THE CP/M USER'S JOURNAL

$2.95

Microsystems

Volume 4/Number 7

July 1983

Explore the wealth of technical information on RCPM and Bulletin Board systems

Telecommunications

Ben Bronson, Kelly Smith and Keith Peterson, pioneers and innovators in the field of public domain software exchange, discuss procedures and etiquette for downloading files from Remote CP/M (RCPM) and Bulletin Board (BBS) systems.

Jud Newell and Kim Levitt have compiled a comprehensive list of these systems, with telephone numbers and working hours.

Steven Fisher describes how you can use the PIP utility to download a telecommunications package such as MODEM7.

Walt Bilofsky shows you how to set up a simple command language that simplifies the process of establishing communication with another computer.

Digital Recording

Joseph Long describes how to use a North Star Horizon for experimenting with the digital recording and playback of audio signals.

Peripheral Interfacing

Tom Wiens discusses the problems involved in customizing the BIOS, especially when you have PROM routines at awkward locations.

Ted Carnevale describes how to interface the MX-80 and other printers to a North Star parallel port.

Software Review

Bruce Hunter reviews the Digital Research PL/I-86 and PL/I-80 compilers.

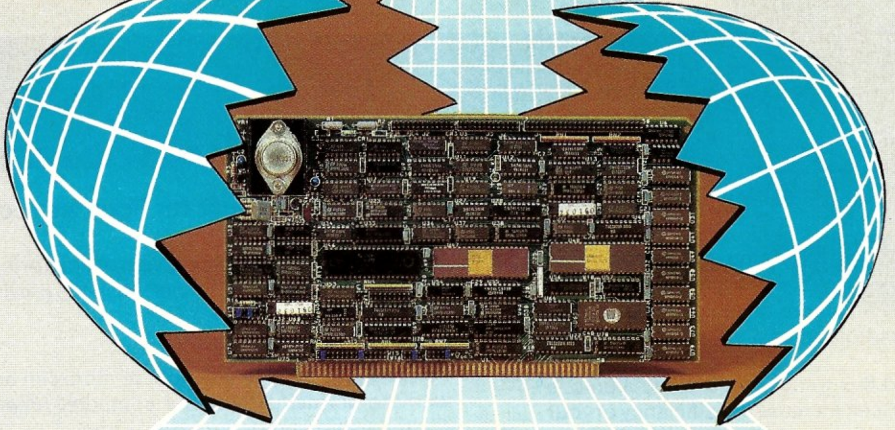
Hardware Review

Mark Zeiger reviews the PMMI MM-VT1 speech-synthesizing telephone answering board.

Richard Newrock and Walter Knesel review the Tecmar A/D and D/A converters and their associated software.

0 14024 14352 07

Incubation Complete

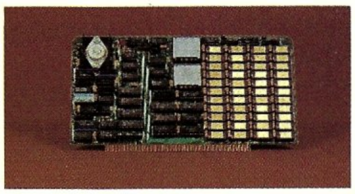


A Third Generation is Born

SBC 300
(Pictured above)

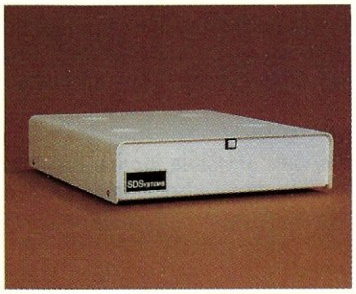
A Z80 based microcomputer board with memory and I/O functions

- ***Fully complies with IEEE 696 Standard***
- 4/6/8 MHz Z80 A/B/H*
- Supports CP/M® Plus
- Operates as bus master/slave for multi-user, multi-processor architecture.
- 64K on board memory, dual ported, parity checked
- 2 serial ports, 1 full SASI port
- All I/O drivers on board
- Memory management
- Full 24 bit address capability
- 3-16 bit CTC's



ExpandoRAM IV—Random access memory board utilizing 64K or 256K NMOS RAM chips

- ***Fully complies with IEEE 696 Standard***
- 256K capacity with 64K chips
- 1024K capacity with 256K chips
- Error checking and correction (2 bit detection, 1 bit correction)
- On board refresh
- Supports both 8 and 16 bit data transfers
- 24 bit addressing



SD300—A new series of compact yet expandable S-100 microcomputers.

- Compact size approximately 4" × 14" × 17"
- 6 Slot motherboard
- Rugged metal enclosure
- Supports up to 5 users

OEM Version: Designed for ease of integration and maximum flexibility

- Z80 CPU
- 256K RAM
- Versafloppy II with free CP/M Plus™

Discless Version: An ideal high performance system for disk intensive applications. Eliminates disk wait states for spread sheets, spelling checkers, and network operation. Utilizes SDSystems RAMDisc and ROMDisc modules.

VFW-3: A single board controller for floppy and Winchester disk drives:

• ***Fully complies with IEEE 696 Standard*** • Free copy of CP/M Plus™ included • Up to 4 floppies and three Winchester drives may be controlled by VFW-3 • Data transfers to and from board under DMA or programmed I/O control • Supports 24 bit address space.

CP/M Plus™: high performance single user operating system.

• CP/M® 2.2. compatible—no modification! • When used with SDSystems 256K memory board speeds are up to 7 times faster than CP/M® 2.2. • High performance file system • MP/M® II file password protection • Time and date stamps on files • Support for 1 to 16 banks of RAM • Support for 1 to 16 drives of up to 512 MB each • Easy to use system utilities with HELP facility • Powerful batch facility • Sophisticated programmer utilities.

RAMDisc 256: A solid state disk emulator that greatly increases system performance by eliminating disk waits in disk intensive applications. Excellent for spreadsheets, spelling checkers and software development.

• 256K capacity • 1 mb total bus capacity • CP/M® 2.2, PLUS ™ compatible • I/O port addresses user selectable • Storage locations addressed by on board 20 bit counter • On board refresh.

ROMDisc 128: An EPROM board that replaces a floppy disk drive for the purposes of booting CP/M® and loading application programs.

