Company is a partnership with Philips to provide hardware and software development systems for CD-I.

These separate entities put us at the forefront of CD-I, yet they allow us to keep our focus on customers in our real-time system software business.

PIPELINES: *What trends in computer technology have most affected Microware?*

KEN: There are three advances that have driven the way we do business and the way we develop products.

First is the phenomenal advancement of the power of microprocessors. Chips have moved from simple controllers to powerful, self-contained computers in a very short time. Microprocessors can now handle networking, graphics and more.

The proliferation of PCs has also had an effect on our products. First came the need for Microware to provide a cross development solution as more engineers put more PCs on their desks. We satisfied that need with PCBridge. Then came the need for an integrated development and industrial environment for PCs. Microware developed OS-9000 with an eye toward the 32-bit Intel CISC processor line and advanced CISC and RISC microprocessors. This gave engineers complete development and embedded solutions for PC hardware.

Finally, our decision early on to follow UNIX has paid off. Computer science students are literally raised on UNIX in universities around the world. This has led to a familiarity with the concepts behind OS-9, as well as the need for Microware to provide UNIX-based development solutions. UniBridge and other products are specifically designed to give UNIX users powerful options for cross development.

Throughout, Microware has used these changes to its advantage rather than fighting trends. This has led to many opportunities as our business expands. It has also kept us in touch with our customers and trends in the real-time marketplace.

PIPELINES: *What issues are facing the real-time marketplace as we move toward the year 2000?*

KEN: Two areas will profoundly affect our market in the coming years: standards and increasingly powerful *integrated* development tools.

Pick up any trade journal and there's talk about a new 'standard.' The term 'standard' is applied to everything. Some are honest, independent standards. Others are politically-motivated by vendors to create interest in their products. Microware makes a point of identifying true standards that are applicable to engineers, then supporting those standards. This is evidenced by our involvement in POSIX, ANSI C, NFS, X Windows and the Open Software Foundation. In fact, we have a representative on the POSIX committee. Standards will continue to help Microware deliver products that are of the highest quality and of the most use to our customers.

As more powerful hardware is developed and applications become more sophisticated, it is going to become increasingly important to provide sophisticated, fully-integrated development tools for real-time design. The days of sourcing compilers, debuggers, GUIs and networking software from different vendors are coming to an end. This means creating a cohesive package from a suite of tools that exceed the current state of the art; a package that will dramatically increase the efficiency and effectiveness of engineers. This requires more than repackaging existing software. It requires careful planning to integrate diverse tools, as well as enhance the power of these individual tools.

Microware's UniBridge and PCBridge represent first-generation products for integrated cross development. We'll soon offer the next generation of products that push the envelope for seamlessly integrated, standards-based development past anything currently offered by the industry. The packages will help make real-time development more efficient in order to allow engineers to bring products to market more quickly.

PIPELINES: *What does the future hold?*

KEN: After 15 years, I honestly am enjoying my job. The early years were rough. The last few have been good, exciting and growing years. I'm looking forward to what the future holds in store for Microware.

Our primary goal is to make some of the best, most advanced software in the world.

Finally, Microware will develop more partnerships around major new technology trends, particularly in the telecommunications area. These partnerships will be built around our core real-time business area, but will help us take a leadership position in multimedia, network and advanced telecommunications arenas.

It's been an exciting 15 years. We've met the challenges put before us and moved to the front of the real-time system software industry. Our challenge for the next 15 years is to use our leadership role and advance the art of real-time engineering. MSC

1984	1985	1986	1987	1988	1989	1990	1991
Established Japanese Subsidiary Signed Philips Consumer Products License	Green Book and CD-RTOS Projects Begin 68000 Pascal Compiler Released High-Level Graphics Software Released	Philips, Sony and Microware CD-I Project Announced 68000 Fortran Compiler Released	OS-9000 Development Begins Established U.K. Subsidiary VME Market Leadership Recognized	Ethernet TCP/IP Software Released UniBridge and PCBridge Released OptImage Venture Formed with Philips	OS-9000 Released RAVE Released Established Microware France and New Japanese Subsidiary	OS-9/N.F.S. Released ISDN Project Announced	OS-9/X Window System Released MicroMall Formed

ON THE C SIDE

Signals and User-State Alarms

by Ric Yeates
Microware Systems Corporation

Signals

SIGNALS ARE A METHOD OF INTERPROCESS COMMUNICATION IN OS-9. They are sometimes called "software interrupts" because they are handled in much the same way as hardware interrupts. Generally, when your process receives a signal it stops what it's doing and executes a signal handler. When the handler finishes, execution resumes where the process was before it received the signal. Signals provide an excellent method of dealing with data and events in real time.

For more information, see the following manuals:

- *OS-9 Technical Manual*: Interprocess Communication (Signals), ***F$Send***, ***F$Icpt*** and ***F$SigMask***.
- *OS-9 C Compiler Library*: ***intercept()***, ***kill()*** and ***sigmask()***.

User-State Alarms

User-state alarms allow a process to arrange to have itself sent a signal at a given time, after a period of time or cyclically. This provides a good method to determine if a particular event has occurred before a time limit. They are also very useful when various tasks have to be accomplished at different time intervals.

For more information, see the following manuals:

- *OS-9 Technical Manual*: Interprocess Communication (User-State Alarms) and ***F$Alarm***.
- *OS-9 C Compiler Library*: ***alm_atdate()***, ***alm_atjul()***, ***alm_cycle()***, ***alm_delete()*** and ***alm_set()***.

