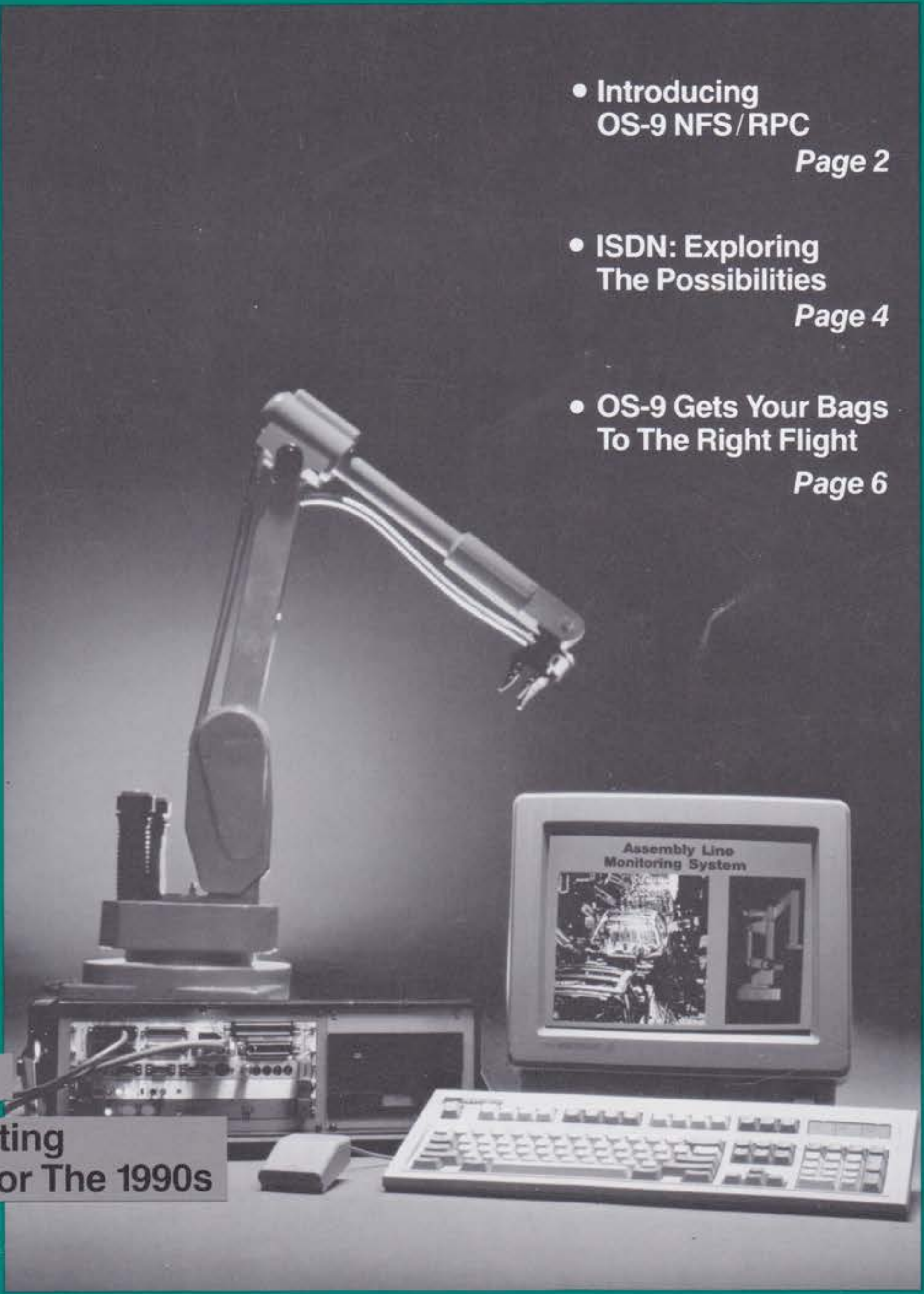
PIPELINES

Covering Microware's Real-Time System Solutions

Volume 5 Number 2

Spring 1990



Connectivity:

Microware Is Setting The Standards For The 1990s

microware®

PIPELINES

Spring 1990
Volume 5 Number 2

PIPELINES is published quarterly by:
Microware Systems Corporation
1900 NW 114th Street
Des Moines, Iowa 50322
(515) 224-1929

Publication Coordinator:
David F. Davis

Editor:
Steve Simpson

Microware Contributors:
Kim Kempf, Mike Burgher, Peter Dibble, Dave West, Yeongleh Lee, Mike Ahrens, James Jones, Ric Yeates

Photography:
David F. Davis

Art Director:
Polly Steele

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.

Call for Articles

In addition to new product listings, Microware encourages our customers to submit articles on unique applications, new procedures or comments about Microware products. Please submit an outline of your article to the Editor of *PIPELINES* for consideration.

Cover Photo

Factory floor automation is just one of the areas that requires powerful networking. Some examples of this connectivity under OS-9 and OS-9000 are covered in this *PIPELINES*.

Evolving from LANs to WANs

Ken Kaplan, Microware's President

One of OS-9's great strengths is its capabilities and flexibility in communications-oriented applications. This month's announcement of the Network File System (NFS) for OS-9 rounds out Microware's offering of local area network solutions for almost any requirement.

Much attention has been focused on the revolution in LAN technology during the last few years. However, the computer and communications industry are at the threshold of revolution in connectivity: wide area networks (WANs).

Introducing OS-9 NFS/RPC: Transparent Networking Across Multiple Platforms

by Michael Burgher

OS-9 NFS/RPC is a package of distributed network applications that provide many useful high-level services between OS-9 and other systems on a local area network.

OS-9 NFS (Network File System) provides transparent remote access to shared file systems over the network. The OS-9 NFS/RPC package contains a complete implementation of NFS, allowing OS-9 systems to access disk files on other systems as if they were on local devices, via the standard OS-9 I/O interface. OS-9 can also be an NFS file serv-

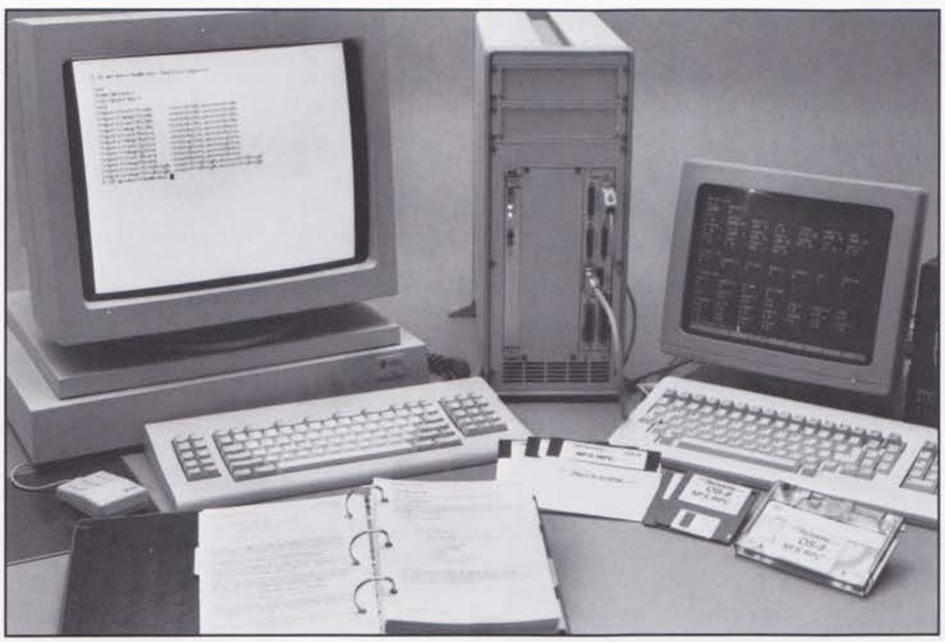
Mike Burgher

This may have even more impact than LANs in the long term. It will allow inexpensive, reliable connections at reasonable speeds between computers in different cities, or even different continents. This eventually will even extend into our homes.

