PIPELINES

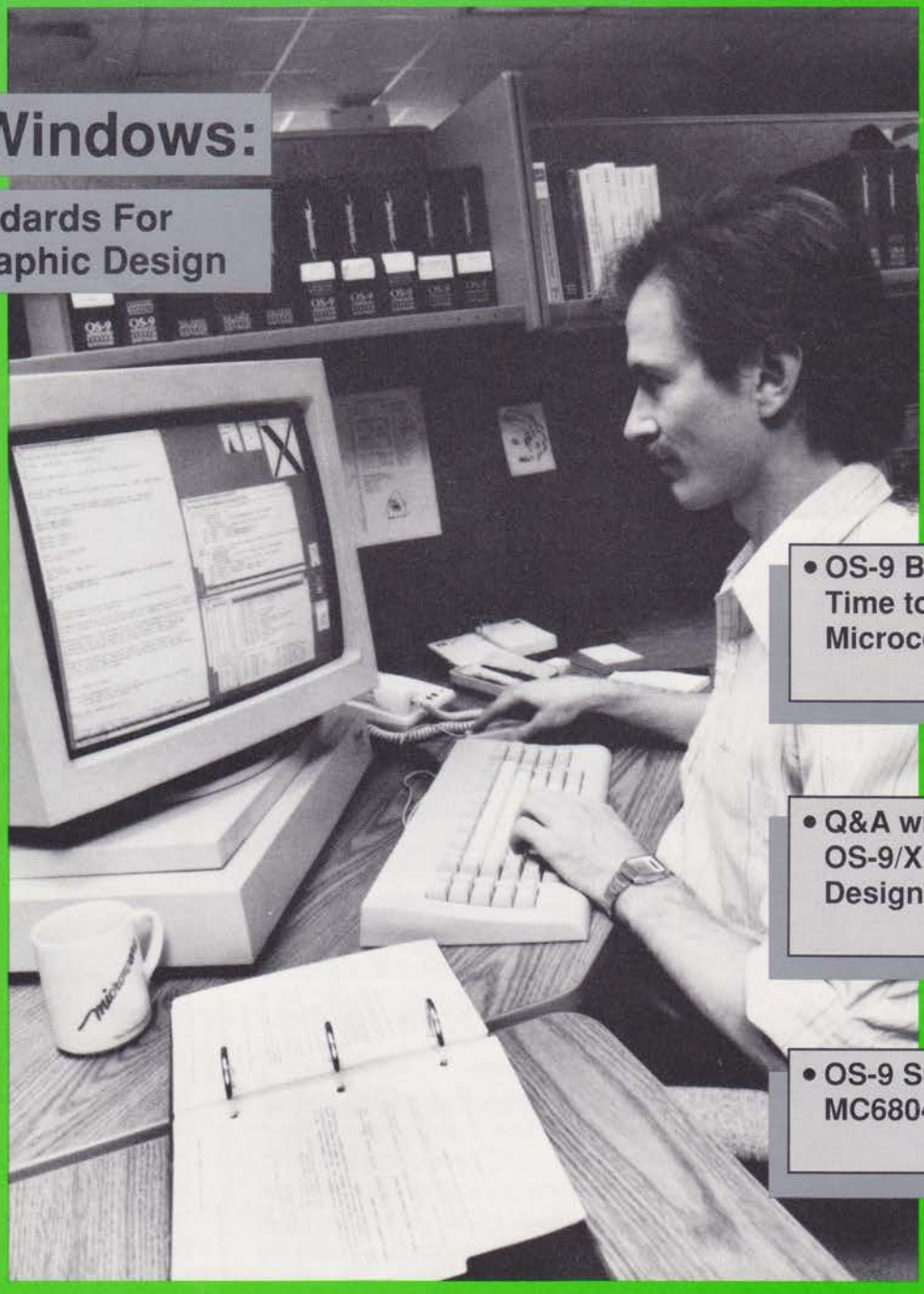
Volume 6 Number 2

Covering Microware's Real-Time System Solutions

Spring 1991

OS-9/X Windows:

Industry Standards For Real-Time Graphic Design



microware

PIPELINES
Spring 1991
Volume 6 Number 2

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

Editor:
Steve Simpson

Art Director:
Polly Steele

Microware Contributors:

Warren Brown	Ric Yeates
Mike Burgher	Todd Earles
Drew Crane	Steve Johnson
Dave West	Mike Ahrens
Bret Farnum	Steve Guntly
Lee Glenn	Ellen Grant
Tom West	Dave Davis

Photography:
David F. Davis

Do You Have New OS-9 or OS-9000 Products?

IF YOU HAVE NEW HARDWARE OR SOFTware products that run under OS-9 or OS-9000, please submit a press release and black & white photograph of the product for consideration for publication in *PIPELINES*. All materials should be sent to the Editor of *PIPELINES* at the address above. For more information, call Steve Simpson at (515) 224-1929.

On The Cover

OS-9/X WINDOWS ALLOWS DEVELOPers to design real-time graphic interfaces across a variety of computer platforms. On the cover, a Microware engineer uses a Sun system to develop and display OS-9/X Client applications running on a remote OS-9 system.

Delivering On Our Promise

Ken Kaplan,
Microware's President

IN THE LAST ISSUE OF *PIPELINES*, I hinted at some of the exciting new products Microware is working on. Well, some of those products are starting to hit the streets. As you look through the pages of this issue, you'll find a number of new and exciting product offerings from Microware.

One area where Microware excels is products for seamless distributed development and implementation. To this end, we are excited about the release of **OS-9/X Windows**. And, we're proud to be the first real-time operating system to support this powerful graphics standard. We have dedicated quite a few pages to X Windows because we feel very strong about the importance of this product to real-time developers.

OS-9/X Windows:
Industry Standards for Real-Time Graphic Development

MICROWARE IS PROUD TO ANNOUNCE the availability of **OS-9/X Windows**. OS-9/X Windows is a robust implementation of MIT's X Windows Version 11 Release 4 for the OS-9 Real-Time Operating System. OS-9/X Windows gives designers the tools necessary for development and implementation of powerful graphics-based applications under OS-9 and other computer platforms across a local area network (LAN).

OS-9/X Windows Overview

Most windowing packages rely heavily on operating system-specific calls to manage a display. The main benefit of X Windows is that it provides easy software development, deployment and supervision of applications in distributed computing environments. This means a single monitor can display applications from a variety of remote systems, or a single remote system can display applications on various monitors, completely independent of the software or hardware platforms being used. This separation of functionality makes it possible for several X Client applications to use the same X Server display where various systems monitor processes and report to a single location such as in a factory floor environment.

X Windows is typically run on workstations that have large bit-mapped graphics displays and some number of input devices (a keyboard, mouse or graphics tablet). X supports both color and monochrome displays, and can support multiple physical displays connected to a single workstation. Systems that support X Windows range from low-resolution personal computers to 256-color high-resolution CAD systems. In addition to general purpose computers, a new type of display hardware, known as an X Terminal, can also act as a display for X Windows. As long as appropriate graphics