_ss_ssig()

Often, a process must allow input from the user and also update a display or monitor an external device at the same time. This can be accomplished using the ***_ss_ssig()*** call for the device. ***_ss_ssig()*** tells the I/O system to lock the device and to send you a signal when data arrives. For example, if you were expecting input from either a pipe or a serial port, you could use ***_ss_ssig()*** on both devices with different signal numbers. Then, simply put your process to sleep until a signal arrives. Based on the signal that arrives, you can read and handle the available data.

The following program uses signals and alarms to implement timed input and output. It will also demonstrate how to wait for input and do output at the same time with the ***_ss_ssig()*** function.

The program does the following:

  - sets thee intercept function to the signal handler
- ***Timed Output***
  - sets an alarm to go off in 5 seconds
  - starts an endless loop of output that can only be stopped by a signal
  - deletes the possibly pending alarm
  - prints a message about how output was stopped
- ***Timed Input***
  - prints a message asking for a line of input in 10 seconds
  - sets an alarm to go off in 10 seconds
  - calls the library to get a line of input
  - deletes a possibly pending alarm
  - prints a message about how the input was stopped
- ***Single Character Timed Input With Output***
  - sets up a 1 second cyclic alarm
  - enters a loop that terminates after 10 seconds or when a termination signal is received
  - deletes the cyclic alarm

Compiling the Program

1. Type the program into a file called ***timed.c***.
2. Type *cc* ***timed.c***.
3. Run the program by typing ***timed*** at a Shell prompt.

```
#include <stdio.h>
#include <signal.h>
#include <time.h>
#include <errno.h>

#define TRUE 1
#define FALSE 0

#define SIG_TIMEOUT       30       /* I/O fatal signal */
#define SIG_GOTCHAR       1000
#define SIG_CYCLE         2000

int stop;                 /* I/O was stopped by user */
int timeout;              /* I/O was stopped by alarm */
int gotchar;              /* character received from user */
int secs;                 /* number of times cyclic alarm
                             expired */
```

Continued

```
void no_echo_read();

void sig_hand(signal)
register short signal;          /* received signal */
{
    switch (signal) {
        case SIG_TIMEOUT:
            timeout = TRUE;     /* timeout signal */
            break;
        case SIGINT:
        case SIGQUIT:
            stop = TRUE;        /* user termination signal */
            break;
        case SIG_GOTCHAR:
            gotchar = TRUE;     /* character in signal */
            break;
        case SIG_CYCLE:
            secs++;             /* another second */
            break;
        default:
            exit(signal);
    }
}

main()
{
    int alm_id;
    char name[128];

    /* set up signal handler */
    intercept(sig_hand);

    /* do timed output */

    /* 5 seconds worth of output */
    stop = timeout = FALSE;
    if ((alm_id = alm_set(SIG_TIMEOUT, CLK_TCK * 5)) == -1)
        exit(_errmsg(errno, "can't set alarm - "));

    while (!stop && !timeout)
        printf(
        "Press ^C or ^E to stop output or wait 5 seconds!!!\n");

    /* delete possibly pending alarm */
    if (alm_delete(0) == -1)
        exit(_errmsg(errno,
                    "can't delete all pending alarms - "));

    /* clear the error off standard output */
    clearerr(stdout);

    if (stop) {
        printf("\nUser stopped output.\n");
        no_echo_read();
    }
    else
        printf("\nOutput timed out.\n");

    /* do timed input */

    printf("Enter your name in less than 10 seconds: ");
    fflush(stdout);

    stop = timeout = FALSE;

    /* set 10 second alarm */
    if ((alm_id = alm_set(SIG_TIMEOUT, CLK_TCK * 10)) == -1)
        exit(_errmsg(errno, "can't set alarm - "));

    gets(name, 128, stdin);

    clearerr(stdin);

    /* delete possibly pending alarm */
    if (alm_delete(0) == -1)
        exit(_errmsg(errno,
                    "can't delete all pending alarms - "));

    if (stop) {
        printf("\nUser stopped input.\n");
        no_echo_read();
```

Continued

```
    }
    else if (timeout)
        printf("\nInput time out.\n");
    else
        printf("Thanks %s, that was close!\n", name);

    /* do timed single character input with output
        to same device */

    stop = timeout = FALSE;

    if (_ss_ssig(fileno(stdin), SIG_GOTCHAR) == -1)
        exit(_errmsg(errno,
                "can't setup signal data ready signal - "));

    printf(
    "Enter a character before self-destruct in 10 seconds!\n");

    /* set 1 second cyclic alarm */
    secs = 0;
    if ((alm_id = alm_cycle(SIG_CYCLE, CLK_TCK)) == -1)
        exit(_errmsg(errno, "can't set alarm - "));

    while (!stop && !gotchar && secs != 10) {
        sleep(0);     /* wait for something to wake us up */
        if (stop)
            printf("\nUser abort.\n");
        else if (gotchar)
            printf("\nThank you and have a nice day.\n");
        else {
            if (secs != 10)
                printf("Self-destruct in %d second%s.\n",
                10 - secs,
                    ((10 - secs > 1) ? "s" : ""));
            else
                printf("Boom!!!\n");
        }
    }

    /* read single character of input */
    if (gotchar || stop) no_echo_read();

    /* delete pending cyclic alarm */
    if (alm_delete(alm_id) == -1)
        exit(_errmsg(errno, "can't delete cyclic alarm - "));
}

#include <sgstat.h>

/* read one characater with echo off */
void no_echo_read()
{
    int path = fileno(stdin);
    struct sgbuf opts, sopts;
    char c = '\0';

    /* Get path options */
    if (_gs_opt(path, &opts) == -1)
        exit(_errmsg(errno, "can't get path options - "));

    sopts = opts;                 /* Save options */
    opts.sg_echo = FALSE;         /* Disable echo */

    /* Set path options */
    if (_ss_opt(path, &opts) == -1)
        exit(_errmsg(errno,
                "can't set path options for echo off - "));

    read(path, &c, 1);

    /* Restore options */
    if (_ss_opt(path, &sopts) == -1)
        exit(_errmsg(errno, "can't restore path options - "));

    if (!c)
        exit(_errmsg(errno, "can't read one character - "));
}
```