ISDN service, which will be the backbone of WANs, is now just beginning. There is still much to be done. Low-cost interfaces for board-level systems need to be developed. ISDN service is just now becoming available in a few locations. There are still unresolved industry standards and network compatibility issues. And of course, system-level software support is needed.

Microware has become involved in early ISDN research and development (see article starting on page four) because we believe that WAN technology will bring a fusion of several technologies in which we have particular interest and expertise, namely: real-time, multimedia, and telecommuncations. As a result, we hope to be able to offer you some of the first effective WAN options for your applications.

— Ken Kaplan
President
Microware



OS-9 NFS/RPC allows transparent networking across a number of platforms, including OS-9, UNIX, PC-DOS and VAX/VMS.

er, providing access to OS-9 files from remote systems. Several administration tools are included which allow OS-9 NFS to be configured to specific target requirements.

OS-9 RPC (Remote Procedure Call) is a protocol for writing distributed network applications. It allows a program to call a function which is executed on another system. RPC can be used to easily implement new distributed applications.

The OS-9 NFS/RPC package contains development tools which greatly simplify the development process. The package also contains several standard RPC services, as well as administration tools which configure OS-9 RPC.

All OS-9 NFS/RPC modules are ROMable, allowing NFS and RPC services to

OS-9 NFS/RPC
Please turn to Page Fourteen

Technical I/O Manuals Now Available

Microware now offers the **Technical I/O Manuals** for OS-9 and OS-9000 as supplements to your OS-9 or OS-9000 Technical Manuals. The Technical I/O Manuals are designed to enable system programmers to create new file managers and device drivers, or adapt existing examples to specific system needs. The information that is required for system developers and integrators is now contained in a smaller, more manageable manual.

The **Technical Manuals** now contain a system overview, an expanded description of the functions provided by the kernel, an overview of the I/O system, an expanded description of interprocess communication facilities, a description of using trap handlers, a description of the math module, an overview of the RBF file system and descriptions of the system call interfaces.

The Technical I/O Manuals contain more detailed information about the I/O system interfaces between file managers, device drivers, and device descriptors, along with detailed information for the device driver interfaces for RBF, SCF, and SBF.

OS-9 and OS-9000 Technical I/O Manuals are now available for just $75. Contact Microware or your authorized Microware representative and specify **OIO68NA68MO** for the *OS-9 Technical I/O Manual* and **OIOPINAPIMO** for the *OS-9000 Technical I/O Manual.*

The OS-9 Technical I/O Manual

ISDN: Exploring The Possibilities

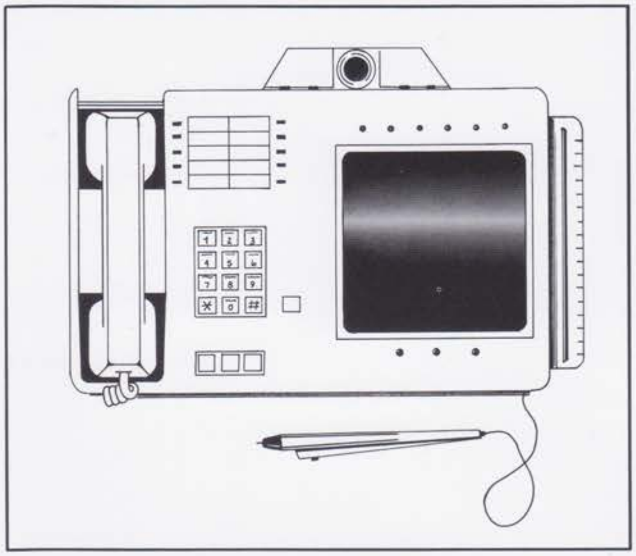
by Dr. Peter Dibble

The telephone network is a tremendously sophisticated system. Taken as a whole, it is certainly one of the world's most impressive technological achievements. However, this sophisticated transmission and switching infrastructure is used to connect telephone devices that have changed little since Alexander Graham Bell first invented them in the 1890s.

"Hissing and Squeaking"

Fax machines and modems have recently expanded the usefulness of the telephone network. This network that was originally designed to carry only voice can now accommodate images as well as anything that can be converted into computer data. But the phone system still works best for voice; other modes are forced into the mold of human voice. This is somewhat cumbersome in that the telephone system converts sound into binary data before it is transmitted over standard telephone lines.

For example, a fax machine converts an image into data, then converts the data into a sequence of "hisses and squeaks" that the telephone will accept. The telephone company's equipment converts the sounds back into data, then regenerates the "hissing and squeaking" to send to the other fax, which converts the noise back to data and then to an image. As one might surmise, this is a rather ineffective way to transmit and share data.

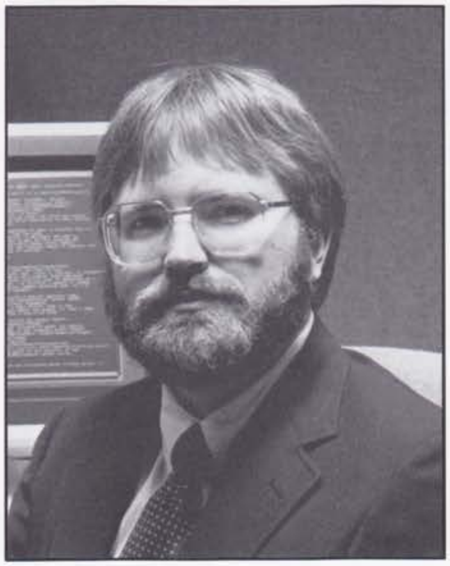


Artist's conception of MINS equipment.
Includes multifunction telelphone,
display, camera, and printer.

The next generation of telephone service, **Integrated Services Digital Network (ISDN)**, expands the capabilities of the existing telephone systems and gives it tremendous new power. Using ISDN, a telephone can now carry voice and data separately or at the same time. It will also carry voice with better quality than today's phones and carry data much faster and "better" than a conventional phone. The possibilities for this new technology in telephone service are limitless, but thus far ISDN development has concentrated primarily on the telephone equipment up to the telephone instrument itself.