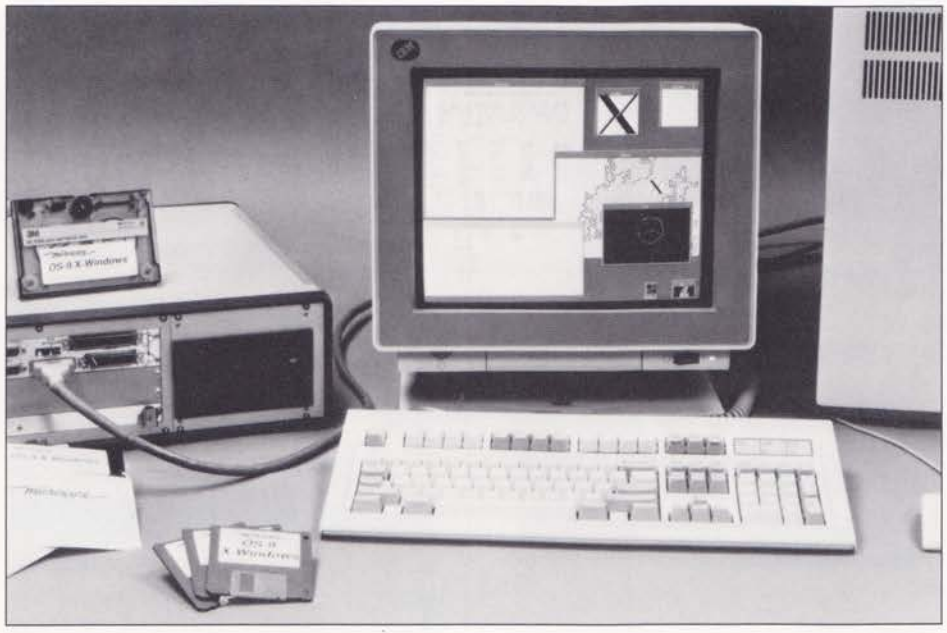
OS-9/X Windows is a perfect fit for Microware's goal of providing distributed development tools, as well as integrated packages for real-time control and supervision. X Windows gives developers a platform-independent package for creating real-time graphical interfaces that can display across local area networks (LANs).

And, there's more! In recent months, Microware has introduced support for Motorola's 68332 and 68340 microcontrollers, as well as their MC68040 microprocessor. The 683XX microcontrollers are fast becoming important platforms for embedded system designers because of their integrated I/O capabilities. The 68040 represents Motorola's latest processor in the CPU32 family. The '040 packs a punch on a single CPU and offers real-time designers many opportunities for development and embedded applications.

Our line of cross development tools is expanding, as well. First, there's the release of **UniBridge** for OS-9000, giving OS-9000 developers an option for distributed development. OS-9/UniBridge support for Silicon Graphics' line of UNIX workstations opens more doors for OS-9 users. And, the **PCBridge Enhanced Pak** builds on our popular PC-DOS cross development platform to enhance DOS-hosted OS-9 development.

We're equally excited about the products that we'll be releasing in coming months. I don't want to tip our hand at this point. But, I assure you that upcoming product announcements will be as exciting as the ones we have detailed in this issue.

—Ken Kaplan



OS-9/X Windows is a robust implementation of MIT's X Windows.

hardware exists, X Windows can be ported to almost any computer platform.

OS-9/X Windows System Architecture

The X Windows System places the display functionality in the control of a user application called an X Server. X applications, or "X Clients," interact with an X Server to control the applications' displays. X Windows simplifies the development of graphics-based applications because applications need only make calls to the X Server. There is no code required within the application to control the display itself. Instead, basic graphics messages are written in the application that instruct the X Server what to display. Therefore, since the X Server

OS-9/X WINDOWS
Please turn to Page Fourteen

OS-9/683XX Evaluation Pak Now Available

MICROWARE RECENTLY INTRODUCED **Industrial OS-9/683XX** for Motorola's 68332 and 68340 microcontrollers. Microware also offers an evaluation package that allows designers to test and download OS-9-based 683XX code to their 683XX Evaluation System (EVS). The **OS-9/683XX Evaluation Pak** includes:

- Industrial OS-9/683XX EPROMs
- Special evaluation copy of PCBridge cross development package†
- OS-9 utilities and ROM image files
- Sample OS-9 C programs
- Evaluation Pak User's Guide

The Evaluation Pak allows you to develop and compile sample real-time C code using the special evaluation copy of PCBridge or a Professional OS-9 system. Then, download your code to your 683XX EVS via a serial link, debug it and execute your application.

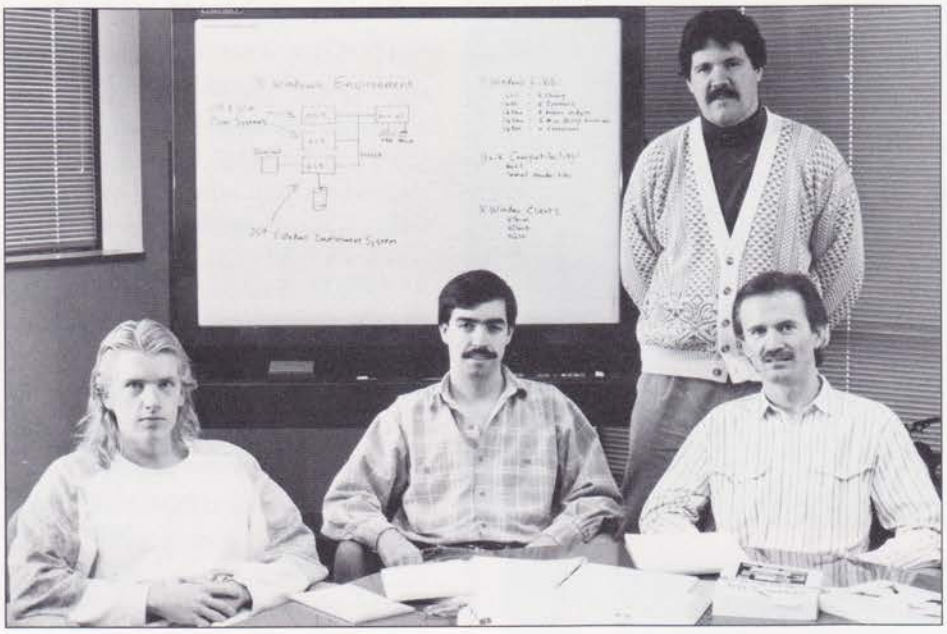
If you are using a PC and PCBridge, you will need DOS Version 3.3 or later, 512K RAM and one serial port. For OS-9 systems, you'll need Professional OS-9 Version 2.3 or later, 512K RAM and two serial ports.

For more information on the Evaluation Pak or Industrial OS-9/683XX, contact Microware or your authorized Microware representative. MSC

† This special evaluation copy of PCBridge allows users to write and compile a limited number of lines of C code for evaluation purposes.

OS-9/683XX Evaluation Pak.

Q&A with the OS-9/X Windows Design Team



The OS-9/X Windows Design Team is (l. to r.) Erik Johnson, Todd Earles, Kim Kempf (standing) and Lee Glenn.

Q *WHAT IS X WINDOWS?*

A X Windows is a world-wide graphics standard. It was originally developed at MIT for the UNIX operating system, but today it is used on a wide variety of operating systems and hardware. ***OS-9 is the first real-time operating system to support the X Windows environment.***

Q *Can I use X Windows to monitor my OS-9 real-time applications?*

A Yes, as a matter of fact, X Windows is probably the best windowing interface to use for real-time applications. This is because, in X Windows, the software for displaying the graphics and handling the user interaction is separate from the actual application.

This means that in a typical real-time environment, you would have your OS-9 system running the time-critical software. This OS-9 system could then be connected to a workstation running the X Server. This way, the OS-9 system does not have to burden itself with the job of drawing graphics and dealing with user input. This is handled by the CPU in the workstation.

Q *Can OS-9/X Clients use DOS or UNIX workstations as the X Server?*

A Absolutely, this is a very attractive solution. Because of the design of X Windows, it is not required that the client (application side) and the server (display side) run under the same operating system, or work on the same kind of hardware.

A very common solution will be to use a VME system running OS-9 tied to a workstation or PC running a UNIX or DOS X Server.

Q *Can multiple X Client applications share the same X Server for display?*

A Yes, it is possible with X Windows to have several applications, each running on a different machine, all display on one X Server. Therefore, one person can monitor and control several processes on several machines at the same time.

Q *Can I port existing UNIX X applications to OS-9?*

A Yes, most X applications are written in C, which is a very portable language. The X Windows code itself is 100% compatible between operating systems. You only need to deal with the non-X Windows part of the code, specifically where differences between operating systems exist.