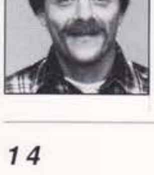
New Faces at Microware

Linda Andersen comes to Microware as an administrative assistant for the Technical Support "Hotline." Before coming to Microware, Linda worked for Shive-Hattery Engineers and Architects (West Des Moines, Iowa). She holds Bachelor of Arts degrees in German and office education from South Dakota State University (Brookings, South Dakota).

Efim Birger comes to Microware as a software engineer. Efim previously worked for MIEL Company (Moscow, Russia) as a lead engineer designing telecommunication and peripheral equipment. He holds a Bachelor of Science degree in electronics engineering from Moscow Telecommunications Institute.

Michael Braverman joins Microware as the multimedia coordinator for the CD-I Publishing Group. Prior to coming to Microware, Michael was an engineer with Iowa Public Television (Johnston, Iowa). He holds a Bachelor of Arts degree in communication studies from the University of Iowa (Iowa City, Iowa).

Ike Brown is Microware's quality program administrator. Before joining Microware, he was the director of the Polk County Computer Department (Des Moines, Iowa).



Dennis Gabler comes to Microware as a senior software engineer. Dennis came to Microware from Compaq Computer (Houston, Texas) where he was a systems engineer.

Gregory Goutman joins Microware as a software engineer. Previously, Gregory worked at Spets System Service (Odessa,Ukraine)as a software engineer. He holds a Masters of Science in physics from the State Institute of the Soviet Union.

Gwenna Jacobsen joins Microware as a senior software engineer. Prior to coming to Microware, Gwenna was a systems analyst at Iowa State University (Ames, Iowa). She holds Master and Bachelor of Science degrees in computer engineering from Iowa State University.

Dave Kimble joins Microware's New Media Systems Group as a software engineer. Dave comes to Microware from Interlight Productions (Tallahassee, Florida) where he developed compact disc-interactive tools and applications. He holds Bachelor of Science degrees in computer science and music composition from Florida State University (Tallahassee).

John Lengeling joins Microware as a senior software engineer. He comes to Microware from Tesseract Information Systems (Fayetteville, Arkansas) where he provided UNIX system administration, hardware maintenance and application development services. John holds a Bachelor of Science degree in computer information systems from Bradley University (Peoria, Illinois).

Anil Purohit joins Microware as a software engineer and is a candidate for a Master of Science degree in computer science from the University of South Carolina (Columbia, South Carolina). Anil specializes in the X Window System and parallel processing, the latter being the subject of his thesis. Anil holds a Bachelor of Science degree in electronic engineering from the University of Bombay (Bombay, India).

KD Roth comes to Microware as a product administrator. Previously, KD did free-lance business consulting. She holds a Bachelor of Science degree in home economics from the University of Wyoming (Laramie, Wyoming).

Amanda-Jane Saker joins Microware U.K. as a sales and office administrator. Before joining Microware, Amanda-Jane was a computer secretary at Testvale Surgery (Southampton, England). She holds a Bachelor of Technology degree in business studies.

Brent Thompson joins Microware as a communications software engineer. Brent comes to Microware from Iowa State University where he recently received his Bachelor of Science degree in computer engineering.

Pamela Weber comes to Microware as a technical support engineer. Previously, Pam was a computer analyst with the Department of Economics Rural Data Project at Iowa State University. She has a Bachelor of Science degree in computer science from Iowa State.

Ben Wheeler joins Microware U.K. as an internal sales specialist. Prior to Microware, Ben was a financial consultant with General Portfolio Insurance (Kingston-upon-Thames, England). He holds a Bachelor of Science degree in computer information systems. MSC

Training and microware Education

Course Offerings Expanded, Spring Schedule

MICROWARE'S TRAINING AND EDUCATION Department now offers a broader range of seminars covering the spectrum of real-time design. In addition to OS-9 seminars, classes are now available for OS-9000 and RAVE (Real-Time Audio/Video Environment).