• Provides non volatile, permanent storage of programs and data • Utilizes 2732 or 2764 EPROMS, (16 max) • 128K capacity per board • 512K system capacity • Use with SDSystems RAMDisc to configure a stand alone or network discless system • CP/M Plus™ available in eproms • Serial port provided.

CP/M® 2.2 and CP/M Plus™ are registered trademarkes of Digital Research, Inc. *Z80 product of Zilog Corp.

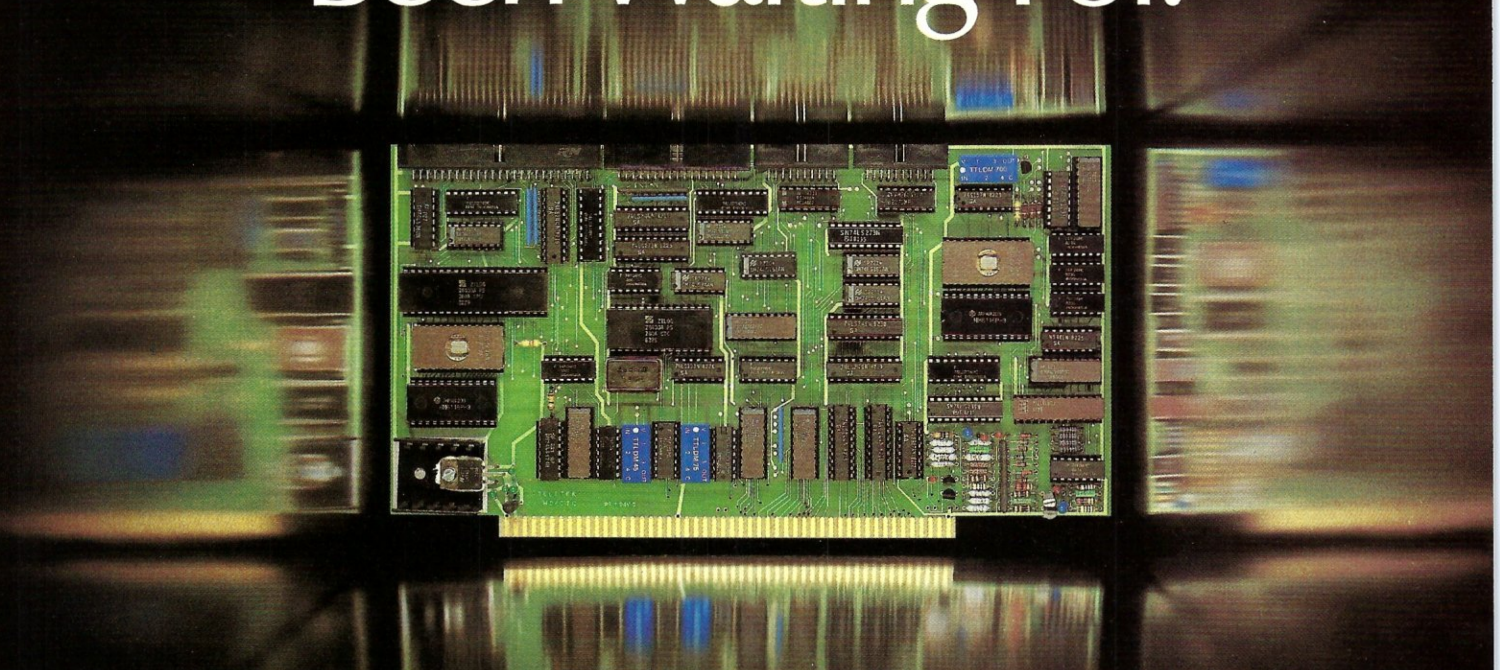
SDSYSTEMS
A Syntech Company

10111 Miller Road • Dallas, Texas • (214) 340-0303 • TLX-682 9016

CIRCLE 3 ON READER SERVICE CARD


WE DIDN'T MAKE IT CUTE,
the Grizzly™
Uninterruptible Power System
EPD
memoryguard
ON
OFF
AC OUTLETS
120 VAC - 60 Hz
UPS OUTLETS
WE MADE IT TOUGH!
The Grizzly™, EPD's uninterruptable power system is as tough as the animal it was named after. Like its namesake, don't let its looks deceive you. This compact system plugs directly into any standard outlet and is ready to go. All you need to do is plug what needs protection into it, flip The Grizzly on and proceed with your normal routine. In the event of a powerline problem such as a brownout or blackout, The Grizzly will take over instantly delivering power at full load for up to twenty minutes. Simultaneously, its sonar alarm will sound signaling you to close out files and shutdown or giving you the option of allowing line power to return.
Functioning as a source of backup or simply removing submicrosecond overvoltage line transients "spikes" or "glitches" from electrical circuits, The Grizzly gives you confidence in your ability to operate at peak performance and eliminate data base loss.
It's maintenance-free, warranteed for one year and backed by EPD's Lloyds of London-insured guarantee. It's tough. It's The Grizzly. Available through your local dealer.
EPD
Electronic Protection Devices Inc.
P.O. Box 673, Waltham, MA 02254
(617) 891-6602 • 1-800-343-1813
CIRCLE 65 ON READER SERVICE CARD

Bored Waiting? Here's The Board You've Been Waiting For.



A hard disk and cartridge tape controller together on one board? Magic? Not really. It's Teletek's HD/CTC. The hard disk and cartridge tape drive controller provide the support necessary to interface both rigid-disk drives and a cartridge tape deck to the S-100 bus.

- A Z-80A CPU (optionally Z-80B) providing intelligent control of the rigid-disk and cartridge tape drives.
- Support of 5¼″ rigid-disk drives with transfer rates of 5 megabits per second. Minor changes of the on-board components allow the support of other drive types/sizes and transfer rates up to 15 megabits per second. (Interface to disk drive is defined by software/firmware on-board.)
- Controller communications with the host processor via 2K FIFO at any speed desirable (limited only by RAM access time) for a data block transfer. Thus the controller does not constrain the host processor in any manner.
- Two 28-pin sockets allowing the use of up to 16K bytes of on-board EPROM and up to 8K bytes of on-board RAM.
- Individual software reset capability.
- Conforms to the proposed IEEE-696 S-100 standard.
- Controller can accommodate two rigid-disk drives and one cartridge tape drive. Expansion is made possible with an external card.