To make the job even easier, Microware provides a UNIX compatibility library with OS-9/X Windows. This is a library of some of the most used UNIX functions.

Q *Can OS-9/X Windows be ROMed?*

A Like all other OS-9 software, the answer is YES.

Q *What special hardware do I need to run an X Client on my OS-9 system?*

A None, all you need to run an X application is hardware capable of running the OS-9 Real-Time Operating System and TCP/IP. This system can then be connected to any X Server.

Q *How do I debug an X application?*

A Just like any other application. OS-9 supports resident development, or cross development from UNIX or PC-DOS systems. In all these environments, you can debug your C program using the source level debuggers provided for OS-9.

Q *What future X Windows products will Microware support?*

A The future for X Windows looks very bright. Microware is already shipping X Windows client support. This allows users to develop X Windows applications to run on any OS-9 system.

Next, Microware will offer server support. Microware will also offer support for Motif under OS-9, as well as support for programs that allow you to interactively create X Windows applications, instead of always having to write the C code yourself. MSC

OS-9 Now Available for MC68040 & MVME 165

MICROWARE RECENTLY INTRODUCED support under OS-9 for Motorola's **MC68040** and **MVME 165**. The 68040 is Motorola's third-generation of 680X0-compatible, 32-bit microprocessors.

OS-9 now supports the MVME 165.

The 68040 microprocessor features an MC68030-compatible integer unit, an MC68881/882-compatible floating-point unit, two independent memory management units (MMUs), and independent 4K instruction and data caches. The chip employs bus-snooping to increase cache coherency and maximize the cache hit ratio. The floating-point unit includes a subset of the MC68882 instruction set, and includes additional instruction formats for single- and double-precision rounding. Any floating-point instructions not supported in hardware are emulated in software.

The MVME 165 is the first single board computer in the 68040 line from Motorola. The MVME 165 includes the 68040 running at 25 MHz, either 4M or 16M RAM, two serial ports, a battery-backed real-time clock and a VSB bus interface.

OS-9/68040 Products

OS-9/68040 supports the power available with the 68040, the instruction and data caches, on-chip integer and floating-point units, and on-chip memory management. The initial release of OS-9 for the 68040 includes the standard OS-9 software math package. This gives users the option of using either the '040 on-chip hardware math or OS-9's software math package. Full floating-point software math emulation will be included in a future release of OS-9/68040.

Microware packages for the 68040 currently include the **Professional OS-9/68040 PortPak**, **Industrial OS-9/68040 PortPak** and **MVME 165 DevPak**. The Professional and Industrial PortPaks are designed for customers who want to install OS-9 on their new 68040 hardware.

The MVME 165 OS-9 DevPak is preconfigured to take advantage of the on-board features. When ordering the MVME 165 DevPak, users can choose from the following boot media options:

- MVME 327 flexible disk
- TEAC SCSI flexible disk
- TEAC cassette
- Archive Viper QIC tape

OS-9/68040 Now Available

The Professional and Industrial OS-9/68040 PortPaks and MVME 165 DevPak are now available. To order, call Microware or your authorized Microware representative. MSC

Silicon Graphics Now Supported By OS-9 UniBridge

SILICON GRAPHICS INCORPORATED (Mountain View, CA) designs and manufactures a full line of visual processing systems and servers. Silicon Graphics' products are targeted for use in mechanical engineering, visual simulation, computational fluid dynamics, animation and color publishing.

Microware announces OS-9 UniBridge for Silicon Graphics systems. UniBridge is an integrated software package designed to support distributed OS-9 C programming, remote debugging and supervision of real-time tasks. Using UniBridge, designers can compile OS-9 C or Assembly language applications on a Silicon Graphics host, then download the executable module to the OS-9 target system via Ethernet.

Supports 4D Workstations

This release of OS-9 UniBridge includes support for Silicon Graphics' 4D family of UNIX-based workstations, as well as Silicon Graphics' IRIX™ operating system. The 4D family is built around multiple MIPS microprocessors. IRIX is Silicon Graphics' parallel processing UNIX operating system that is based on AT&T System V3 with Berkeley extensions.

This introduction provides benefits for both OS-9 and 4D users:

- Silicon Graphics users can now take advantage of the power of OS-9/680X0 when developing embedded or distributed real-time applications.
- OS-9 UniBridge provides a fully integrated development environment which reduces development time, complexity and cost.
- OS-9 users can now utilize the advanced visual processing power of the Silicon Graphics family when developing distributed applications.

To order, or for more information, call Microware or your authorized Microware representative. MSC

Silicon Graphics' workstation.

PCBridge V1.3 and PCBridge Enhanced Pak

PCBRIDGE PROVIDES A PC-BASED platform for OS-9 cross development. Microware recently introduced the **PCBridge Enhanced Pak** to extend PCBridge's capabilities.

PCBridge Version 1.3

PCBridge has been updated to Version 1.3 primarily to support the release of 68332 and 68340 products. The PC Cross C Compiler has been upgraded to Version 3.2.1 and will correctly compile code for all 680X0 and 683XX targets. PCBridge Version 1.3 also includes maintenance to previous versions.

Announcing PCBridge Enhanced Pak

The PCBridge Enhanced Pak provides the cross development power of PCBridge, as well as additional development tools. The following tools are included in the Enhanced Pak:

- Random Block File Manager (RBF) with RAM disk driver, descriptors and utilities for the creation and use of RAM disks
- OS-9 C Source Level Debugger for interactive execution of C code
- OS-9 System State Debugger aids in testing of device drivers and descriptors
- POLYTRON's PolyMake utility, an advanced MAKE utility for the PC
- PCBridge Menu System Enhancements to reflect the new tools

Special Upgrade Offer

Microware now has a special limited-time offer for PCBridge users who wish to upgrade to the PCBridge Enhanced Pak.

Through July 1, 1991, current PCBridge users may purchase the upgrade to the Enhanced Pak at special prices from Microware and authorized Microware distributors. This offer is available only to PCBridge users who purchased PCBridge before May 1, 1991.

Contact Microware or your authorized Microware representative for more information. MSC

PCBridge provides a seamless DOS-hosted OS-9 cross development environment for a variety of PC-DOS systems.

OS-9000 UniBridge Now Available

MICROWARE ANNOUNCES THE RELEASE of OS-9000 UniBridge. OS-9000 UniBridge provides a complete UNIX cross development option for OS-9000 developers.

UniBridge Overview

UniBridge is a software package for C and Assembly language development and communication connecting UNIX to either OS-9000 or OS-9. The package includes UNIX and OS-9000 or OS-9 software modules for distributed real-time programming, remote debugging and UNIX supervision of real-time processes.

UniBridge provides a gateway that allows UNIX to exist in every phase of system integration. From prototype testing to target installation, each module of UniBridge can be run on either ROM- or disk-based target systems.

System Requirements

OS-9000 UniBridge operates on the following UNIX workstations:

- Sun 3 or Sun 4 running SUN/OS
- HP 9000 Series 300 running HP/UX

OS-9000 system requirements are:

- OS-9000 Version 1.2 or greater on 68020/30/40 or 80386 processor
- Ethernet controller

Contact Microware or your authorized Microware representative to order OS-9000 UniBridge. MSC

OS-9000 UniBridge is now available.

New Employees at Microware

Front row (left to right): Kim Dunn, Susan Muhrer and Usha Larson.
Middle row: Michael Soolkin, Del Wadle and Brian Lewis.
Back row: Curt Schwaderer and Tim Wilson.

Kim Dunn is the marketing department's account administrator for U.S. sales. Kim comes to Microware from Younkers Department Stores (Des Moines, Iowa) where she was a corporate trainer. She has a Bachelor of Arts degree in management from Simpson College (Indianola, Iowa). Kim enjoys restoring antique furniture in her spare time.