The classes listed below are available for either OS-9 or OS-9000:

- The **Starter Seminar** is designed for managers who want to learn more about Microware's real-time system software. Little or no programming experience is needed. (1 day)
- The **Intermediate Seminar** is for users relatively new to OS-9 or OS-9000 who need to gain product knowledge quickly. (2 days)
- The **Advanced Seminar** offers experienced programmers extensive information about the function and operation of Microware's advanced I/O system. (2 days)
- The **Intermediate/Advanced Seminar** combines the two seminars above. (4 days)
- The **Driver Seminar** is specifically designed for programmers who will be writing device drivers for custom hardware. (2 days)
- The **Porting Seminar** is for those programmers who are planning to port OS-9 or OS-9000 to a new target. (1 day)
- The **RAVE Technical Seminar** explores the use of RAVE for creating real-time graphical interfaces. (2 days)
- The **Advanced Topics in CD-RTOS** (Compact Disc Real-Time Operating System) **Seminar** is for experienced Compact Disc-Interactive (CD-I) programmers who need firsthand knowledge from the developers of CD-RTOS. (2 days)
- The **Intermediate Topics in CD-RTOS Seminar** adds a day of OS-9 basics to the above seminar and is designed for new CD-I programmers. (3 days)

You may sign up for individual classes by contacting Microware or your authorized Microware representative. Intermediate and Advanced Seminars are scheduled on consecutive days so that attendees may sign up for either individual session or both sessions. The following courses will be offered through the winter and early spring of 1992:

FEBRUARY 25-28
OS-9 Intermediate/Advanced
*SOUTHERN CALIFORNIA**

FEBRUARY 27-28
Advanced CD-RTOS
OAKVILLE, ONTARIO, CANADA

MARCH 2
OS-9 Starter
SANTA CLARA, CALIFORNIA

MARCH 3-6
OS-9 Intermediate/Advanced
SANTA CLARA, CALIFORNIA

MARCH 23
OS-9 Starter
DES MOINES, IOWA

MARCH 24-27
OS-9 Intermediate/Advanced
DES MOINES, IOWA

MARCH 30-31
OS-9 Driver
DES MOINES, IOWA

APRIL 27
OS-9000 Starter
DES MOINES, IOWA

APRIL 28-May 1
OS-9000 Intermediate/Advanced
DES MOINES, IOWA

* No location had been set for this session at press time. Contact Microware or your authorized Microware representative for additional information. MSC

NEW PRODUCT

OS-9 WORM File Manager Now Available

MICROWARE RECENTLY COMPLETED THE **WORM (Write Once Read Many) File Manager (WOF)** running on OS-9. WOF gives users the functionality and capabilities of Microware's RBF (Random Block File Manager) on an optical disc platform. This gives users greater data storage capability—about 600M per disc—and permanence for important data.

Supports RBF System Calls

WOF uses the standard entry points for an OS-9 file manager, and supports most RBF system calls and C function calls. Some of its many features include the ability to store files in a hierarchical directory structure, disc partitioning, the ability to modify file attributes and the deletion of files. Although no data can actually be removed, file deletion is accomplished by marking a file's directory entry as deleted without physically deleting the file. This enables you to recover any data that might accidently be deleted.

WOF was designed for maximum disc use, while ensuring data retrieval after unexpected power losses by first writing a file's directory entry. The file manager was written in C for maintainability and portability.

The WORM File Manager is currently available for use with the Cherokee M600 optical SCSI drive. For more information about WOF, contact Microware or your authorized Microware representative. MSC

Synchronizing Data Module Access in OS-9

An Object-Oriented Approach

by Chuck Wesolowski
Native American Services, Inc.

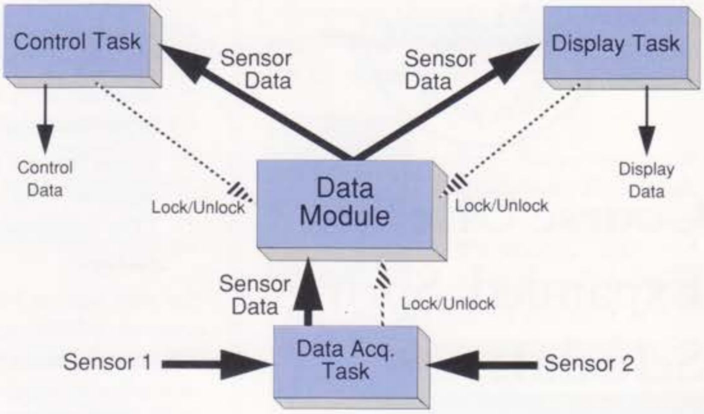
OS-9 DATA MODULES PROVIDE AN EFFICIENT MECHANISM FOR COMMUNIcating between processes. Synchronizing access to the data module can be accomplished using the OS-9 event system. Below is a description of a method for integrating of an event with a data module to provide processes with a mechanism to safely share data.

Interprocess transactions using data modules are extremely efficient because concerned tasks access data via a pointer reference. There is no system overhead involved. However, problems can occur if an updating task is interrupted by the scheduler before the entire update is complete. A consumer task may be scheduled in the interim and perform its function using incomplete or stale data. The lesson is that, while data modules allow efficient information transfer, they do not provide for synchronization of access to that information.

OS-9's event subsystem provides a mechanism for synchronizing the execution of tasks. An event may be defined which acts as a semaphore. Tasks sharing the event use the signal/wait mechanisms to allow orderly execution. A task holding the event or lock will block other tasks waiting on the event from executing until the holding task signals the event. This approach, when used with a data module, provides an effective method for synchronizing data transfer between tasks. A task need not worry about the scheduler interrupting it in the middle of an update because it is insured that consumer tasks will not be considered for execution while it holds the event. This is, in effect, a lock on the data module. When the update is complete, the holding task signals the event and the data module is made available to the next task in the event queue.

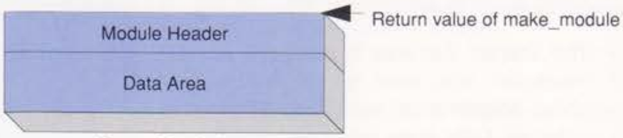
Because the event is so tightly coupled to the data module, it is convenient to take an object-oriented approach that defines the data module as an object and event operations as methods for accessing the object. This is accomplished by constructing a set of library routines (***dmod.l***, whose C source code is found at the end of this article) which transparently manage the event handling. When a task creates a data module, an event is automatically created and the event ID is embedded in the module itself, when a task links to a data module it automatically links to the associated event. When the data module disappears, so does the event. The only limitation is that module names are limited to twelve characters, the maximum length for an OS-9 event name.