Teletek's HD/CTC Offers A Hard Disk Controller, Plus Cartridge Tape Controller, All On One Board.

TELETEK

9767F Business Park Drive Sacramento, CA 95827 (916) 361-1777 Telex #4991834 Answer back-Teletek

CIRCLE 220 ON READER SERVICE CARD

Contents

Microsystems

Volume 4/Number 7
July 1983

Microsystems

STAFF

Sol Libes	editor
Chris Terry	technical editor
Ian Darwin/Dave Fiedler/Dave Hardy/Bill Machrone/ Ernest E. Mau/Bruce Ratoff/Anthony Skjellum	contributing editors
Andrew Bender/David Gewirtz/Fred Gohlke/ Steve Leibson/Don Libes/Randy Reitz	assisting editors
Ann Ovodow	editorial coordinator
Nancy Metz	editorial secretary
Mariano Nicieza	art editor
Jim Beloff	advertising manager

ADVERTISING SALES OFFICES

New England, Midatlantic
Jim Beloff, *Microsystems*
Ziff-Davis Publishing Company
One Park Avenue
New York, NY 10016
(212) 725-3452

Advertising Coordinator
Rosemarie Caruso, *Microsystems*
Ziff-Davis Publishing Company
One Park Avenue
New York, NY 10016
(212) 725-5386

Southeast
Mark Browning, *Browning Publications*
P.O. Box 81306
Atlanta, GA 30366
(404) 455-3430

Midwest
Jeff Edman, *The Pattis Group*
4761 W. Touhy Avenue
Lincolnwood, IL 60646
(312) 679-1100

Southern California, Southwest
Barbara Farkas,
Ziff-Davis Publishing
3460 Wilshire Blvd.
Los Angeles, CA 90010
(213) 387-2100

Northern California, Northwest
Jeff Cohen,
Ziff-Davis Publishing
3030 Bridgeway
Sausalito, CA 94965
(415) 331-7133

Canada
Frank Lederer, *The Pattis Group*
1623 Yonge St.
Toronto, Ontario M4T241
(416) 482-6288

Direct Retail Sales
Lynn Kujawa, *Ziff-Davis Publishing*
One Park Ave.
New York, NY 10016
(212) 725-7679

CONSUMER COMPUTER & ELECTRONICS DIVISION

President	Larry Sporn
Vice President Marketing	J. Scott Briggs
Vice President Circulation	Carole Mandel
Vice President General Manager	Eileen G. Markowitz
Creative Director	Peter J. Blank

ZIFF-DAVIS PUBLISHING

Richard P. Friese	**President**
Albert S. Traina	**President, Consumer Magazine Division**
Furman Hebb	**Executive Vice President**
Paul H. Chook	**Executive Vice President, Marketing and Circulation**
Phillip T. Heffernan	**Senior Vice President**
Sidney Holtz	**Senior Vice President**
Edward D. Muhlfeld	**Senior Vice President**
Philip Sine	**Senior Vice President**
Baird Davis	**Vice President**
George Morrissey	**Vice President**
Selwyn Taubman	**Treasurer**
Bertram A. Abrams	**Secretary**

MICROSYSTEMS (ISSN #0199-7955) is published monthly by Ahl Computing, Inc., a subsidiary of Ziff-Davis Publishing Company, One Park Avenue, New York, N.Y., 10016. David Ahl, President; Elizabeth B. Staples, Vice-President; Selwyn Taubman, Treasurer; Bertram A. Abrams, Secretary.

Second Class postage paid at New York, NY 10016 and at additional mailing offices. POSTMASTER: Send address changes to MICROSYSTEMS, PO Box 1987, Morristown, NJ 07960. Subscriptions are $24.97 for 12 issues. Canadian prices are $5.00 per year additional; other foreign $8.00 per year additional (U.S. currency only). For information or questions about subscriptions phone: (800) 631-8112.

Editorial correspondence is welcomed and should be sent to: MICROSYSTEMS, One Park Avenue, New York, NY 10016. Phone: (212) 725-6856.

For information on commercial advertising, write to: MICROSYSTEMS Advertising Dep't., One Park Avenue, New York, NY 10016, or call Jim Beloff at (212) 725-3452.

Editor's Page

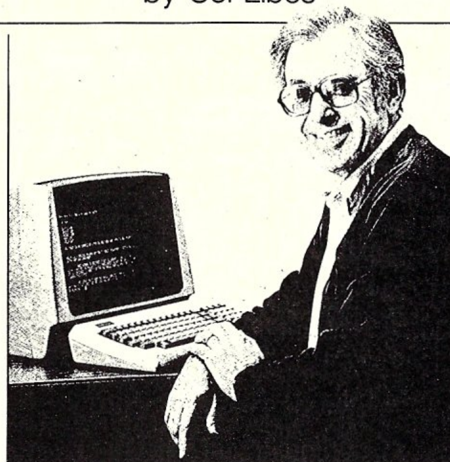
by Sol Libes

Within the last four years a loosely organized network of about 200 Remote CP/M systems (called RCPMs for short) has developed here in the United States, and also in countries such as Australia and England. Most of these systems can be used without charge. A list of these systems is provided in the article entitled "Directory of Remote CP/M Software Exchange Systems."

These RCPM systems have been the backbone of the tremendous growth of the CP/M operating system among serious amateur computer users in the late '70s, and are now turning into a communications medium for scientific and commercial users. After all, amateurs have always pioneered the way in basic technology.

These RCPM systems provide two basic services. The first is that of communications between CP/M users and between users of specific languages and hardware systems. Thus, as you study the list of RCPM systems, you will note systems that specialize in the C language, UNIX, and support for hardware systems such as the Exidy Sorcerer, etc. This communication service is provided via a "bulletin board" maintained on most of these systems. The bulletin board system (for short called a BBS) allows users of the system to leave messages for each other.