Microware's Role In ISDN Research

Microware, in cooperation with the Computer Science Department of Iowa State University, is investigating the new potential in the telephone instrument. Microware brings to this project extensive experience with multimedia communication, plus more than a decade of experience in the field of demanding real-time computing (necessary to effectively implement ISDN technology).



Dr. Dibble

Microware's central role in the new Compact Disc-Interactive (CD-I) technology forms a particularly useful foundation. Iowa State's Computer Science Department is also interested in this new form of communications. Iowa State's Dr. Johnny Wong has a particular interest in ISDN and multimedia data, and is closely involved with this Microware-funded research project.

The **Multimedia Interactive Network System (MINS)** uses advanced technology to combine sound with images and other computer data and send them over a voice-grade connection. The instrument to run this software will probably look very much like a telephone. A view screen and camera will be useful options. ISDN technology will make it possible to transmit a very small "live" picture of the person speaking on the phone. However, that would be an incidental benefit of ISDN technology and not nearly as

useful as developing a method for sharing and transferring sophisticated data.

Initial Applications

A more practical application would be to use the live image capabilities of ISDN as a combination corkboard and blackboard. As an example, the participants in a telephone conversation can "join one another at the board," so to speak, and share drawings, pictures and text as if they were at a common location. They will be able to call on mechanized and human experts and share graphical information and appropriate slices of the history of the session with a new participant. Professional people will find that multimedia interactive communication offers a much richer environment to share data and information than a conventional voice connection.

MINS is also well suited to education. The MINS communication environment is close to that of a real classroom, and the two-way nature of MINS is superior to a simple televised course. MINS offers the student great flexibility in course attendance. First, a student can "dial up" a course anywhere in the world. Second, the students and teachers can phase in and out of the discussion in a 24-hour-per-day class. Participants need not miss the discussion when they are not connected. The system can save history for those off line or for review.

The initial implementations of MINS hardware will probably be too expensive for personal use. But, like with all electronic equipment, it will likely get less expensive as the technology evolves and its use becomes more widespread. Areas that hold great potential for the home/personal user will be sophisticated banking services, bulletin board-style information exchanges, comprehensive home shopping capabilities and much more — all accomplished by using only the telephone.

The current generation of ISDN and the corresponding MINS software are designed to work with the telephone wiring that is in place now. The next generation of ISDN — broad band ISDN — will require new telephone wiring (probably fiber optics), but will carry vastly more data. For example, telephones connected to broad band ISDN will be able to handle high-fidelity television pictures.

First-generation ISDN service is several years away from general availability. Furthermore, the initial offering of ISDN will probably be priced out of the range of most general telephone customers. The research Microware is now involved with will be most useful to those individuals and companies that currently have an ISDN connection and MINS instrument. However, the work that Microware is pioneering today will dramatically impact the use and availability of ISDN in the years ahead.

As Microware continues to develop and refine this new technology, the opportunity for extending ISDN into all areas of society moves closer to becoming a reality.

Dr. Dibble is a computer scientist with Microware Systems Corporation and is at the forefront of Microware's ISDN efforts. Peter holds a Ph.D. in Computer Science from the University of Rochester. MSC

Iowa Governor Terry Branstad (left) and Microware's Ken Kaplan (right) demonstrate ISDN at a recent press conference.

ISDN Press Conference

On March 8, 1990, Microware announced its joint ISDN research project with the Computer Science Department of Iowa State University, as well as the first installation of ISDN in Iowa at Microware. The announcements were made during an Iowa High Tech Week press conference held by Iowa's Governor Terry Branstad.

The purpose of the ISDN research is to explore various means of applying this emerging technology to business and personal uses.

Baggage Sortation Systems Needed Real-Time Solution

by Roland Young
Stearns Airport Equipment Co.

"OS-9 provides the complete development environment needed to efficiently design the application. By the same token, OS-9 is compact enough that no disk is required in the target system."

Stearns Airport Equipment Company, Inc. (Ft. Worth, Texas) designs and produces material handling conveyors and systems, including airline baggage handling equipment and passenger boarding bridges.

Stearns, part of Hobart Airport Systems Group, recently installed their Automated Baggage Sortation System for Delta Airlines at Phoenix's Sky Harbor Airport. The project involved replacing the previous programmable logic controller (PLC) with a new PLC and computer for enhanced performance and functionality, while using the existing belt conveyor components.

Getting Your Bags To Your Flight

When airline baggage is checked in, an adhesive bar code tag is placed on it identifying the appropriate sorting location and the baggage is loaded onto a conveyor. The bag then passes through a laser scanner array that reads the bar code tag. A serial signal is sent to the computer and a lookup table is used to determine the bag's destination, as well as the appropriate puller needed to deliver the bag to the correct pier.

A photoelectric eye synchronizes and starts a tracking routine, and a pulse encoder measures the distance from the photo eye to the appropriate puller. When a bag travels the appropriate distance, the puller is activated by the PLC and pulls the bag onto the pier. Baggage is then manually loaded onto carts that deliver it to the aircraft.

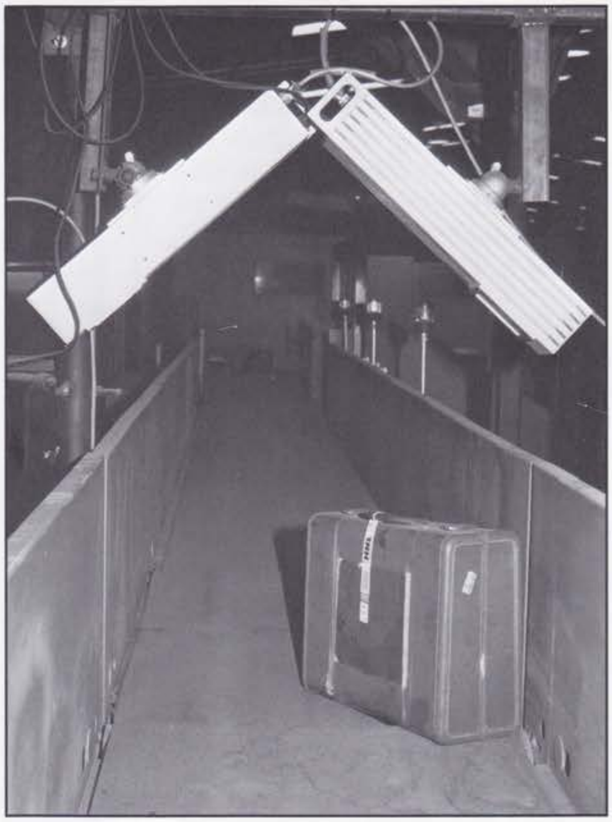
This system poses a number of real-time challenges. The computer simultaneously monitors the speed of the conveyor, the processing of the laser scanning signal and destination lookup, the pulse encoder and the PLC that controls the baggage puller mechanisms. Speed and accuracy are necessary to ensure that baggage is placed on the correct flight.