Tasks use the ***dmod_lock*** and ***dmod_unlock*** functions to gain and relinquish exclusive access to the data module. The following data flow diagram illustrates three tasks sharing a data module.



In addition to managing the event handling, the ***dmod_ functions*** simplify data module usage by returning a pointer to the data area, not the module header. The user can therefore concentrate on the application instead of the OS-9 internals. ***Dmod_create***, ***dmod_link*** and ***dmod_unlink*** all operate using the name of the module. The lock and unlock functions use the pointer to the data area.

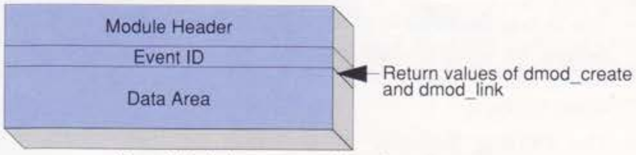
Internally, when OS-9 creates a data module, it constructs a module header which is linked into the module directory. The structure looks like this:



Data Module created by make_module

The value returned by ***make_module*** points to the module header which contains housekeeping information for OS-9. The user is concerned with the data area through which information is transferred, which is located just after the header for a data module. This is the value that ***dmod_create*** and ***dmod_link*** return.

When ***dmod_create*** makes a data module, it reserves room for an event ID in what OS-9 considers the first longword of the data area, but the value returned to the caller points to just after the event ID, which is the application's usable data area. If the caller requests a data module requiring 16 bytes, ***dmod_create*** asks OS-9 for 20 bytes. This is illustrated below.



Data Module created by dmod_create

Note that ***dmod_create*** makes the data module using system RAM and having public read/write permission. The low level ***_dmod_create*** function affords the programmer greater control over both permission and memory "color."

In conclusion, the synchronization of access to shared resources is an important concern in real-time systems design. The use of OS-9's event subsystem in conjunction with data modules assists in addressing this matter. The object-oriented ***dmod*** library provides a simple and efficient method for achieving synchronized data access.

```
/*
** Module                : dmod_
**
** Purpose               : Data Module library routines
**
** Date/Originator       : 10/30/91 --Chuck Wesolowski, NAS Inc.
**                                    Huntsville, AL
*/
#include      <module.h>          /* OS/9 module definitions*/
#include      <events.h>          /* OS/9 event definitions*/

#define       MTYPE     (mktypelang(MT_DATA,0))
#define       MATTR     (mkattrevs(MA_REENT,0))
#define       PERM      0x0333

/*
** Module                : _dmod_create
** Purpose               : Create data module (low level)
** Returns:
**            ok : Pointer to data module
**            err: 0 -- errno set to OS/9 error code
*/
void    *_dmod_create(name,size,perm,color)
char    *name;               /* Pointer to module/lock name*/
int     size;                /* Size in bytes*/
int     perm;                /* Permission */
int     color;               /* Colored memory*/
{
   char      *mp;            /* Data module pointer*/
   int       lock;           /* Lock event */
   int       msize;          /* Module size*/
/*
** Create lock
*/
   if ( (lock = _ev_creat(0,-1,1,name)) == -1)
        return(0);
/*
** Create data module
*/
   msize = size + sizeof(lock);
   if (((int) mp = (int) make_module(name,msize,MATTR,perm,-
        MTYPE,color)) == -1)
   {
        _ev_unlink(lock);
        _ev_delete(name);
        return(0);
   }

   mp += sizeof(struct modhcom);/* Point past module header*/
   *(int *)mp = lock;             /* Place lock in first field*/
   return(mp+sizeof(lock));       /* Return pointer to data area*/
}

/*
** Module                : dmod_create
** Purpose               : Create data module
** Returns:
**            ok : Pointer to data module
**            err: 0 -- errno set to OS/9 error code
*/
void    *dmod_create(name,size)
char    *name;               /* Pointer to module/lock name*/
int     size;                /* Size in bytes*/
{
/*
** Create data module
*/
   return(_dmod_create(name,size,PERM,0));
}

/*
** Module                : dmod_link
** Purpose               : Link to data module
** Returns:
**            ok : Pointer to data module
**            err: 0 -- errno set to OS/9 error code
*/
void    *dmod_link(name)
char    *name;               /* Pointer to module/lock name*/
{
   char      *mp;            /* Data module pointer*/
   int       lock;           /* Lock event id*/
/*
** Link to lock event
*/
   if ( (lock = _ev_link(name)) == -1)
        return(0);
/*
** Link to data module
*/
   if (((int) mp = (int) modlink(name,MTYPE)) == -1)
   {
        _ev_unlink(lock);
        return(0);
   }
```