The second service provided by RCPMs is that of allowing immediate and wide dissemination of public domain software. A caller into an RCPM system will find that there are many programs that can be easily downloaded to his own system. Public domain software is software written by individuals who have chosen not to copyright their software, but rather to make it available, free, for the world to use. Most of this software is distributed on disk through groups such as SIG/M and CPMUG (see "News &



Views"). Those interested in obtaining copies of complete volumes of SIG/M or CPMUG software would do best to contact the organizations or local computer clubs to obtain the disks. However, those who want only one or a few programs can usually obtain them from an RCPM system just for the price of a phone call. A few RCPM systems already have hard disks, and have virtually the entire SIG/M and RCPM systems on line and available for downloading. For hints on how to best use RCPM systems, see the article entitled "Using RCPM Systems Effectively."

Of course, the first software that a CP/M user should obtain is software that allows modem communication and the transferring of files. A discussion of modem utility programs that are available in the public domain will be found in Chris Terry's column, "In the Public Domain."

A little history

A large number of computer amateurs have worked long and hard to build RCPMing into what it is today. If we could list them all, I am sure that the list would contain several hundred names. But several individuals have provided outstanding service and deserve recognition here. First, and possibly foremost, is Ward Christensen, who, back in the late '70s, together with Randy Suess, put the first microcomputer bulletin board system into operation. Ward made his software available to others and provided support to get them up and on line.

Further, Ward developed a program called MODEM for doing error-free automatic file transfers between systems. Over the years the program has been improved and expanded by a large number of RCPM operators. However, the basic protocol for transferring files remains the same and has come to be known as the Ward Christensen file transfer format. In fact, this format has also been adopted by many mini and large mainframe systems to allow transferring of files between micros, minis, and mainframes.

Ward was recently honored by a "Computer Hobbyist of the Year" award at the recent Trenton Computer Festival. This award was given not only for his CBBS work, but also for developing and putting into the public domain a large number of software development utility programs that have become classics in the CP/M world. His programs live on not only because of their great utility, but also because their source code is so beautifully written and commented that they serve as excellent models for those learning to write 8080/Z80 assembler code.

While I am giving credit to monumental contributors of software to the public domain, I must mention Rick Conn (see "ZCPR2," *Microsystems*, June 1983) who alone has contributed over a dozen volumes of software to SIG/M. These include such classics as ZCCPR-I and II, SYSLIB, and the HELP system.

Other individuals who deserve credit for the work they have done in building and improving the RCPM network are: Dave Hardy (who runs an RCPM system for RCPM system operators in addition to a

Editor's Page
continued . . .

standard RCPM), Keith Petersen (who has improved many of the modem utilities in the public domain and coordinates the work of others in this area), Bruce Ratoff (who operates the SIG/M BBS and has written some of the software used by RCPMers) and Kelly Smith and Jud Newell (who run three of the largest RCPM systems in current operation and have assisted many in setting up their RCPM systems). This list could go on and on.

The future of RCPM systems

The number of RCPM systems continues to grow at a consistent rate. However, there are fears that the coming increase in telephone rates, next year, will put a damper on RCPM activities. Ways are being sought by amateurs to cope with this problem. A few systems already have the ability to operate at 1200 baud, and there is no doubt that by this time next year virtually all RCPM systems will have this capability.

At meetings of RCPM sysops held in January at CP/M '83 and in April at the Trenton Computer Festival, this problem—and particularly how it affects communications between sysops—was discussed. Solutions to this problem are being investigated.

Further, several groups of amateur radio operators are currently developing a radio repeater network that allows transmission of data through the air. And another group is working on the use of satellites for data communications.

RCPM and modem communications software is being continually improved to provide more reliable and more powerful features. Thus there is no doubt that public domain software and the public domain software distribution systems such as the RCPMs, SIG/M and CPMUG are here to stay, despite the rapid growth of commercialism in microcomputing.

News & Views

by Sol Libes

Random rumors

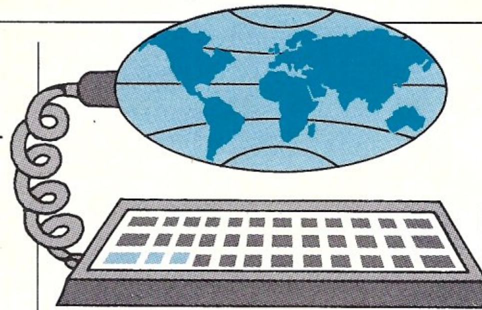
Digital Research is rumored working on a VMS-Like multiuser, multitasking disk operating system for the National 16032 chip. The DOS will not be related to CP/M. A first for DRI MS/DOS in the C language, it will attempt to bring out versions for the Z80 and 68000. . . . Intel is expected to supplement its ROM implementation of CP/M with an implementation of MS/DOS, while Digital Equipment Corp. is expected to implement its proprietary VMS operating system in ROM. . . . Altos Computer Systems is expected to release a new terminal with a built-in Trackball, using the NEC 7220 graphics controller chip. Integrated high-density software packages to support the terminals will be produced also.

Zilog to release Z800

Zilog has finally officially released the Z80 microprocessor, although samples will not be available until the fall and production is expected in the first quarter of '84. The Z800 is in effect a greatly enhanced Z80. . . . Zilog claims five times greater performance. The device incorporates circuitry that until now required external circuitry plus an expanded instruction set. It has a clock rate of 10-25 MHz (Z80 current maximum is 8 MHz).

The Z800 will include a memory manager (for up to 16MB in 64K blocks), DMA controller, counter/timers, serial I/O, 256-byte cache (program, data, program/data and local memory), interrupt controller and memory refresh logic.

The added functions include instructions for hardware multiply/divide, 16-bit arithmetic, 16-bit load, system/user calls (for multiuser/multitasking) and test/set (for multiprocessing). A floating-point math coprocessor (Z8070) was also announced. Four additional ad-



dressing modes have been added, plus four additional registers (includes a second stack pointer). The Z800 will come in four versions (prices are the 1000-piece price):

	I/O	Pins	Address Lines	Price
Z8108	8 bit	40	19	$40
Z8116	16 bit	40	19	48
Z8208	8 bit	64	24	60
Z8216	16 bit	64	24	72

The 8-bit I/O versions interface directly with the Z80 peripheral chips, while the 16-bit I/O versions work with the Z8000 family of devices and have about twice the throughput. The 40-pin devices will also lack some of the circuitry, such as serial I/O and DMA controller.