Stearns had some very specific needs for a real-time operating system (RTOS) for this project. A resident development environment, as well as a fully-developed disk I/O system was needed. Because of the rugged nature where the target system is located, with temperature ranges from O to 70°C (32 to 158°F), the RTOS also has to provide exceptional diskless operation.

Microware's OS-9 met the real-time needs for the Delta/Phoenix project. OS-9 provides the complete development environment needed to efficiently design the application. By the same token, OS-9 is compact enough that no disk is required in the target system.

System Design

Stearns' system requires both RS-422 and RS-232 serial interfaces. Bar code



Baggage passes through a laser scanner array. A bar code tag is read that identifies a bag's destination

laser scanners use the RS-232 interface and the PLC connects to the system via a simple RS-422 connection using a binary block-oriented protocol.

The system is built around a Matrix MS-CPU115 68000 CPU with 512K memory. To this, Stearns added two MS-RPN memory cards configured for the OS-9 memory modules and the ROM disk, and an MS-RPC memory card configured as the RAM disk. A Cutler-Hammer PLC provides control of the baggage pullers at each pier. Finally, Matrix's MS-SIO4 and MS-X401 provides four RS-232 and four RS-422 channels.

Specialized Features Implemented

In addition to standard OS-9 mechanisms, Stearns has implemented some specialized features for the baggage sortation system. These include a reduced periodic task time, periodic saving of data and remote access to the system.

First, a periodic task of 5 milliseconds (ms) is used for all parallel I/O. The speed of the main sortation conveyor and the degree of resolution required made it necessary for Stearns to implement this task that is one half of OS-9's standard 10 ms tick.

Next, all data on the system is saved to the RAM disk every 5 seconds. This provides a system that is all but immune to power outages. When a blackout does occur, all processing statistics are maintained. The only items not included are ones on the belt after the laser scanners. These items are not pulled onto piers, but proceed to a "runout" pier for manual sorting. When a blackout occurred with the old system, a total loss of the PLC could result. Replacing the PLC after each blackout was not feasible.

Finally, Stearns has developed four levels of remote access to the systems via a modem. At the first level, a remote system can obtain a display of the current system status including number of bags processed, percentage of bar code "good reads", and the destinations. On the second level, a user at a remote site can access an OS-9 Shell and update the system while the system is still operational. Because the C Compiler is in ROM on the baggage sortation system, files can be edited and recompiled on the Phoenix system by the remote user. The last two levels provide for remote access to the PLC and restarting of the entire system.

As the bag moves down the conveyor, a divertor is activated and the bag is moved to the appropriate pier. The bag is then loaded onto the aircraft.

Stearns chose OS-9 for its exceptional development environment and real-time performance in industrial settings. OS-9 allowed Stearns to design a system that was easily developed, offers superior performance and is simple to update.

Roland Young is Project Manager for Stearns Airport Equipment and holds a Bachelor of Science in Computer Science Engineering from LeTourneau College (Longview, Texas). He has been involved in designing hardware and software systems for real-time industrial control systems including data acquisition systems and central passenger bridge monitoring systems. He enjoys woodworking and playing chess with his sons Zachary and Lucas.

Acknowledgments: In addition to Roland Young, Morteza Mortezi was the PLC logic design and motor control system engineer on the Delta/Phoenix project. Stearns also thanks Laurent Meilleur and Stan Booth of Matrix Corporation (Raleigh, NC) for their advice and help on the development of this system. MSC

ARCNET Solutions: "Easier Than You Think"

by Paul Morgan and Colin Bartram
Comendec Limited

OS-9 users have one thing in common – they are all different, and so are their applications. ARCNET can provide networking solutions to a wide range of OS-9 users.

These options start at simple access to the network for sending and receiving packet-sized amounts of data, and go all the way to OS-9/Network File Manager.

At The Packet Level

Just sending and receiving individual packets of data is often all that is needed for real-time process monitoring and control applications, and the efficiency and speed achieved when using ARCNET to do this is always an eye-opener for first-time ARCNET users. In cases of data display, data collection and archiving, command/response in factory, process automation or anytime small amounts of data need to be moved between programs running on several types of operating systems, sending and receiving packets can be the answer.

Because ARCNET provides excellent flow control features, writing applications that use the network directly tends to be easier than with Ethernet. These packet-level drivers are all compatible so it is easy to build networks with machines running several operating systems. And, hardware poses no problem because ARCNET is available on most buses.

On OS-9, access to the network for sending and receiving packets is available through Comendec's Variable Block File driver for ARCNET (**VBF/ARC**). With VBF/ARC, the programmer defines a number of logical channels which can be accessed from high-level languages or assembler using standard OS-9 requests. VBF/ARC provides full independent, multi-channel access to the network for any number of OS-9 processes. Although network access is simple, it can be used within powerful and sophisticated applications.

Fast File Transfer

OS-9 users often need to move files between OS-9 and other platforms, such as PCs and MicroVAXs. As an example, an OS-9 system provides real-time data collection on a test rig and the collected data needs to be transferred to a PC or other system for analysis, archiving or other processing. Using Kermit over a serial line can be suitable for small files, but it can be less than adequate for larger amounts of data.

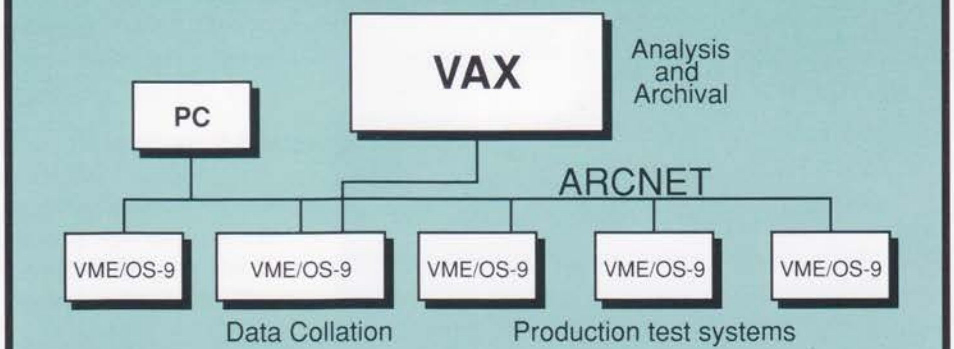
Fast File Transfer (FFT) is a simple file transfer protocol for ARCNET, developed by Comendec. With FFT, file transfers can be accomplished between any combination of OS-9, PC-DOS or VAX/VMS systems with all necessary format conversions handled by FFT. These transfers can be controlled manually or by the application, making automated backups a straightforward process.

FFT Server Command

- ***FFTSRV***: Runs an FFT server. Usually a background task on OS-9.

FFT Client Commands

- ***FFT WHO***: The ***who*** utility polls and reports all FFT servers on the network.
- ***FFT DIR***: Displays a directory of an FFT server.
- ***FFT GET***: Fetches a file from an FFT server.