Usha Larson is a software engineer in Microware's product integration department. Previously, Usha was a senior programmer with Retail Management Systems (Des Moines, Iowa). Usha has a Bachelor of Science degree in computer engineering from Iowa State University (Ames, Iowa). When not working, she enjoys gourmet cooking, alpine skiing, water skiing and listening to music.

Brian Lewis joins Microware as a software duplication technician. Prior to coming to Microware, Brian was working for a Kansas City, Missouri construction firm while completing college. He holds a Bachelor of Arts degree in management and human resources from Mid American Nazarene College (Olathe, Kansas). Brian enjoys coaching little league baseball and basketball, running, weight training and playing basketball.

Susan Muhrer is the voice of Microware's "Hotline," working as the technical support "Hotline" receptionist. Previously, Susan was a computer instructor and personal computer consultant for First Interstate Bank (Los Angeles, California). In her spare time, she enjoys nordic skiing, acting in community theater and drawing.

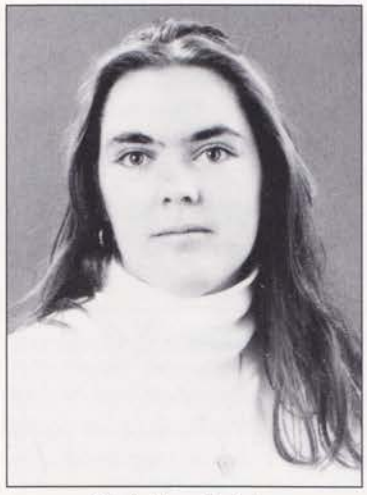
Curt Schwaderer joins Microware as an ISDN development engineer. Curt comes to Microware from Electrospace Systems (Richardson, Texas) where he designed and installed telecommunications switching systems. He holds a Bachelor of Science degree in computer engineering from Iowa State University. Curt enjoys playing basketball and lifting weights.

Michael Soolkin comes to Microware as a systems development engineer. Prior to joining Microware, Michael was a senior project engineer with MNTPKC (Moscow, USSR) where he developed controller devices for 8080 and 8088 microprocessors. He also worked as an engineer with the Central R&D Bureau of the Ministry of Telecommunications (Moscow) and holds a Bachelor of Science degree in electronic engineering from Moscow Telecommunication University. In his spare time, Michael enjoys tennis and reading.

Del Wadle is our accounting department's financial information systems manager. Del comes to Microware from W.G. Jaques Company (Des Moines, Iowa) where he was their corporate systems manager/product manager. He holds a Bachelor of Science in economics from Creighton University (Omaha, Nebraska). In his spare time, Del enjoys studying business, boating, camping, hunting and car care.

Tim Wilson joins Microware's "Hotline" as a technical support engineer. Previously, Tim was an OS-9 systems programmer/operator with CDC/Quester (Des Moines, Iowa). He holds a Bachelor of Arts degree in computer science from Westmar College (Le Mars, Iowa). Tim enjoys disk jockeying in his spare time.

Christine Folcher joins Microware's French office as secretary. Christine has her Secretarial Baccalaureate. She enjoys walking and traveling. MSC



Christine Folcher

Real-Time Spreadsheet Integrates RAVE

Richard Clarke
RTware, Inc.

CONTROLCALC IS A NEW PARADIGM for industrial and real-time control based on the most widely understood software method, the spreadsheet. ControlCalc, from RTware, Inc. (Durham, North Carolina) can be used to produce a wide range of systems, ranging from embedded, diskless controllers to large, graphics-based applications using Microware's Real-Time Audio/Visual Environment (RAVE). All the system components and user-created templates are OS-9 and OS-9000 data modules which can be ROMed and auto-booted using standard OS-9 and OS-9000 methods.

ControlCalc is a fourth generation language for control systems. It moves beyond traditional sequential programming languages to a fill-in-the-blank, object oriented approach which is easily mastered.

ControlCalc has a built-in compiler that is hidden from the user and acts to supercharge the recalculation. ControlCalc also provides direct digital and analog I/O functions and shared memory communication to external programs.

Analog Feedback Control

Compiled performance and real-world I/O allow ControlCalc users to create feedback control strategies directly inside the spreadsheet. Most control strategies can be implemented as a series of equations operating on input data and controlling an output.

Each equation is entered into a spreadsheet cell. Equations can be included which directly access A/D and D/A hardware from a variety of vendors. The results of the equations are automatically stored in the cell each time the spreadsheet is evaluated. Cells can be given symbolic names.

Port numbers can be interactively assigned by the user to actual hardware or to shared memory I/O implemented with data modules. The user can also configure how each of the 128 pages of the ControlCalc spreadsheet is evaluated and at what priority. Evaluation modes include cyclic (down to 2 ticks), event-driven or interrupt driven.

With deterministic cyclic scheduling and the mathematical flexibility of the spreadsheet, ControlCalc can be used to implement almost any analog control strategy. ControlCalc also provides the standard spreadsheet INDEX function for accessing tables of data within the spreadsheet. This feature can be used for sensor compensation, non-linear transfer functions or state-table strategies. In order to optimize performance, ControlCalc supports both 32-bit integer and 64-bit double precision floating-point equations.

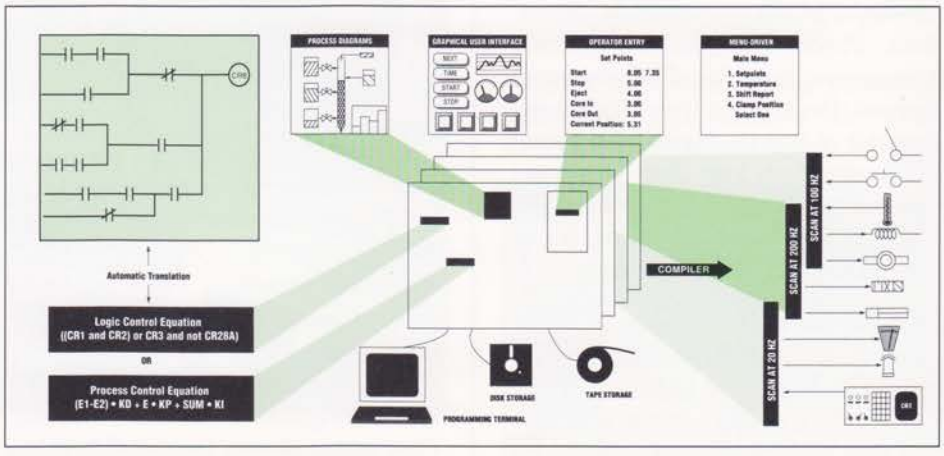
Digital Control Laws with Ladder Diagrams

ControlCalc includes an on-line ladder diagram translator and editor. Boolean equations can be entered with the ladder editor or in equational form, with interactive translation between both forms. The equations are stored in Boolean cells in equational form and are converted to diagrams in the ladder editor on demand. In this way, Boolean equations are part of the spreadsheet while still able to be visualized in ladder form. If a ladder is being displayed while the scans are running, the rungs are continuously highlighted to show the active "circuits" to aid debugging. The ladder diagram uses only ASCII characters and can therefore be used on any serial device. It does not require a bit-mapped graphics display.

IF Functions, Timers and Counters

The IF function is an extremely flexible tool for a number of requirements, including timers and counters. In general, the IF function takes three parameters. The first must be a Boolean expression which results in an ON/OFF or TRUE/FALSE result. That result is tested to determine which of the other two parameters is evaluated. The IF function returns the result of the evaluation of the selected parameter. The only restriction is that both the selectable parameters must be the same type of expressions, whether that be Boolean, integer or floating point.

There is also a select function which acts like a C language switch statement (or a CASE statement in PASCAL). This function uses its first parameter to deter-



System overview of RTWare's ControlCalc real-time spreadsheet.

mine which of a list of parameters to return.