Public domain software news

The SIG/M subgroup of the Amateur Computer Group of New Jersey continues to release new public domain software at a high rate. This month they released 12 new volumes to bring their total up to 112.

Volumes 88 through 108 include SYSLIB (an integrated library of assembly language subroutines for use with Microsoft Basic) and ZCPR2 (an enhanced replacement for CP/M's CCP module; for a description of ZCPR2, see *Microsystems,* June 1983, pg. 90). Volume 109 is a complete word processor package. Volume 110 is a collection of dBASE-II and SuperCalc programs. Volume 111 contains several utilities for managing back-up files with hard-disk systems. And Volume 112 contains programs for use by RBBS and RCPM systems.

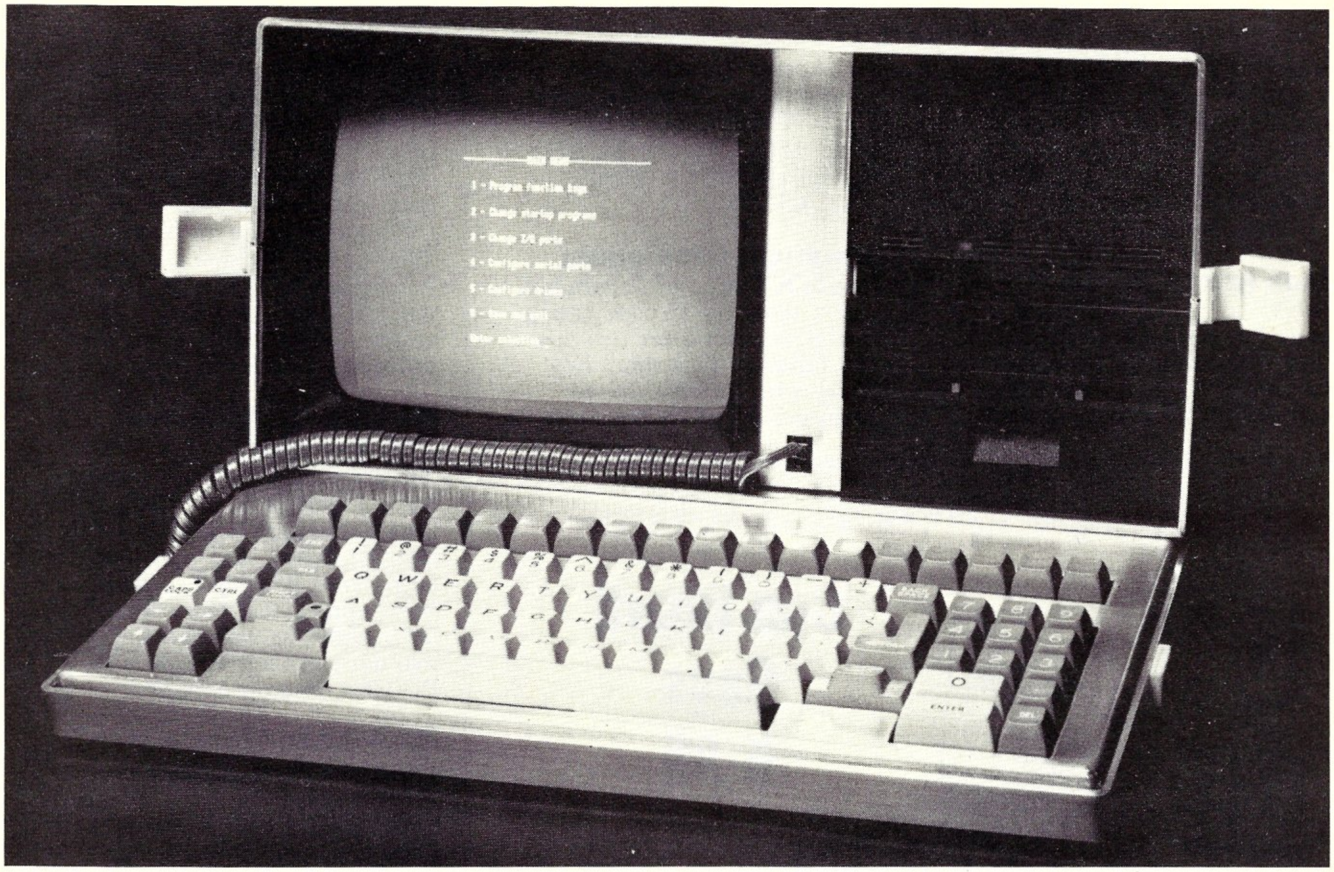
SIG/M has released a new edition of their printed catalog, which lists all their software up through volume 112. Cost is $2 domestic ($2.50 foreign). Write **SIG/M,** *Box 97, Iselin NJ 08830.*

The CPMUG group did not release any new software this month. However, they have raised the price for their software. Charges are now $13/8″ disk volume domestic ($18 foreign) and $17/5.25″ North Star and Apple disk volumes domestic ($21 foreign). Their printed catalog is $10 domestic/$15 foreign. Also, we have been notified that CPMUG can now be reached by telephone. Call (212) 860-0300, extension 343, and ask for either Marcia Coltun or Anna Jourdain. **CPMUG's** address is *1651 Third Avenue, NY NY 10028.*

UNIX News

American Bell (that new AT&T independent subsidiary?) has begun, quietly, to demo their new UNIX-based micros (using the 32-bit MAC-32 microprocessor) to select customers (mainly within the AT&T organization). These are all high-performance and high-priced systems using the Fortune Systems 32/16 as workstations. AB is also using components from Onyx and DEC in these systems. And AB is expected to use a version of Basic developed at Bell Labs that is compatible with Microsoft Basic. Other applications software packages, such as a DBMS, are expected and are rumored coming from outside suppliers.

This represents a major challenge to Microsoft's Xenix, licensed from AT&T (via Western Electric). With IBM rumored ready to select the Human Computing Resources (Toronto, Canada) version of UNIX for the PC and UniSoft doing well with their UniPlus, Microsoft is struggling to be competitive. Gossip has it that,



fearful of losing Radio Shack to another UNIX supplier, Microsoft made an all-out crash effort to transport Xenix to the Radio Shack Model 16. Working around the clock, they a 3-week deadline.