Application Flexibility

In a typical application for OS-9 and ARCNET, a production testing department wants to collect product performance data from a network of OS-9/VME test systems, collate it at one VME system, and then regularly transfer it to a MicroVAX for analysis and archival. All the OS-9 systems collecting the data are running OS-9/NFM to communicate, both with the central OS-9 system and with each other for coordination, and FFT running on ARCNET provides the connection up to the MicroVAX. NFM/PC provides easy access to data on the OS-9 systems for the test department staff, when queries need immediate answers. Additional PCs can be added if the need for instant access grows.

For example, ***fft get required.txt -o:/text/newname.txt*** will get ***required.txt*** from the default server, and, in this example, the optional -o: output switch tells FFT to rename the file to ***newname.txt*** and put it into the directory ***text***.

- ***FFT SND***: Sends a file to an FFT server.
  fft send myfile.txt will send ***myfile.txt*** to the default directory on the default server, whereas
  fft send myfile.txt -o:LPT1 will result in the file being sent to LPT1 on the server. This last example shows how FFT can be used for cross-network printing.

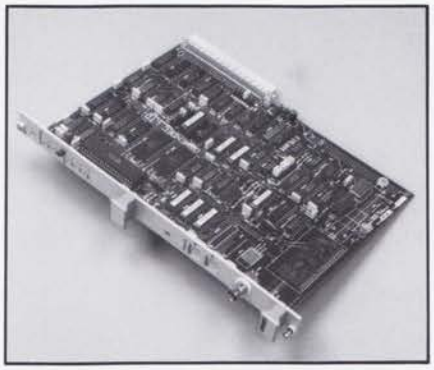
OS-9/NFM

At the top of the scale in sophistication, but still straightforward to use, is **OS-9/NFM (Network File Manager)**. This package is gaining popularity with industrial systems integrators who have realized how convenient it is to be able to get at any other OS-9 system on the network by simply extending familiar OS-9 path names.

OS-9/NFM is the ideal package for connecting a network and retaining the OS-9 architecture. NFM supports file transfer and cross-network interprocess messaging by means of cross-network pipes. In addition, all OS-9 Shell commands and related system calls can operate on remote systems over the network merely by extending the path list. For example, ***chd /net/vme2/h0/cmds*** changes the data directory to system ***vme2***, and ***load /net/ste3/h0/scratch/datafile*** loads ***datafile*** from system ***ste3***.

NFM/PC is a new package from Comendec that allows PCs to connect directly into a network of OS-9 systems running OS-9/NFM. Each PC can send and receive files to any OS-9 system on the network, even at the same time as the OS-9 machines exchange data or files with each other.

NFM/PC includes a change processor command so PCs can log onto any OS-9 system on the network. PCs can be used as system supervisors, for example, on a network of ROMed OS-9 systems. A PC can download files, log in to a remote system, start processes and log out. The package also includes an optional source library which enables PC applications to capture data directly from OS-9 machines running NFM.



Comendec's V-ARC02
ARCNET Board.

Why ARCNET for OS-9?

The key to success in all these types of applications is ARCNET firmware and the fact that all three million or so ARCNET nodes around the world conform to that firmware specification. This conformity eliminates worries like "whose ARCNET?".

The protocol handshakes carried out identically by every ARCNET chip were originally designed to provide maximum efficiency in distributed computing. This translates directly to maximum efficiency in real-time distributed applications.

ARCNET's basic idea is to provide safe, acknowledged transfer of data packets, with an effective coupling between sending and receiving systems to ensure that packets sent can be received. This last feature is unique to ARCNET and is vital to ensuring that no network time is wasted and that supervisory systems can't "drop" data packets at critical times. With Ethernet or Token Ring, a busy system can fail to catch packets sent to it by not having a buffer available and ready to receive. Such a situation usually arises, for instance, in a process alarm situation, just when you most need certainty in your network.

With ARCNET, the receiving node can't get swamped because the sending hardware checks with the receiving hardware first to make sure there's a buffer ready. And, because the sender is then holding the token, no one else can corrupt the system by getting in and sending another packet.

One common misconception is that ARCNET delays access to the network by a significant amount. The actual time spent passing the token is low, just 28 microseconds per system on the network. This fades into insignificance in relation to the overhead of standard software protocols, such as TCP. Therefore, the only time there is any significant delay in accessing the network is when there is already someone sending data. The end result is efficient use of the network, up to 96 percent of bandwidth, and predictable performance with simpler, faster software.

OS-9 users have one consistent message when it comes to ARCNET-based networking: *It's easier than you think.*

Key Feature	Ethernet	Token Ring	ARCNET
Buffer/Flow Control			✔
Data Checksumming	✔	✔	✔
Packet Acknowledge		✔	✔

This article was coauthored by Paul Morgan and Colin Bartram, Technical and Marketing Directors at Comendec in the U.K. Paul is a postgraduate in microsystems design and was responsible for many of Comendec's early ARCNET designs. Colin's training as a Chartered Engineer in manufacturing keeps the company in touch with users' requirements. The company can be reached in the United Kingdom at (0)21 359 0998 and in the United States through C&C Technology at (708) 231-0015. MSC

On The C Side

Events

When writing your real-time applications it sometimes becomes necessary to synchronize two or more processes. This synchronization may be needed because two processes want to access shared resources such as data files, data modules, pipes, or CRT screens. This can easily be done using ***events***. OS-9 events are multi-value semaphores. The multi-value aspect of events makes them a very powerful tool when it comes to implementing semaphores.

An OS-9 event is a 32-byte system data structure maintained by the Kernel. Each event on the system contains the following fields:

- **Event ID:** This number and the event's array postion in the event path table are used to create a unique ID number. (2-bytes)
- **Event Name:** Twelve-character (max.) name used to reference the event. (12-bytes)
- **Event Value:** Integer value containing the actual value of the event. (4-bytes)
- **Wait Increment:** Value which is added to the Event Value after a process has done a successful wait for the Event Value to be in a certain range. (2-bytes)
- **Signal Increment:** Value added to the Event Value whenever this event is signaled. (2-bytes)
- **Link Count:** Keeps track of the number of processes using the event.(2-bytes)
- **Next Process:** Pointer to the next process in the event queue. A circular queue is kept of waiting processes.
- **Previous Process:** Pointer to the previous process in the event wait queue.(4-bytes)

The two most common operations performed on events are Wait and Signal. The Wait operation suspends the process until the event value is within a specified range, adds the wait increment to the current event value, and then returns control to the calling process. The Signal operation adds the signal increment to the current event value, and then checks to see if other waiting processes should be awakened.

For example, events could be used to synchronize the use of a printer. The event value could be initialized to one, which represents the number of printers on the system. The wait increment is then initialized to -1, denoting that when a printer is in use the event value will be decremented by one. And, the signal increment is +1, denoting when a process is done printing, it restores the event value back to its original value. When a process wants to print it must follow a protocol of checking to see if a printer is available by waiting for the event value to be in the range of (1, number of printers).

An event value within the specified range indicates that the printer is available; the printer is then immediately marked as busy (by applying the wait increment of -1 to the event value) and the process is then allowed to use the printer. An event value out of range indicates that the printer is busy and the process is put to sleep on the event queue.