Once the person creating the application has entered the equations and selected the scan options, the application portion of the spreadsheet can be saved as a separate data file. This file is called a spreadsheet template and contains all the equations, text cells and other definitions which the user created. This data file can be converted into an OS-9 or OS-9000 data module with a supplied utility program, and loaded and autostartedfrom either memory or the file system.

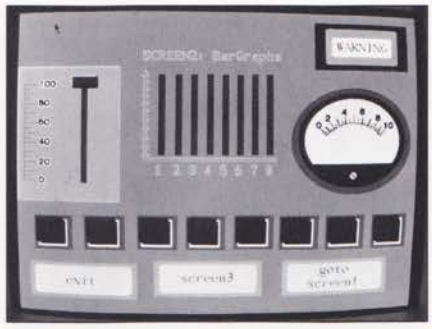
RAVE: No Programming Required

Sophisticated graphical user interfaces are provided by the integration of Microware's RAVE package. RAVE is an object-oriented graphics and audio package designed for control users.

RAVE itself produces a set of C language subroutines which allow access to any screen created within RAVE. Normally, an application program must be written to drive the screens and objects as required. ControlCalc eliminates this programming requirement by exploiting the common object-oriented nature of both RAVE and spreadsheets. A utility is provided which will build user-created RAVE screens directly into ControlCalc. The utility is told only the name of each screen available. Once the screens are linked into ControlCalc, a graphics interface utility allows the interactive attachment of screen objects to spreadsheet cells. The objects are referenced by the names the user gives them in the RAVE editor, and those names are read automatically by ControlCalc from the files produced by RAVE.

There are three ways to attach cells to screen objects. Indicators can be attached to any spreadsheet cell which has a value, allowing indicators to continuously reflect the results of any control result. Activators can be attached to setpoints, either numeric or logic (for push-buttons). Finally, Activators can be attached to spreadsheet macro cells. ControlCalc has command macros which, for example, display a new graphics screen. The operator can then use a complete set of graphic control panels which are directly interacting with the control system.

Once RAVE objects are attached to cells, ControlCalc automatically updates the displays as required and handles all the operator actions automatically within the spreadsheet.



Sample RAVE screen for ControlCalc.

Memory Module I/O

Given the wide range of I/O devices and interfaces on the market today, ControlCalc includes a shared memory I/O facility which hides the device details from the control system. Shared memory I/O allows the user to define an I/O port as shared memory rather than as a physical device. Shared memory size is limited only by the amount of memory in the system. The control equations use the normal input and output functions to access shared memory ports, but the channel selection parameter actually functions as an index into the memory area. At the start of each shared memory area is a table specifying the sizes and types of data in the area.

Shared memory is implemented with OS-9 and OS-9000 data modules, which lets any other program in the system "link" to the memory, and read and write values in ControlCalc. Combined with the SIGNAL function, shared memory allows drivers to be added to the system without requiring any change to ControlCalc itself. These drivers could be network interfaces for high-speed data acquisition, serial drivers communicating with instrumentation or links to other slave processors in the system. With advanced bus systems such as VME, shared memory multiprocessing can be implemented quite easily.

The user can also configure an I/O port to be the "mirror image" of another shared memory I/O port. This means that equations can read the outputs from one port as inputs from the other. Since the channel number functions as an index into shared memory arrays and can be computed, the control scans can treat shared memory I/O as unlimited size buffers for data acquisition and history recording. One interesting application area is simulation. One scan can be doing feedback control, while another could implement the simulation equations for the process under control. The control scan would be unaware that the I/O port it uses is a memory module. This allows rapid testing of control strategies or creation of off-line training systems using the same control system as is used in production. To convert the simulator to an actual controller, the user only needs to redefine the I/O port to access a physical I/O controller and turn off the simulation scan.

ControlCalc: 4GL For Real-Time Control

With real-time extensions to spreadsheet programming, ControlCalc is a fourth generation language for control systems. It moves beyond traditional sequential programming languages to a fill-in-the-blank, object oriented approach which is easily mastered. Standard features such as file support, screen displays, graphics and I/O are configured with interactive menus where the user provides only the information necessary to define the function. Interactive system design with full compiled speed combine to make ControlCalc an effective tool for a wide range of control professionals.

Richard Clarke is president of RTware, Inc. RTware will announce OS-9000 support for ControlCalc at Control Expo 1991, May 21-23 at the O'Hare Exhibition Center, Rosemount, Illinois. For more information about RTware's ControlCalc, call (919) 286-3114. MSC

New Vendor Products

Mizar Incorporated

Micro-Link

MPD International

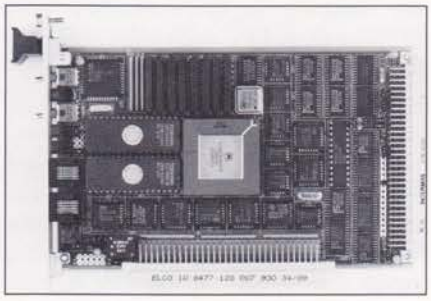
General Micro Systems

Compcontrol B.V.

Syntel Microsystems

Mizar Releases Three New Boards

Mizar Incorporated (Carrollton, Texas) recently introduced their MZ 8134, MZ 7132 and MZ 7115 boards. The **MZ 8134** is a 3U VMEbus board with a 25 MHz MC68030 or MC68EC030 CPU, 4M dual-ported DRAM, up to 1M EPROM, two RS-232 ports, an interrupt handler, mailbox interrupt support and optional battery-backed time-of-day clock. The board includes an MXbus expansion interface, allowing connection of standard and custom side modules. These modules can be used for enhanced memory, I/O or other functions.



Mizar's 68030-based MZ 8134.

The **MZ 7132** is a VMEbus CPU board that features either an MC68030 or MC68EC030 CPU at 25 MHz, optional 68882 FPCP, 16M dual-ported DRAM with parity, up to 1M EPROM, 16K external cache, interrupt handler, interrupt generator and mailbox interrupt support. I/O facilities include an on-board high-speed SCSI interface, two RS-232 ports and an optional battery-backed time-of-day clock. The board offers a 32-bit master/slave interface to the VMEbus with full system controller functions, including four-level arbitration. The MZ 7132 also provides Mizar's Function Expansion Interface which can provide additional features such as Ethernet, VSB memory expansion interface, and additional serial and parallel I/O.

The **MZ 7115** features a 68010 CPU at 12.5 MHz, 2M dual-ported DRAM with parity, 64K battery-backed SRAM, on-board high-speed SCSI interface, two RS-232 ports, two 8-bit parallel ports and a battery-backed real-time clock. The board also features an on-board four-channel 68450 Direct Memory Access Controller, interrupt handler, master/slave interface to the VMEbus and full system controller functions.

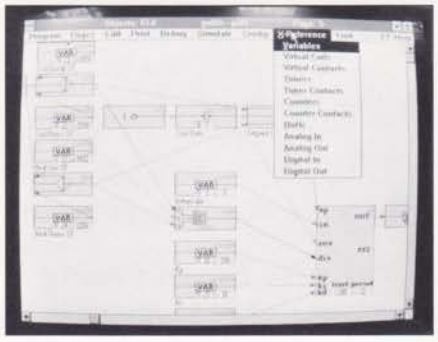
For more information, contact Mizar, 1419 Dunn Drive, Carrollton, Texas 75006. Phone: (214) 446-2664.

GELLO Now Available from Micro-Link

Micro-Link (Carmel, Indiana) recently announced its joint development with Event Technologies Inc. (Indianapolis, Indiana) to provide an object-oriented industrial control language. **GELLO** allows control engineers to develop automation and process control just as they sketch it out on paper.

GELLO is a fifth generation object-oriented programming language with a graphical user interface. GELLIX, the GELLO editor, allows control engineers to place objects, such as PID or analog I/O, on the screen and connect them using a mouse. The control scheme can be simulated on the development system for debugging and is then downloaded to the controller, where the GELLO engine runs the GELLO objects.