User group news

CBUG, a CBasic Compiler User Group, has been formed. They will publish a newsletter (12 issues/yr) and cover topics such as CBasic, Display Manager, Access Manager, news, tips and topics. Membership is $12/yr. Write: CBUG, c/o Al Dallas, 11669 Valerio St. #213, No. Hollywood, CA 91605; (213) 765-3957.

Mitsubishi tries again

Mitsubishi Corp., of Japan, will make a third try at penetrating the U.S. market after two previous failures. This time they will try with an 8086-based system using CP/M for single and MP/M for multiuser systems, and a proprietary local network. Mitsubishi's initial attempt was in 1979 with an 8080-based machine, of which they sold 200. In 1980 they tried again, using an AMD-2900 based machine designed to compete with the IBM System/38. They sold under 30 systems. Both computers used proprietary operating systems.

Zilog sues NEC over Z80 patent

Zilog Inc. has filed suit against NEC Electronics, charging that they unlawfully copied the Z80 microprocessor and infringed on patents. They allege that NEC copied the Z80 and related chips, as well as Zilog manuals, in their PD780 and PD780-1 microprocessors.

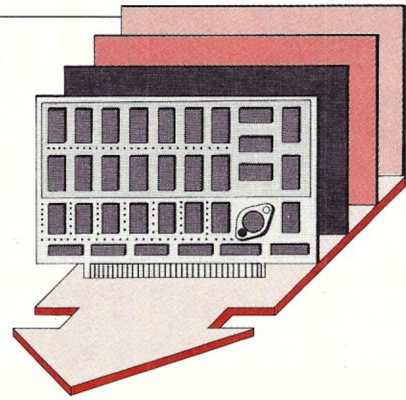
IEEE LAN Standard nearing completion

The proposed IEEE 802.3 standard for Local Area Networking has moved a step nearer adoption with the endorsement by 13 key hardware and software vendors. This proposed standard has been in the works longer than any other of the IEEE computer standards and has stirred the greatest controversy. The proposed standard has moved to the next level of approval by the IEEE microcomputer standards committee. From there it goes to the computer standards committee and IEEE standards board for final adoption. If everything goes well, we may see it officially adopted by year-end.

The standard conforms very closely to the Ethernet LAN as proposed by DEC, Intel, and Xerox. This LAN standard is expected to be used by most high-performance LAN systems. It should be noted that IBM and AT&T are known to be working on their own LAN systems, which do not conform to the proposed standard. Furthermore, the overwhelming majority of LAN systems already installed are of the low- and medium-performance type, where nothing approaching standards exists.

The S-100 Bus

by David Hardy



As mentioned in the last "S-100 Bus," this month's column will discuss the "special bus operations" of the IEEE-696 standard. Although these operations are clearly described in the IEEE-696 standard, their importance in S-100 systems that use multiple processors and TMA makes them worth special attention.

The real importance of these special operations becomes apparent when one realizes that "special bus operations" include almost all DMA-like operations (that is, all TMA) that are performed on the bus, and that they also include the arbitration protocol used to determine the priority and access granted to boards that simultaneously request TMA on the bus.

As you may recall, TMA, or Temporary Master Access, is the term chosen by the authors of IEEE-696 to describe the procedure by which a board connected to the S-100 bus may temporarily "take over" the S-100 bus to perform such things as Direct Memory Access or other bus cycles. It replaces the older, less accurate term DMA.

In case you think that your system has no need of such a sophisticated method of bus access, bear in mind that many S-100 disk controller boards must perform some sort of TMA in order to transfer data between themselves and system RAM. (In most S-100 systems, this is the only TMA function performed.) Because it is so necessary for disk I/O, this is probably the most important immediate reason for TMA.

Because the IEEE-696 standard was designed with the future in mind, it provides what is really another, "higher" level of temporary master access, in which provision has been made via a bus arbitration procedure to allow up to 16 separate TMA devices to exist simultaneously, without conflict, on an S-100 bus. The importance of this feature is rather obscure if one thinks in terms of multiple disk controller boards, but become clear when one realizes that TMA can also be performed by other processor boards. In other words, using the bus arbitration protocol described in the IEEE-696 standard, it is possible to have up to *16 processors* simultaneously existing on the same S-100 bus. This is a major feature of the IEEE-696 standard, and something that was sorely lacking in the old S-100 bus.

Because of the complex nature of the S-100 bus, transfer of bus control from the permanent master to a temporary master board is a rather complicated procedure. The bus Transfer State Diagram shown here (and in IEEE-696 section 2.8) shows the procedure that must be used by a TMA-requesting board to obtain temporary control over the bus.

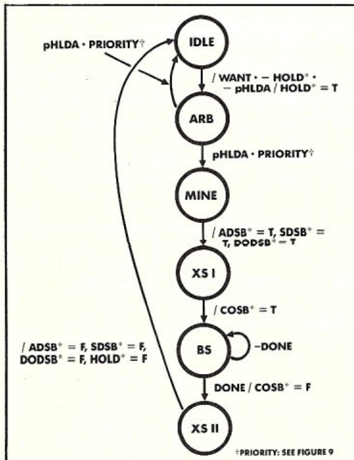


Figure 7. *Bus transfer state diagram.*

The diagram shows the six states that occur during the bus control transfer procedure. Briefly, they are:

IDLE—the time before the board wants the bus, or while it is waiting for it.

ARBitration—if the board must contest with other TMA-requesting boards; else no arbitration is needed.

MINE—when bus access is granted to the requesting board.

TS I—the beginning part of the bus transfer sequence, where the permanent bus master's address, status, and data output drivers are disabled, and the temporary master's control output drivers are enabled. Then the permanent master's control drivers are disabled, and the temporary master's address, status, and data output drivers are enabled. Note that during this process, there is a short time where the control output lines are actually driven by the permanent and the temporary master as the same time! This is done to prevent any "spurious" signals from being injected into the control lines during the bus transfer. Both boards must drive the control output lines in identical prearranged states during this time.