Continued

```
/*
** Return pointer to data area
*/
   return(mp += (sizeof(struct modhcom)+sizeof(lock)));
}

/*
** Module                : dmod_lock
** Purpose               : Lock data module
** Returns:
**            ok : Module pointer
**            err: 0 -- errno set to OS/9 error code
*/
void    *dmod_lock(mod)
int     *mod;                /* Pointer to module*/
{
   if (_ev_wait(*(mod-1),0,0) == -1)
        return(0);
   return(mod);
}

/*
** Module                : dmod_unlink
** Purpose               : Unlink from data module
** Returns:
**            ok : Pointer to data module name
**            err: 0 -- errno set to OS/9 error code
*/
char    *dmod_unlink(name)
char    *name;                      /* Pointer to module/lock name*/
{
   int       *mp;                   /* Data module pointer*/
/*
** Link to data module
*/
   if (((int) mp = (int) modlink(name,MTYPE)) == -1)
        return(0);

   (char *) mp += sizeof(struct modhcom);
   _ev_unlink(*mp);                 /* Unlink from lock event*/
   _ev_delete(name);                /* Attempt delete of lock event*/
/*
** Unlink from module
*/
   munload(name,MTYPE);
   munload(name,MTYPE);
   return(name);
}

/*
** Module                : dmod_unlock
** Purpose               : Unlock data module
** Returns:
**            ok : Module pointer
**            err: 0 -- errno set to OS/9 error code
*/
void    *dmod_unlock(mod)
int     *mod;                /* Pointer to module*/
{
   if (_ev_signal(*(mod-1),0) == -1)
        return(0);
   return(mod);
}
```

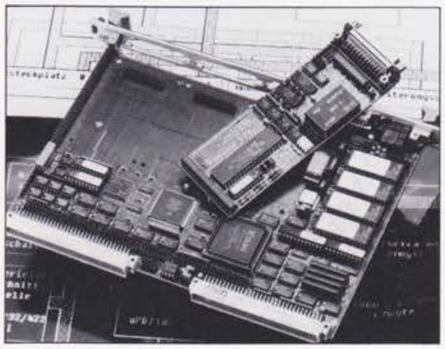
Chuck Wesolowski is a senior software engineer with Native American Services, Inc. (NAS) in Huntsville, AL. He holds a Bachelor of Arts degree from Emory University (Atlanta, GA). Chuck recently completed work on the Advanced Helicopter Control System for the U.S. Army and a rewrite of the OS-9 Kernel for FORCE Computer's ISIO/2 intelligent serial I/O controller. NAS specializes in real-time embedded systems and engineering research. NAS was named NASA Minority Contractor of the Year in 1990. MSC

New Vendor Products

MEN Mikro Elektronik
Themis Computer
Synergy Microsystems
Compcontrol International
EKF Elektronik
MATRIX Corporation
Syntel Microsystems
RTware
Datentechnik Reischke

68332 Board from MEN

MEN Mikro Elektronik GmbH (Nuremberg, Germany) recently introduced several new products including an MC68332-based board. The **A4** is a 6U multi-function board with a 16 or 20 MHz 68332 microcontroller, four-channel DMA controller, up to 2M DRAM and up to 2M EPROM/EEPROM. The 68332 includes 2K SRAM, two serial interfaces, programmable watchdog, periodical interrupt timer and programmable chip-select logic. The A4 has three expansion slots that allow the board to be configured for use as an intelligent peripheral controller, a single-master CPU or a stand-alone CPU.



The A4 68332-based board from MEN.

For more information, contact MEN Mikro Elektronik GmbH, Wiesentalstrasse 40, D-8500 Nuremberg 90, Germany. Phone: (49) 911/332755. Fax: (49) 911/396686.

Two Development Systems from Themis

Themis Computer (Les Ulis Cedex, France) recently announced two OS-9 VME development systems. The **TSVME 905-40** is built around a 20 MHz 68030 and includes a 68882 FPCP, 4M dual-ported RAM, battery-backed clock, two serial ports, Centronics port, SCSI interface, Ethernet interface, 40M hard disk drive and 1.44M flexible disk drive. The system is provided with OS-9, and TCP/IP, ***telnet*** and ***ftp*** facilities.

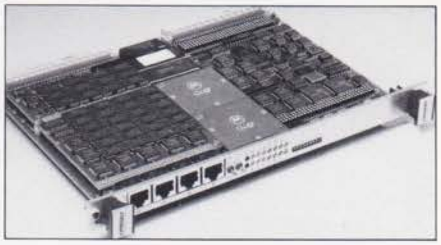
The **TSVME 901-40** features a 10 MHz 68030, 1M DRAM, 64K battery-backed SRAM, battery-backed clock, two serial ports, one parallel port, SCSI interface, 40M hard disk drive and 1.44M flexible disk drive. OS-9 is provided with the system as well.

TSVME 910-40 Development System.

For more information, contact Themis Computer, 29 Avenue de la Baltique, 91953 Les Ulis Cedex, France. Phone: (33) 1.69.82.20.00. Fax: (33) 1.64.46.45.50.

Dual 68040 SBC from Synergy

Synergy Microsystems Inc. (Encinitas, California) has released two 68040-based single board computers. The **SV420** features two 25 MHz 68040 CPUs, up to 32M of shared high-speed DRAM, four RS-232/422 ports, EPROM/Flash EPROM support, EZ-bus expansion interface, five 16-bit timers, real-time clock, 8-bit ID port and an optional DMA controller supporting SCSI, Ethernet and four additional serial ports. The dual 68040 CPUs can be configured for tightly- or loosely-coupled operation.



Synergy's dual 68040 SBC.

The **SV430** features a 25 or 33 MHz 68040 CPU, 68EC030 I/O coprocessor, up to 32M DRAM, SCSI controller with DMA, Ethernet controller with DMA, four RS-232/422 serial ports, four serial channels with DMA, EPROM/Flash EPROM support, EZ-bus expansion interface, five 16-bit timers, real-time clock and 8-bit ID port.