For more information about GELLO, contact Micro-Link Products Division, SEA-ILAN, Inc. 14602 North U.S. Highway 31, Carmel, Indiana 46032. Phone: 1-800-428-6155..



Development screen from GELLIX, the application editor for GELLO.

MPD Has New Address and BBS

MPD International (Cincinnati, Ohio) has moved to new offices and has started a bulletin board system for **Sculptor 4GL+SQL** developers. The company's new address is:

MPD International, Inc.
2826 Mack Road, Suite B
Cincinnati, Ohio 45014
Phone: (513) 870-0700
Fax: (513) 870-0800

The MPD bulletin board system can be reached at (513) 870-0529 with 1200 and 2400 baud rates available (8-N-1). The BBS supports most popular transfer protocols.

For more information, contact MPD at the address above.

General Micro Systems Introduces Three Boards

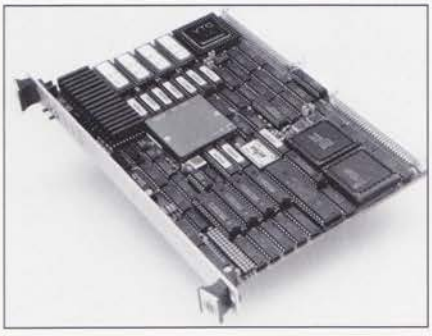
General Micro Systems Inc. (Montclair, California) recently introduced three new boards, including a 68040-based SBC.

The **GMS V49** is a 6U VMEbus SBC that features a 68040 at 25 MHz, Ethernet/Cheapernet transceiver interface,

SCSI interface, eight RS-232/422 ports, 2M or 8M two-way interleaved DRAM, up to 4M battery-backed SRAM, up to 4M EPROM and a battery-backed real-time clock. Mini-SAMs (Special Applications Modules) may be incorporated for expanded capabilities.

The **GMS V17** is available in commercial, ruggedized or militarized versions. The board features a 68030 CPU at up to 25 MHz, optional 68882 FPCP, two RS-232/422/485 ports, and unique location monitor/mailbox interrupt techniques that allow for real-time multiprocessing. The board accepts SAMs (Special Applications Modules) for additional capabilities.

The **GMS V36** features a 68030 CPU at 40 MHz, Ethernet/Cheapernet transceiver interface, SCSI bus interface, optional 68882 FPCP, eight RS-232/422 ports, 2M or 8M two-way interleaved DRAM, up to 2M EPROM, up to 1M battery-backed SRAM and a battery-backed real-time clock.

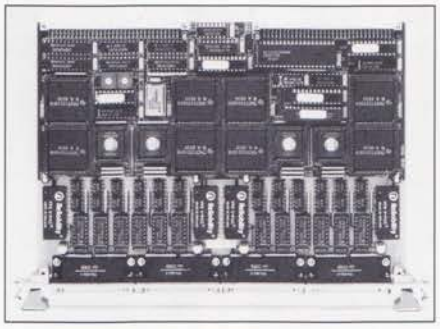


68040-based SBC from General Micro Systems.

For additional information, contact General Micro Systems, 4740 Brooks Street, Montclair, California 91763. Phone: (714) 625-5475.

Compcontrol Releases Incremental Encoder Interface Board

Compcontrol B.V. (Eindhoven, Holland) recently introduces two new VMEbus boards. The **CC133** is a 12-channel incremental encoder interface board with 500 V optical isolation on each channel. The board accepts pulse trains with frequencies up to 1 MHz from transducers having two output signals in quadrature. The CC133 has four independent groups of three channels, with 500 V isolation between each channel group. Each of the four groups has two cascaded THCT12316 triple incremental encoder counters, internal synchronization and control logic, interrupt level and vector, DC/DC converter and D25 front-panel connector.



CC133 incremental encoder from Compcontrol.

The **CC150** is a VMEbus board that features a 68000 CPU at either 8 or 16 MHz, up to 1M battery-backed static local RAM, up to 1M EPROM, watchdog, 16-bit programmable timer counter, power monitor, 8-bit input port, 8-bit output port, two RS-232 ports and IEC-603 I/O expansion connector. The board also features an A24:D16,D8 VMEbus slave interface, a programmable seven-level interrupt requester and selectable base address for shared RAM. An HDLC (High-Level Data Link Control) driver provides fast, reliable point-to-point communications links.

Compcontrol has also moved to new facilities. For more information contact Compcontrol B.V., Science Park Eindhoven 55, P.O. Box 193, 5600 AD Eindhoven, Holland, Phone: (31) 40-414025. Fax: (31) 40-414035.

Syntel Adds to Product Line

Syntel Microsystems (Huddersfield, England) recently introduced three new products—a 68302-based board, a Professional OS-9 system and an industrial I/O communications system.

The **MP302** is a 3U SBC featuring a 68302 CPU, 2M DRAM, 256K battery-backed SRAM, up to 4M EPROM, battery-backed calendar/clock, G64 I/O interface, two RS-232/485 ports, a TOPAZ LAN port and 15 digital I/O lines.

Syntel's **LC850** is 68020-based development system that features a 68020 CPU at either 16 or 20 MHz, 68881 or 68882 FPCP, up to 8M 32-bit wide DRAM, up to 4M EPROM, four serial ports, SCSI interface, VME/G-64 dual-bus interface and a battery-backed calendar/clock. Standard mass storage devices are a 2M flexible drive and 40M hard disk. Optional storage devices are a 100M hard drive and an integral 3 1/2 inch streaming tape drive.

REFLEX physically separates a computer from an industrial environment. A module at the computer end provides an interface between the computer's bus (VME, G-64 or G-96) and a high-speed, 8-bit wide I/O bus—the **REFLEX R-26 Bus**. The R-26 bus consists of a 26-way ribbon cable, up to 5 meters (5 1/2 yards), driven by noise-resistent, terminated line drivers. Along the ribbon cable, IDC connections are made to up to 16 DIN-rail mounted I/O modules.

REFLEX industrial I/O system.

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

On The C Side: Data Modules

by Ric Yeates
Microware Systems Corporation

DATA MODULES ARE AN EXCELLENT METHOD OF ALLOWING multiple processes access to the same data. A manufacturing system (with many different processes) might use a single data module to hold all the information about the work in progress. This way, a monitor process could be written that would only have to examine the data module to report on the current state rather than having to query all the processes about their current status.

The following two programs use data modules to play the "guess a number" game. ***Know.c*** compiles into a process that knows the random number and ***guess.c*** compiles into a process that tries to guess the number. ***Datmod.h*** is simply a header file shared by the two files. The general flow of the processes is this (assuming ***know*** runs before ***guess***):

KNOW creates the data module
finds the random number
waits for ***guess*** to put a guess into the data module

GUESS links to the data module
places a guess into the data module
waits for a reply from ***know***

KNOW sees the guess
places a reply into the data module
waits for another guess

GUESS places a guess into the data module based on the reply
waits for another reply from ***know***

This process continues until guess zeros in on the number. Then they both unlink from the data module and it is taken out of memory by the kernel.

In the interest of listing size, a busy-waiting method was used to wait in both processes. This is by no means the appropriate way to accomplish synchronization between processes (See page 10 of the Spring 1990 issue of *PIPELINES* for information on events).