BS—This is the state in which the temporary master conducts its bus cycles (i.e., this is when the temporary master does whatever it is supposed to do, like disk transfers or whatever).

TS II—This is the exact reverse of TS I. When this state has ended, the permanent master once again has complete control of the bus, and the temporary master is, once again, in the IDLE state.

All of the above states are fairly straightforward, except

HOW TO CONTROL YOUR CP/M™ MICRO–AT ONLY $3. PER COMMAND.

A Remarkable Program For CP/M Users.

Of course, CP/M is a wonderful operating system. That's why so much serious business software has been created for it.

BUT, CP/M is not easy to work with. That's why you need to take the **POWER!** trip.

POWER! is a super-power-packed, user-friendly program that lets you take immediate and complete control of CP/M. And at a cost of only $3. per command, it's the software buy of the year.

Over 55 Housekeeping Utilities.

POWER! is over 55 prompted, user-friendly CP/M utility programs all rolled into one 15k package. It takes care of all of these frustrations and more:

—BDOS errors? POWER! ends BDOS errors and gives you a way out.

—Accidentally erased a file? If you accidentally erase a program or disk file, **POWER!** restores the erased files.

—Can't remember file names? **POWER!** assigns a number to each file on your disk. So, to copy files from disk to disk, you don't have to fiddle with PIP anymore. You just pick the file from a numbered menu and **POWER!** copies it for you. No more typing errors! **POWER!** also marks original files and their copies for you; and you can compare files to find identical copies regardless of name.

—Lose data on a glitched disk? If a glitched disk makes it impossible to call up a long word processing text, **POWER!** can fix the glitch. This means you may have to retype only a couple of sentences instead of losing 20 pages of text.

—Trouble with "bargain" disks? **POWER!'s** disk testing function gathers any bad sectors of the disk into a special file so that CP/M thinks those parts of the disk are already used and never attempts to write to them. The rest of the disk is then safe to use.

—CP/M scrolls too fast through text files? **POWER!** spools through files for you, page by page, file by file, or line by line with instant halt by touching the space bar.

—Need to reorganize files? **POWER!** sorts and formats the directory in 4 different ways. And you can easily copy or move files from user area to user area. **POWER!** creates 32 user areas instead of CP/M's 16.

—Need to change memory? **POWER!** searches, displays and lets you change memory wherever you want. You can even automatically run software anywhere in memory. And you can inter-mix your search with as many wild card jokers as you need to find, for instance, all occurrences of "Sam Jones" and "Sid James" just by typing "S??J??". And **POWER!** also lets you read or write to any sector or track very simply.

—Changing disks? You can forget the ubiquitous Control C to change disks. **POWER!** can do it for you automatically. And **POWER!** doesn't require a system disk in any drive, so Drive A is open for use, when **POWER!** is in control of CP/M.

—Afraid of HEX numbers? POWER! automatically converts Hex to Decimal, Binary or ASCII.

Special Password Protection, Too.

POWER! now includes a special program that lets you lock sensitive files, so that only you can access them. Without the secret PASSWORD which you can create and change at will, no prying eyes will ever know your secret file even exists. A great way to protect financial or scientific data from unauthorized eyes. Just this single program alone would be worth the price of **POWER!**, but there are over 55 more just as valuable programs in this power-packed-package.

At $169., It's A Bargain.

Space doesn't permit describing all the wonderful ways **POWER!** can put you in complete control of your CP/M micro. But see for yourself. There's a Money Back Guarantee. At the low price of $169., each powerful command costs you less than $3. A true bargain!

InfoWorld
Software Report Card
Power Version 2.55

	Poor	Fair	Good	Excellent
Performance	☐	☐	☐	☑
Documentation	☐	☐	☑	☐
Ease of Use	☐	☐	☐	☑
Error Handling	☐	☐	☐	☑

© 1982 by Popular Computing, Inc. A subsidiary of CW Communications Inc., Framingham MA. Reprinted from InfoWorld.

POWER! Is Better Than Ever!

Eventhough "InforWorld", "Microsystems" and "Interface Age" call **POWER!** great, we have improved **POWER!**—including a completely rewritten 120-page easy-to-read documentation. (Previous purchasers of **POWER!** may exchange their original disk for an updated version with the new commands and a brand new manual—for only $35.)

Take The POWER! Trip Today!

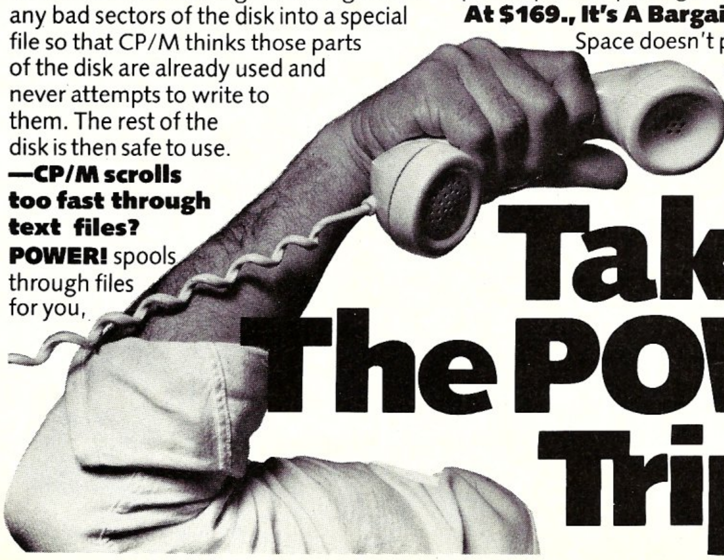
POWER! will operate in any standard CP/M or MP/M system, including CP/M-86, IBM PC, Apple (Z80 card), Osborne, Kaypro, HP, TeleVideo, TRS-80 conversions, S100's including NorthStar, Vector, Morrow, CompuPro, etc. Up and running immediately, no configuration necessary—for hard disks and floppies.

At only $3. per command, you can afford to **Take the POWER! Trip.** Call or send in your order today.

NOW AVAILABLE FOR MS-DOS, TOO.

ONLY $169. Money Back Guarantee.
Charge & COD Orders Welcome.