For more information, contact Tom Powell, Synergy Microsystems Inc., 179 Calle Magdalena, Encinitas, California 92024. Phone: (619) 753-2191. Fax: (619) 753-0903.

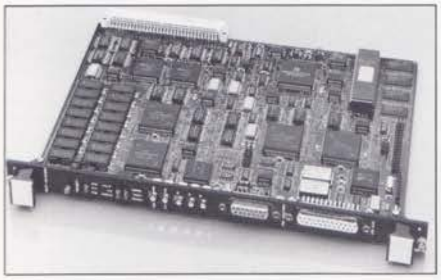
PC to OS-9 Networking from Compcontrol

ARC/TCP from Compcontrol International B.V. (Eindhoven, The Netherlands) allows MS-DOS and OS-9 systems to be linked by ARCNET with support for TCP/IP ***ftp*** and ***telnet*** facilities. OS-9 systems use the OS-9 Internet Support Package (OS-9/ISP) with ARCNET link layer drivers for communication across ARCNET. The use of ARCNET helps ensure a real-time deterministic connection. **RING/TCP** provides the same functionality across a Token Ring network. ARC/TCP and RING/TCP are available for MS-DOS and OS/2 systems.

For more information, contact Compcontrol International B.V., Science Park Eindhoven 5110, P.O. Box 921, 5600 AX Eindhoven, The Netherlands. Phone: (31) 40-414025. Fax: (31) 40-414035.

New Products from EKF

EKF Elektronik Messtechnik GmbH (Hamm, Germany) recently introduced new hardware and software for OS-9 users. The **EKF VME 68400-AGDC** is a 6U VME color graphics module based on the Hitachi ACRTC 63484 graphics controller, and includes one GMIC 63485 memory controller, two GVAC 63486 video attribute controllers, 2M video RAM, 16 MHz 68HC000 coprocessor and two serial ports. The board features programmable pixel rates up to 1280 × 1024 at 8-bits/pixel with 256 colors from a palette of 262,144. Direct lightpen input is available. Outputs include TTL RGB, analog RGB, composite sync, V-Sync and H-Sync.



EKF's color graphics module.

The **EKF VME 68390-TC** 6U timer/counter is built around two Am9513A system timing controller chips and offers 10 individually-programmable 16-bit up/down counters. Each counter has its own clock input and output, plus two additional gates.

XTREE provides a graphical method for users to move through their file systems. Using cursor control keys, a user can change directories and display the current contents. Standard functions can be executed with a single key. For example, a file is highlighted with the cursor and the letter "d" is typed to ***dump*** the file's contents.

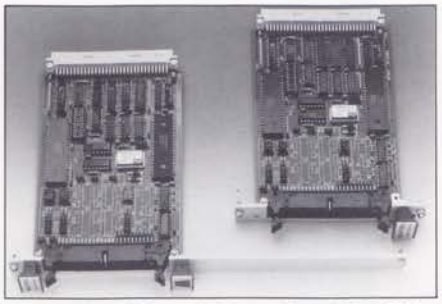
For more information, contact B. Kleeburg, EKF Elektronik Messtechnik GmbH, Philipp-Reis-Strasse 4, D-4700 Hamm 1, Germany. Phone: (49) 23 81/68 90-0. Fax: (49) 23 81/68 90 90.

Mass Storage, Serial Board from MATRIX

MATRIX Corporation (Raleigh, North Carolina) recently announced the release of their Data Storage Module and MSX-SIO4A serial communications board. The **Data Storage Module (DSM)** combines one or two disk drives and controller in a 6U, two-VME slot design. The DSM can be configured with any combination of disk drives, including 50M or 100M SCSI hard disk drive and 1.44M flexible disk drive. A VMEbus P2 connector extends the SCSI interface for up to five more devices.

The MSX-SIO4A is a 3U VME board with an 85C30 controller. The board provides four independent, full-duplex serial ports that can be configured to handle asynchronous, synchronous, byte-synchronous or bit-oriented protocols. The board operates in extended temperature ranges from -40°C to +85°C.

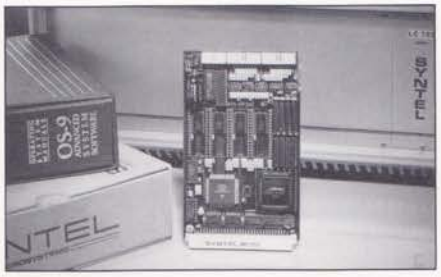
For more information, contact MATRIX Corporation, 1203 New Hope Road, Raleigh, North Carolina 27610. Phone: (919) 231-8000. Fax: (919) 231-8001.



The MSX-SIO4 serial board from MATRIX.

Syntel Graphics Board, Development System

Two new products were recently released by Syntel Microsystems (Huddersfield, England). The **VFG2000** (VME Fast Graphics) is based on the AMD 95C60 QPDM graphics processor chip with dual-ported video RAM. The board is capable of rectangle fill rates of 50 million pixels per second and block fills of 16 million pixels per second. The VFG2000 offers display memory resolution of 2048 × 2048 at 4-bits/pixel and screen resolution up to 1280 × 960 pixels.



Syntel's new OS-9 development system.

The LC702 development system is built around a 16 MHz MP302 CPU card (based on Motorola's 68302), up to 4M DRAM, 384K battery-backed SRAM, 40M or 100M hard disk drive and a 1M flexible disk drive. Additional features can be added through the 12-slot G-64 bus backplane.