To compile the programs:

1. type each one into files named: ***datmod.h***, ***know.c***, and ***guess.c***
2. compile ***know.c***
   cc know.c
3. compile ***guess.c***
   cc guess.c
4. run the programs from the shell
   know & guess

```
/*****************************
        datmod.h
   This file is used by both guess.c and know.c for
   common strings and other values.
*****************************/

#include <stdio.h>
#include <module.h>
#include <errno.h>

char *datmod_name = "shared";/* data module name */
char *correct_str = "Correct!";/* reply for correct guess */
char *higher_str = "Higher"; /* reply for guess too low */
char *lower_str = "Lower";/* reply for guess too high */

/* the form of the data portion of the data file */
typedef struct datmod_form {
   unsigned int reply_flag : 1;/* reply valid flag */
   unsigned int guess_flag : 1;/* guess valid flag */
   char reply[10];             /* buffer for reply */
   int guess;                  /* buffer for guess */
} datmod_form;

/* data module attribute revision word:
   re-entrant, edition #1 */
#define MOD_ATTR (MA_REENT << 8) | 0x01

/* data module permissions: owner may read and/or write */
#define MOD_PERM MP_OWNER_READ | MP_OWNER_WRITE

/* these functions return a pointer to mod_exec
   for our purposes */
mod_exec *_mkdata_module(), *modlink();
```

```
/*****************************
        know.c
   This process(know) figures a random number and lets guess
   try and figure it out. For each guess it will put a reply
   in the data module,telling guess what sort of number should
   be guessed next.
   The basic flow is:
        link/create the data module
        formulate a random number
        while "number not guessed" {
             wait for guess to flag a valid guess
             figure a response based on the guess
             copy a "hint" string into the data module
             flag that a valid reply exists
        }
        unlink from the data module
*****************************/

#include <datmod.h>
#include <setsys.h>

main()
{
   register int wrong,/* amount that guesser is wrong by */
                num;/* number guesser is seeking */
   register mod_exec *mod_head;/* data module base pointer */
   register datmod_form *data;/* data module data pointer */

   /* access/create data module */
   if ((mod_head = modlink(datmod_name, 0)) == (mod_exec *)-1)
```

Continued

```
        {

        /* check for unexpected errors */
        if (errno != E_MNF) exit(_errmsg(errno,
        "fatal error linking to data module!\n"));

        /* could not link, so try to create the data module */
        if ((mod_head = _mkdata_module(datmod_name,
        sizeof(datmod_form), MOD_ATTR, MOD_PERM)) ==
        (mod_exec *)-1)
            exit(_errmsg(errno,"can't link or create
            data module\n"));
    }
    /* compute address of usable data area in module */
    data = (datmod_form *)((char *)mod_head +
    mod_head->_mexec);

    /* make up a random number in the range 0 to 999 */
    seed(_getsys(D_Ticks,4));
    num = random(1000);

    wrong = 1;/* start with guess being wrong */
    /* while guesser is still wrong */
    while (wrong) {

        /* wait for the guesser to signal that a guess
        has been made */
        while (data->guess_flag == 0) tsleep(1);

        data->guess_flag = 0;

        /* set wrong with distance from guess to num */
        wrong = num - data->guess;

        /* fill reply field */
        if (wrong == 0)
            strcpy(data->reply,correct_str);/* got it */

        else if (wrong < 0)
            strcpy(data->reply,lower_str);/* too high */

        else
            strcpy(data->reply,higher_str);/* too low */

        _errmsg(0, "replying '%s'\n",data->reply);

        /* signal that a reply has been placed in
        the reply field */
        data->reply_flag = 1;
    }
    /* unlink from the data module */
    munlink(mod_head);
}

#define A 25173
#define C 13849

int _seed;

/* random(size) - returns a random number from
        0 to (size - 1) */
int random(size)
register int size;

{
    _seed = _seed * A + C; return(abs(_seed % size));
}

/* seed(s) - seeds random number generator */
int seed(s)
register int s;

{
    _seed=s; return(1);
}
```

```
/*******************************
                guess.c
    This process(guess) trys to guess the number that the
    process know has made up. It uses a standard binary
    search method to guess the number in the fewest guesses
    possible. The basic flow is:
        link/create the data module
        while "number not guessed" {
            formulate a guess into the data module
            flag that a valid guess exists
            wait for know to flag a valid reply
            act on the reply
        }
        unlink from the data module
*******************************/
#include <datmod.h>

main()
{
    register int lb = 0,/* lower bound for next guess */
                 ub = 999,    /* upper bound for next guess */
                 wrong;       /* 1=still wrong, 0=correct */
    register mod_exec *mod_head;/* data module base pointer */
    register datmod_form *data;/* data module data pointer */

    /* access/create data module */
    if ((mod_head = modlink(datmod_name, 0)) == (mod_exec *)-1)
        {

        /* check for unexpected errors */
        if (errno != E_MNF) exit(_errmsg(errno,
        "fatal error linking to data module!\n"));

        /* could not link, so try to create the data module */
        if ((mod_head = _mkdata_module(datmod_name,
        sizeof(datmod_form), MOD_ATTR, MOD_PERM)) ==
        (mod_exec *)-1)
            exit(_errmsg(errno,"can't link or create
            data module\n"));
    }
    /* compute address of usable data area in module */
    data = (datmod_form *)((char *)mod_head +
    mod_head->_mexec);

    wrong = 1;   /* start being wrong */

    /* while still not right */
    while (wrong) {

        /* fill guess field */
        data->guess = lb + ((ub - lb) / 2);
        _errmsg(0,"guessing %d\n",data->guess);

        /* signal that a guess has been made */
        data->guess_flag = 1;

        /* wait for the knower to signal that a reply
        has been given */
        while (data->reply_flag == 0) tsleep(1);
        data->reply_flag = 0;

        /* adjust values according to reply */
        if (strcmp(data->reply, correct_str) == 0)
            wrong = 0;   /* exactly right */

        else if (strcmp(data->reply, lower_str) == 0)
            ub = data->guess - 1;/* move upper bound down */

        else
            lb = data->guess + 1;/* move lower bound up */
    }
    /* unlink from the data module */
    munlink(mod_head);
}
```

Continued from Page Three

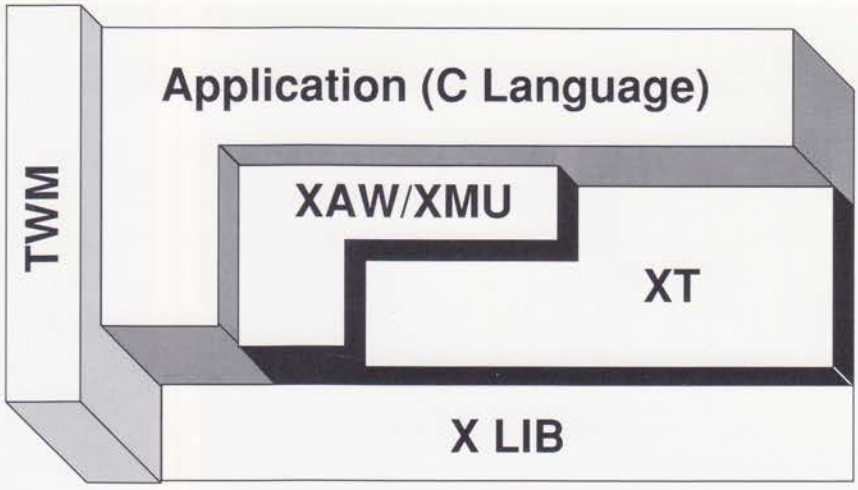
is entirely responsible for interacting with the display hardware, X Windows Client applications are highly portable.

An X Windows Client program makes requests to interact with a display (or "base window system") controlled by an X Server using the X Network Protocol. Building these requests from scratch is time-consuming and inefficient. Instead, most X applications usually interface to the standard library of C language subroutines called ***Xlib*** that is included as part of the MIT X Windows Release.

- ***Xlib***—X Window library provides the base set of X library commands for communicating between X applications and requesting base window system services.
- ***Xaw***—X Windows Athena Widgets Library is a set of abstract objects such as buttons, scroll bars, slide controls and other such objects. X Client applications are easily built from a number of "widgets". The X Client does not have direct control of the actual appearance of a widget, only its general form and size or contents. The actual appearance of a widget is determined by the "toolkit" that is employed.
- ***Xt***—X Toolkit Intrinsics Library is one of many toolkits that defines a framework for creating, deleting and managing widgets for an X Client. The Intrinsics Library simplifies the design and development of an application's user interface by providing a standard set of widget manipulation commands.
- ***Xmu***—X Window Miscellaneous Utilities Library provides some additional widget and user-interface commands that were not included as part of the ***Xt*** specification.
- ***Xdmcp***—Display Manager Control Protocol library.