When a process is finished with the printer, the process signals the event by applying the signal increment to the event value. Then the event's queue is searched for a waiting process to wake up. If such a process is found, the process is activated, it then applies the wait increment to the event value, and then uses the printer.

To coordinate sharing of a non-sharable resource, programs must:

- Create or Link to an event. ***_ev_create()*** or ***_ev_link()***
- Wait for the resource to become available. ***_ev_wait()***
- Use the resource. /* ***Go ahead and print*** */
- Signal that resource is available again. ***_ev_signal()***
- Unlink from or delete the event. ***_ev_unlink() _ev_delete()***

The following program uses a binary semaphore to illustrate the use of events. To execute this example:

- Type the code into a file called ***semaphore1.c***
- Copy ***semaphore1.c*** to ***semaphore2.c***
- Compile both programs
- Run both programs with the command ***$semaphore1& semaphore2***

The program creates an event with initial event value of 1 indicating no one is using the resource, a wait increment of -1 and a signal increment of +1. Then the program enters a loop and waits for the event. After ten times through the loop, the program unlinks itself from the event and deletes the event from the system.

```
#include <errno.h>

#define ONLYONE 0
#define ALLPOSSIBLE 0x8000

char * event_name = "semaphore"; /* name of event to be used */
int ev_id; /* id number of event */

main()
{
    int count ; /* loop counter */
```

```
    /* create or link to the event */
    if ((ev_id = _ev_creat(1,-1,1,event_name)) == -1)
        if ((ev_id = _ev_link(event_name)) == -1)
            exit(_errmsg(errno,"Can't create or link to event"));

    for(count=0;count <10; count++) {
        /* wait on the event */
        if (_ev_wait(ev_id,1,1) == -1)
            exit(_errmsg(errno,"error waiting on event"));

        /* You are now able to use the resource. */
        /* Emulate this by printing a message to the screen. */
        /* Note: Error messages print the processes' name also.*/
        _errmsg(0,"Is now using the resource\n");
        sleep(2); /* Simulate doing something useful */
        _errmsg(0,"Is now giving up the resource.\n\n");

        /* Give up resource by signalling the event */
        if (_ev_signal(ev_id,ONLYONE) == -1)
            exit(_errmsg(errno,"Error signalling the event "));
    }

    /* now unlink from the event */
    if (_ev_unlink(ev_id) == -1)
        exit(_errmsg(errno,"error unlinking from event "));

    /* And now delete the event if it's link count is zero */
    if((_ev_delete(event_name) == -1) && (errno != E_EVBUSY))
        exit(_errmsg(errno,"error deleting event from system"));

    _errmsg(0," Terminating Normally.\n\n");
}
```

OS-9/Internet Update Brings Increased Speed, Raw Socket Support

OS-9/ISP (Internet Support Package) Version 1.3 features throughput speeds two to three times faster than previous versions of ISP. In addition, Ethernet support has been added to ISP to provide an interface to raw Ethernet sockets.

OS-9 Internet Support Package.

Engineers at Microware have fine-tuned TCP/IP (Transmission Control Protocol/Internet Protocol) algorithms to greatly increase throughput capabilities of OS-9/ISP. As a result, ISP users will see throughput increase approximately two to three times, depending on their particular hardware.

This new version of ISP will also include full support for raw Ethernet sockets, in addition to current support for stream sockets and datagrams. Users now have the option to send raw data directly between systems, thus increasing the speed of Internet communications.

Other major new or improved features with this version include:

- Upgrade of the ***arpstat*** utility and loopback driver.
- Support for dynamic network routing.

OS-9/ISP Version 1.3 will be released on June 1, 1990. Contact Microware or your authorized Microware distributor to order or for more information. MSC

Using I$ReadLn From Microware BASIC

by James E. Truesdale
JBM Electronics Company

The ***syscall*** assembler subroutine published in the appendix of the OS-9/68000 Microware Basic Manual can be used to implement OS-9 system calls from Microware BASIC. The following demo procedure runs an equivalent of the ***READ LINE*** statement available in other versions of BASIC. The techniques shown here can be used as a model for using syscall to call other operating system routines from BASIC.

The calling routine has to set up an array to hold copies of the 680X0 registers to be passed to the operating system call via the syscall subroutine. The register storage area is defined by the variable regs in the demo program. See pages 10-13 through 10-15 of the Microware BASIC Manual (Revision E) for an explanation and diagram of how BASIC passes parameters to machine language modules such as syscall.

The syscall routine is loaded into system memory at the start of the program so that it can be found in memory when called instead of out on disk. To determine the numerical value that corresponds to the desired operating system call, examine the output from ***rdump -a /dd/lib/usr.l***. The output shows that the symbol I$ReadLn has a value of $8B, which is the value used in the program.

When end-of-file (EOF) of the input file is detected, the input file is closed and the syscall routine removed from memory since it is no longer needed.

For more information about this demo procedure, contact James E. Truesdale, JBM Electronics, 4645 LaGuardia, St. Louis, Missouri 63134-9906. Phone: (314) 426-7781.

```
(* File: demo
(* James E. (Jay) Truesdale, JBM Electronics, 3/1990
(* Program to demonstrate the implementation of a READ LINE routine
(* using the syscall subroutine and the I$ReadLn system call.
(* The user is prompted for a filename which is then opened and read
(* line by line and echoed to the screen.

DIM file_path:BYTE
DIM line:STRING[81]
DIM path_name:STRING[40]
DIM regs(14):INTEGER

SHELL "load syscall;link syscall"

INPUT "Enter file name: ",path_name
OPEN #file_path,path_name:READ

regs(1):=file_path\ (* setup path number in d0
regs(9):=ADDR(line)\ (* addr of variable in a0

WHILE NOT(EOF(#file_path)) DO
    (* Store max bytes to read in d1. Using the SIZE
    (* function makes the code easier to maintain, and
    (* says precisely what is meant.
    regs(2):=SIZE(line)
    RUN syscall($8b,regs)
    (* Wipe out the ending <CR> (if there is one) with a null
    (* so BASIC can later process the string properly.
    IF PEEK(ADDR(line)+regs(2)-1)=$0d THEN
       POKE ADDR(line)+regs(2)-1,$00
    ENDIF
    PRINT line
ENDWHILE

CLOSE #file_path
SHELL "unlink syscall"
END

(* End of program demo *)
```

New Faces At Microware

Debora Jackson has transferred to Microware's Western Regional Office as office manager. Before taking the position in Santa Clara, Deb was administrative assistant to Ken Kaplan at Microware's corporate offices.

Tammy Parsons joins Microware as Ken's new administrative assistant. Prior to Microware, she was administrative assistant to an executive vice president with a regional department store in Des Moines, Iowa. Tammy is an avid water skier and enjoys aerobics, horses and drag races.

Microware's new Manager of Technical Documentation is *Eileen Beck*. Eileen comes to us from Greyhound Lines, Inc., where she was the senior documentation specialist. Eileen also was a documentation and education specialist for Financial Information Trust, a service bureau for financial institutions. Eileen holds a Bachelor of Arts degree from the University of Northern Iowa and enjoys alpine skiing, bicycling and gardening.