For more information, contact Paul Wilson, Syntel Microsystems, Queens Mill Road, Huddersfield HD1 3PG, England. Phone: (44) 484 535101. Fax: (44) 484 519363.

ControlCalc Now Available for OS-9000

ControlCalc is a real-time spreadsheet system for control, data acquisition and test applications (see *PIPELINES*, Spring 1991, pp. 8-9). ControlCalc was originally developed to interface with OS-9/RAVE as a method of controlling and monitoring multiple processes. RTware, Inc. (Durham, North Carolina) recently announced the availability of ControlCalc for the OS-9000 Real-Time Operating System. In addition to RAVE, ControlCalc also provides links to the X Window System.

ControlCalc is now available for OS-9000.

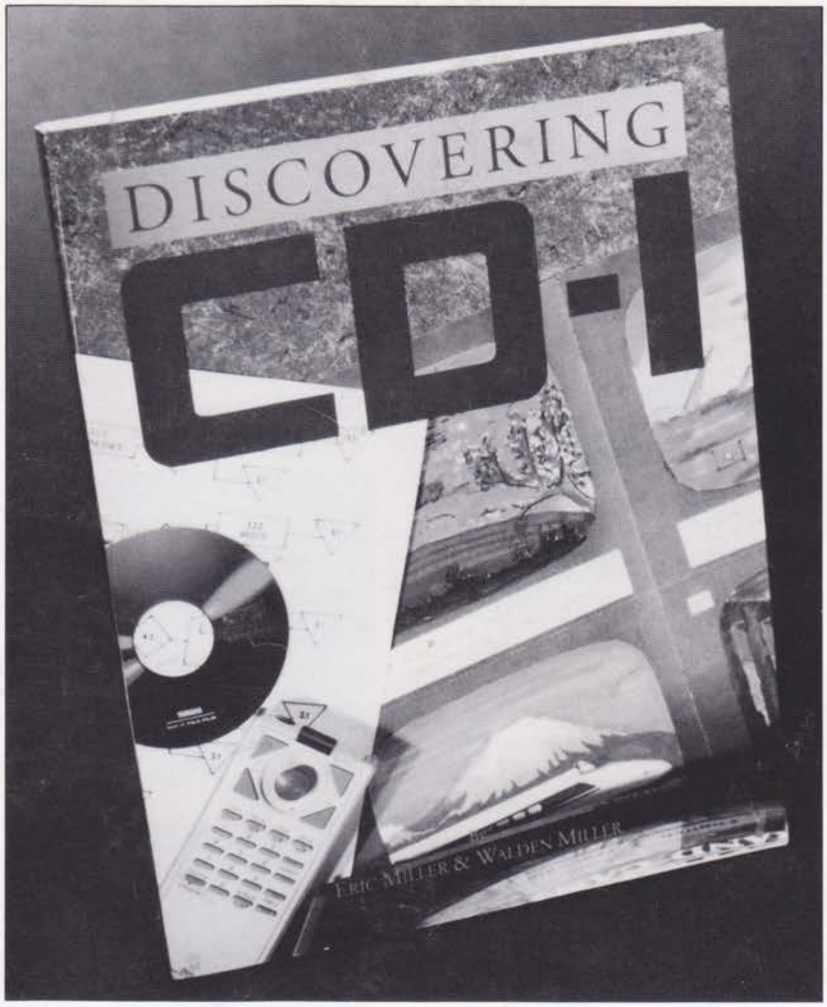
For more information, contact Laurent Meilleur, RTware, Inc., 714 9th Street, Suite 206, Durham, North Carolina 27705. Phone: (919) 286-3114.

Read and Write OS-9 Diskettes on PCs

Datentechnik Reischke (Kiel, Germany) recently introduced a software package that allows users to read and write OS-9 diskettes on PC and compatible systems. Standard directory commands (such as ***chd***, ***del***, ***copy***, ***makdir/deldir***, ***attr***) are fully supported. The software transparently detects most standard OS-9 formats, as well as allows for detection of special formats.

For more information, contact O. Reischke, Datentechnik Reischke, Bremerstrasse 2, DW-2300 Kiel 1, Germany. Phone: (49) 431 805293. Fax: (49) 431 82000. MSC

Definitive Guide to CD-I



Discovering CD-I, by Eric Miller and Walden Miller, is the definitive resource on the compact disc-interactive (CD-I) multimedia system. The book is a comprehensive guide to this exciting, new technology and is written for technical as well as non-technical readers. Whether involved in management, creative or technical development, readers will gain a clear understanding of the capabilities of CD-I for business and consumer applications. Subjects include:

- Getting Into CD-I—Take a look at the possibilities for CD-I applications in the business and consumer markets.
- A Technical Perspective—Examine the CD-I Player, current authoring system software, title design and production considerations.
- Making The Decision—What are the economic and development concerns for firms evaluating CD-I? The answers are here.

The Millers have extensive experience in CD-I. Eric is the manager of Microware's New Media Systems and has been working on various aspects of CD-RTOS (Compact Disc Real-Time Operating System) and the CD-I Green Book standard since 1986. Walden Miller worked as manager of Microware's Technical Documentation department and helped write much of the documentation for CD-RTOS.

Discovering CD-I can be ordered for $45 plus shipping and handling by calling 1-800-475-9000 (U.S. credit card customers) or (515) 224-1929. Contact Microware or your authorized Microware representative for quantity discounts. MSC

microware®
MICROWARE SYSTEMS CORPORATION
1900 N.W. 114th Street
Des Moines, Iowa 50325-7077