The following figure shows how X Client applications interface to the X Window development libraries to produce a graphics application.

Along with the standard set of X Development Libraries, OS-9/X Windows includes a number of X Client applications that are commonly found in all X Window implementations. These Client programs provide basic X Windows services such as opening a terminal emulation window on an X Server workstation, initializing the X Window system or setting the color bitmap for an X Server.



Libraries Supported by OS-9/X Windows

X LIB	=	X Library	XT	=	X Intrinsics
XAW	=	Athena Widgets	TWM	=	Tab Window Manager
XMU	=	Miscellaneous Utility Functions			

Schematic showing how X Client applications interface to the X Window development libraries to produce a graphics application.

Under OS-9/X Windows a special X Client program called the "window manager" is run on behalf of each X Server. The window manager will typically place some form of window decoration around the outside of each X Client window that includes resize and move buttons, as well as a title bar. It then becomes a function of the window manager to resize, move or rearrange a window on the display according to the wishes of the user. Using the window manager's control menu, or a mouse or other pointing device, a user can effect changes to any of the windows on the display. Microware's OS-9/X Windows Client Development Package includes the MIT Tab Window Manager. However, OS-9/X Windows allows users to implement other standard window managers (such as OSF/Motif and OPEN LOOK) or develop their own custom window manager.

The X Window Server program on a particular workstation has complete control over the physical display hardware, including any input devices that may be connected. X Client programs send X Protocol messages to the X Server to interact with the display. X Servers, therefore, must be partially written for a specific computer and operating system, unlike X Clients, which are hardware- and operating system-independent.

The initial release of OS-9/X Windows includes a product called the OS-9/X Windows Server Source Code Package. This package includes the standard MIT X Server source code, as well as information regarding specific OS-9 implementation details. Microware customers that wish to develop an OS-9/X Server for a specific graphics system will find this package extremely useful.

OS-9/X Windows Packages

Microware offers four X Windows packages that are a robust implementation of the X Windows Systems. These packages allow OS-9 developers to take advantage of the power of the X Windows System in real-time design.

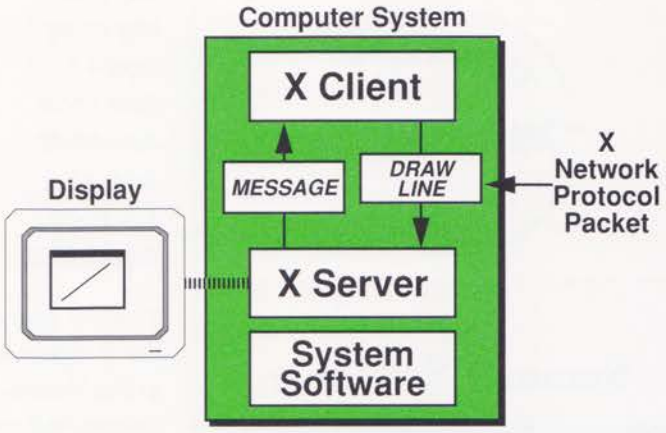
OS-9/X Windows excels in distributed environments using the Transmission Control Protocol/Internet Protocol (TCP/IP) for communication between

OS-9 systems and other remote X Windows systems. OS-9/X Windows is compatible with all versions of the X Windows System that comply with the MIT X11.4 release.

The **OS-9/X Windows Client Development Package** is a set of tools to develop X Windows applications for OS-9 systems. X Client applications developed using this package are displayed on a remote X Server over an Ethernet network. With the addition of X graphic hardware and the OS-9/X Windows Server Sample Source Code Package, an X Server can be developed for the OS-9 host.

The **OS-9/X Windows Client Run-Time Package** is for developers who have OS-9/X Client applications and want to deploy them on OS-9 target systems. The Client Run-Time Pak includes the OS-9 modules needed to bring up and administer the OS-9/X Windows system and OS-9/X Client applications.

The **OS-9/X Windows Full Source Package** includes all of the MIT X Windows System source as modified by Microware to run under OS-9. The Full Source Pak is offered for real-time developers who wish to modify or extend Microware's implementation of X Windows.



OS-9/X Windows' Client/Server architecture separates the management of a graphics display and X Client. Communications between the display manager, or X Server, and X Clients is accomplished using the X Network Protocol.

The **OS-9/X Windows Server Sample Source Code Package** includes all the standard MIT X Windows Server source code and specific information for implementing an OS-9/ X Server.

Editor's Note: Microware wishes to acknowledge the efforts of Antonio Pastore, CERN (Geneva, Switzerland). It was Mr. Pastore's initial work with OS-9 and X Windows that led to the introduction of OS-9/X Windows. MSC

Updated Products From Microware

THREE PRODUCTS HAVE BEEN UPDATED in recent months. These are primarily maintenance updates.

OS-9 N.F.S. Version 1.1

Version 1.1 of OS-9 N.F.S./RPC is now available from Microware. It includes many new features, as well as performance improvements, including:

- **VCCS Support**—The N.F.S. Server, *nfsd*, now supports OS-9 V2.4 variable sector size disk formats. It is backward compatible with OS-9 Version 2.3.
- **Performance**—When used with OS-9 Internet 1.3, the OS-9 N.F.S. system is quite a bit faster than the initial release. The improvements

OS-9 N.F.S. Version 1.1 is now available.

are a combination of *Mbuf* handling, support for larger UDP transfers in ISP, and RPC's interaction with the network via the *select()* function.

- **Access and Validation**—The OS-9 N.F.S. server now validates all file access requests. The server and N.F.S. File Manager support optional super user access via N.F.S., and aliasing super users to another group.
- **OS-9 Utilities**—Several OS-9 utilities which did not work with Version 1.0 of OS-9 N.F.S./RPC now operate correctly. These include *pd*, *dsave*, *free*, *del* and *deldir*.

OS-9000 Version 1.2

Version 1.2 of **OS-9000** includes maintenance updates to the OS-9000 utilities, C Compiler, Kernel, file managers and DEFS files.

RAVE Version 1.3

In addition to a maintenance update, some changes in **RAVE Version 1.3** may affect compatibility with previous versions, primarily Version 1.1. Most importantly, changes were made to the layout of the device static storage and the layout of the path descriptor options. Due to these changes, Version 1.3 of the Graphics File Manager (GFM) will not work with older drivers or descriptors.

For more information about these updates, contact Microware or your authorized Microware representative. MSC

Training and microware Education

Summer Schedule

The following OS-9 seminars will be held during the coming months at Microware offices in the following cities. All classes are scheduled as four-day combinations of Intermediate and Advanced Seminars. Students may sign up for an entire session or for the individual Intermediate or Advanced portions.

May 7-10	*Des Moines, IA*
May 21-24	*Santa Clara, CA*
June 4-7	*Santa Clara, CA*
June 11-14	*Santa Clara, CA*
June 25-28	*Des Moines, IA*
July9-12	*Des Moines, IA*
July 23-26	*Des Moines, IA*
August 13-16	*Des Moines, IA*

To make reservations or for additional information about classes in Des Moines, call Mike Ahrens, training and education coordinator, at (515) 224-1929. For classes in Santa Clara, reservations can be made by calling Bret Farnum, at Microware's Western Regional Office, (408) 980-0201.

Microware staff is also available for on-site training for your company. On-site training is designed for companies that have five or more students, and allows users to learn more about OS-9 or OS-9000 for their particular applications.

If you need additional information about Training and Education programs, please call Microware or your authorized Microware representative. MSC

A Microware Training & Education Seminar.

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