Daryl Jarman comes to Microware as a technical support engineer. Before joining Microware, Daryl was a system engineer with Voice Response, Inc. (Davenport, Iowa). Voice Response is a Microware OEM that develops interactive voice messaging systems. Daryl holds Bachelor of Science degrees in both computer engineering and computer science from Iowa State University. He enjoys strategy and role playing games.

Theresa Hadaway has joined Microware's accounting department as an accounts receivable accountant. She comes to Microware from a local medical center where she was a data analyst. Theresa holds an Associate of Arts degree in accounting and business administration from Des Moines Area Community College. She enjoys bicycling, reading and cooking. MSC

Pictured clockwise from upper left: Tammy Parsons, Theresa Hadaway, Eileen Beck and Daryl Jarman (seated).

OS-9 Version and Edition Numbers

Microware provides this product update and edition number information to assist OS-9 users in determining whether or not they have the latest update or edition of a software product. This list only includes those products that have changed since the Winter 1990 edition of *PIPELINES*.

You can use the ***ident*** utility to check your software against this list. If you don't have the latest editions or updates, call your system manufacturer or Microware for information on updating your software.

Language Modules

Module	Edition
cc	#44
cpp	#41
c68	#357
c68020	#357
o68	#19
r68	#67
r68020	#95
l68	#64

Module	CRC
cio	$E9ADF1
cio020	$2CCEB9

System Utilities

Module	Edition
spl	#31
splman	#31
splprt	#31

OS-9 Version 2.3 Updated Products

Product	Current Version
Atari/OS-9	2.3
Basic Runtime Module	2.3
Cross C Compilers (all)	3.2
Internet Support Pak	1.3
PCBridge	1.2
Resident C Compiler	3.2

Continued from Page Three

operate in a diskless OS-9 environment.

OS-9 Network File System

- Remote access to shared file systems
- OS-9 NFS file server configurations
- Industrial OS-9 configurations (diskless) with access to remote file systems
- Internet to OS-9/Net gateway
- Network administration tools and utilities

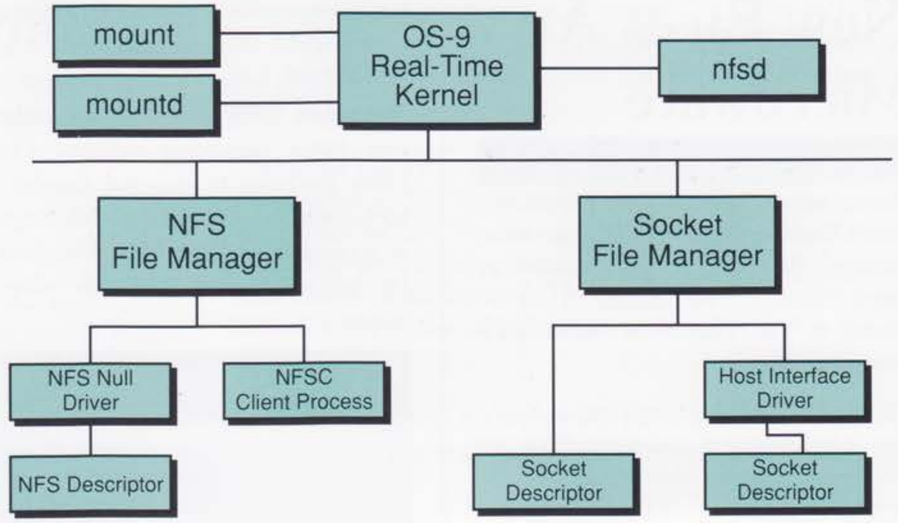
NFS has become the de facto industry standard for sharing file systems across a network. NFS is machine-, operating system-, network architecture-, and transport protocol-independent because it uses RPC. NFS implementations exist for such diverse machines as PCs, workstations, mainframes and supercomputers.

OS-9 provides a full implementation of NFS with both client and server capabilities. Utilities and network administration tools are also provided allowing configuration of NFS, remote access requirements, backup requirements and display of NFS statistics.

OS-9 Remote Procedure Call

- Call a C language function which is executed on remote systems
- Arbitrary application data structures
- Advanced features such as control of retry and time-out parameters, RPC broadcast, RPC callback procedures and batching RPC requests
- Remote execution of a process, system statistics and user information
- Development tools for implementation of new RPC services

RPC is a protocol for writing distributed applications. It provides a method for executing a C language function remotely (on another CPU or computer system). Each system on the network can provide any number of servers (remote procedures) that can be dynamically called by client programs (standard utilities or other application programs) on other systems.



Example of OS-9 NFS/RPC configuration

These applications are easy to implement because they avoid low-level primitives such as sockets. Applications are written and tested on stand-alone systems and then later split easily into "client" and "server" procedures. Remote procedures are called in the same manner as local C language functions. When remote procedures are called, RPC sets up a client-to-server communications link, and sends the data to the server. When the data arrives, the server calls a dispatch routine, performs whatever service is requested, sends back a reply and the procedure call returns to the client.

RPC can handle arbitrary data structures, regardless of different systems' byte orders or structure layout conventions. RPC uses the XDR (EXternal Data Representation) standard for transferring data across the network. XDR defines a representation for all data while on the network regardless of the data representation on the host system. This allows OS-9 to interact transparently with any other system supporting RPC.

System Requirements

OS-9 NFS/RPC uses Transmission Control Protocol/Internet Protocol (TCP/IP) and User Datagram Protocol/Internet Protocol (UDP/IP) as transport layers. The target system must support the Internet Protocol to run RPC/NFS. Microware provides OS-9 Internet networking features and capabilities in a number of products, configurable to specific target hardware requirements. These products include OS-9/Ethernet Support Package (ESP), OS-9/Internet Support Package (ISP) and UniBridge®.

Installation of the OS-9 NFS/RPC package requires:

- One system running Professional OS-9 Version 2.3 or later
- One Internet support option (ISP, ESP or UniBridge)
- One megabyte of available disk space

Runtime requirements for OS-9 NFS/RPC vary and are dependent on the target configuration and which RPC services are included. OS-9 NFS/RPC can operate in a ROM-based environment, with Industrial OS-9, Internet, and NFS/RPC in ROM.

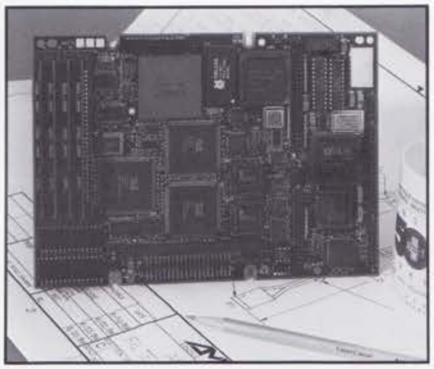
Mike Burgher is the Technical Manager for Microware's Western Regional Office (Santa Clara, California). He is the principal designer of NFS/RPC under OS-9, and has 15 years experience in computer systems software, networking, and real-time applications. He has a Bachelor of Science from Drake University (Des Moines, Iowa). MSC

New Vendor Products

Ampro Brings 80386 SBC for OS-9000 To Market

Ampro Computers recently launched their **Little Board/386**, a complete 80386 system designed for embedded OS-9000 applications. The Little Board/386 measures just 5.75 × 8 inches (146 × 203 mm) and consists of a 20 MHz 80386 CPU, up to 4M RAM, multi-density flexible disk controller, battery-backed real-time clock, dual RS-232 serial ports, a parallel printer port and an optional high-performance SCSI interface.

Expansion boards, called **MiniModules** measure 3.5 × 3.8 in. (89 × 97 mm) and fit within the board's form factor envelope. These MiniModules add CGA, EGA, LCD and VGA display interfaces, additional system I/O, a 2400 baud modem, LAN interface and intelligent disk drive interface capabilities.



Little Board/386 from Ampro.

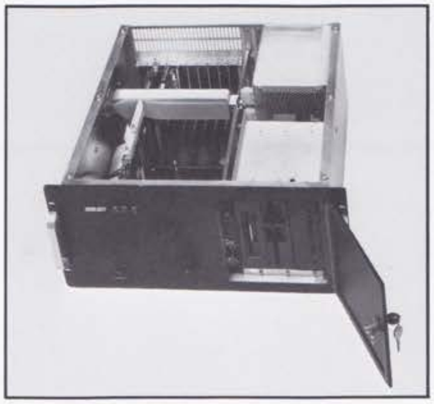
For more information, contact Paul Rosenfeld, VP Marketing, Ampro Computers, 1130 Mountain View/Alviso Road, Sunnyvale, California 94089. Phone: (408) 734-2800.

386 SBC and System for OS-9000 from Qualogy

Qualogy now offers an 80386-based SBC and complete system. The **QPC-5142** is a 386 SBC for passive backplanes. The board operates at 16, 20 or 25 MHz, and features up to 8M on-board DRAM, two serial ports, one parallel port, a clock/calendar and a coprocessor socket.

The **QPC-386IC** is a complete 386 system, including a 20 MHz QPC-5142 board enclosed in a 10-slot QPC-7200 chassis. The system includes 1M DRAM, one flexible disk drive, one 40M hard drive, EGA card and keyboard.

For more information, contact the Sales Department, Qualogy, Inc., 1751 McCarthy Blvd., Milpitas, California 95035. Phone: (408) 434-5200.

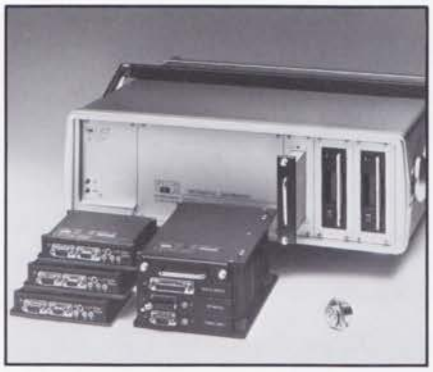


QPC-386IC from Qualogy.

SPECTRALAB Offers Mobile Data Collection System

SPECTRALAB (Kilchberg, Switzerland) now offers **SPECTRAPOT**. This mobile data collection system is ideal for independent avionic and vehicle load tracking, as well as structural, flight and fatigue data recording.

The SPECTRAPOT features a 68000/070 CPU, and can accept input from eight or more analog channels and eight or more binary inputs. Data is recorded on a solid state memory cassette of up to 2M. This removable memory cassette can be shipped to a stationary VMEbus system for further inspection.



SPECTRAPOT
Data Collection System.

SPECTRAPOT is currently used in aircraft, cars, trucks, trains and construction equipment. Current programs include sequential peak valley, real-time rainflow, fatigue indexing, range pair counting and pattern recognition.

For more information, contact Bruno Fricker, SPECTRALAB, Brunnenmoosstrasse 7, CH-8802 Kilchberg, Switzerland. Phone: (41) 1 715 38 07.

EKF Introduces Intelligent Timer/Counter

EKF Elektronik GmbH (Hamm, West Germany) has introduced its **VME 68390-TC** VMEbus module. The VME 68390-TC is based on two Am9513A System Timing Controller Chips and provides 10 individually programmable 16-bit up/down counters. Each counter has its own clock input, one-shot or continuous output, and two additional gates.

The module is designed for a variety of counting, sequencing and timing applications including programmable frequency synthesis, programmable duty cycle waveforms, retriggerable digital one-shots and more.

For more information, please contact B. Kleeburg, EKF Elektronik, Weidekamp-

NEW VENDOR PRODUCTS
Please turn to Back Cover

NEW VENDOR PRODUCTS

Continued from Page Fifteen

strasse 1a, D-4700 Hamm 1, West Germany. Phone: (49) 2381 12630.

Arcom Offers New 68020 and 68000 CPUs, Paged Memory STEbus Board

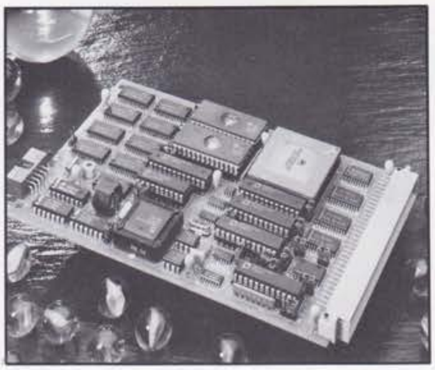
Arcom Control Systems (Cambridge, England) has introduced two new CPU boards, as well as a new paged memory expansion board for STEbus users.

The **VSC020T** provides dual interfaces to both VME and STE buses. The VSC020T features a 68020 CPU with cache, a socket for 68881 FPCP, up to 2M zero wait-state DRAM, two serial ports, two EPROM sockets, watchdog timer and a battery-backed real-time clock. One megabyte of the DRAM is dual-ported to the VMEbus and 32K SRAM is dual-ported to the STEbus. The VMEbus interface supports up to 32-bit data transfers and includes a single-level arbiter, a source for a system clock and reset.

Arcom's **SC68000** is a 68000 STEbus CPU. It is available in either 8 or 16 MHz and features 256K zero wait-state battery-backed SRAM, two sockets for up to 512K EPROM, two serial ports, watchdog and battery-backed real-time clock.

Arcom's page memory expansion board, named **SPER**, allows designers to build diskless systems with rugged solid-state memory for industrial applications. SPER can be populated entirely with ROM, entirely with RAM or with a mix of ROM and RAM. Two of the RAM sockets are battery-backed for 100-hour data retention in case of a power failure.

For more information, contact Paul Cuthbert, Arcom Control Systems Ltd., Units 8-10 Clifton Road, Cambridge CB1 4WH England. Phone: (44) 223 411200. MSC



Arcom's VSC020T.

microware®

MICROWARE SYSTEMS CORPORATION
1900 N.W. 114th Street
Des Moines, Iowa 50322

Bulk Rate
U.S. Postage
Paid
Des Moines,
IA
Permit #2864