TOLL FREE (800) 428-7825 Ext.96AB
IN CA: (800) 428-7824 Ext.96AB
DEALERS AND OEM'S (415) 567-1634 Ext.96AB



Take The POWER!™ Trip.

COMPUTING!
2519**AB** Greenwich San Francisco, CA 94123

TOLL FREE (800) 428-7825 Ext.96AB
IN CA: (800) 428-7824 Ext.96AB
DEALERS AND OEM'S (415) 567-1634 Ext.96AB

ONLY $169. Calif. add 6½% sales tax
☐ **CP/M $169.** ☐ **CP/M-86 $169.** ☐ **MP/M $249.**

Card No. ____________
Exp. Date ____________
Computer ____________
Your Name ____________
Company Name ____________
Address ____________
City/State/Zip ____________

CIRCLE 14 ON READER SERVICE CARD

Do you have the the entire S-100

We do.

And now you can too: The most powerful multi-processing CPU board you can buy. The world's first 256K static RAM memory board. And the most sophisticated hard disk controller board available.

Run 8-bit CP/M 80* and 16-bit CP/M 86* simultaneously in a single or multi-user environment. $895.

Now you can upgrade to 16-bit without forsaking your investment in 8-bit software. Octagon's plug-and-play CPU Board 8/16™ is a complete computer-on-a-board with all this:

- Dual processors: 4MHz NSC-800 (executes full Z80* instruction set) and 8MHz 8088.
- Intel 8272 floppy disk controller with 24-bit DMA. Runs up to four 5¼" or 8" floppies in any combination at the same time.
- 8K PROM monitor. Boots operating system and also contains all the disk interface software you need for almost any operating system. Plus many useful debugging features, such as memory test, memory dump and memory search.
- Two serial ports (baud rate software selectable up to 19.2 Kbauds.)
- Interrupt controller with eight vectored interrupts.
- Fixed frequency real-time clock interrupt for multi-user dispatching.
- IEEE-696 S-100 compatible.

Add the optional 8087 math co-processor for just $300.

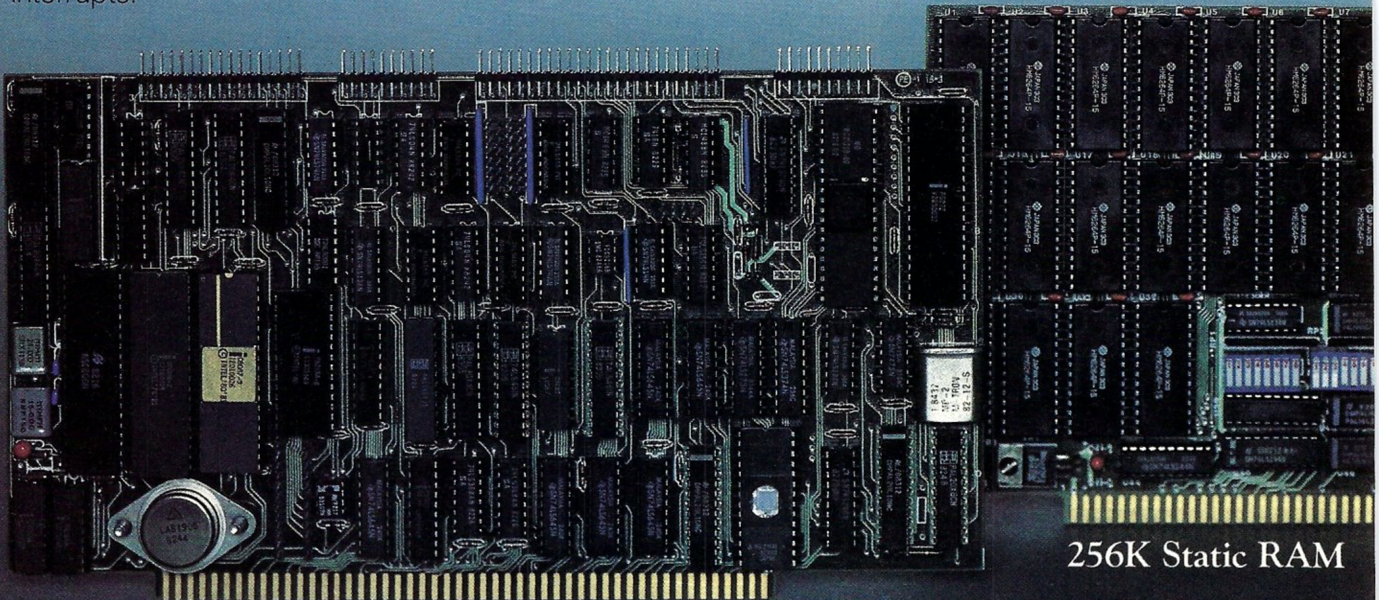
And save $155 when you buy the operating system from Octagon, whether you take Concurrent CP/M 86 at $195, CP/M 86 at $150 or MP/M 86* at $495.

CPU Board 8/16 is completely ready to run with no additional software. And both operating systems automatically include your CP/M 80 emulator, so you can run any 8-bit CP/M program—as well as auto-select bios that let you transfer files back and forth between 5¼" and 8" floppies.

World's first 256K Static RAM Memory Board. $1850

Memory may be configured as either four totally independent 64K software-selectable blocks or as a single 256K block which responds to IEEE-696 extended addressing lines. Will accept either 8-bit or 16-bit bus requests.

And each 64K block can be one of 256 addresses—so theoretically you could address



CPU Board 8/16

256K Static RAM

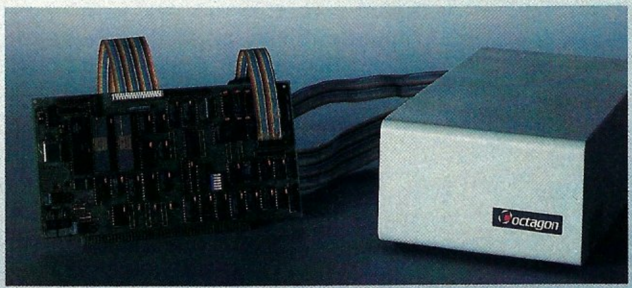
guts to take on micro industry?

up to 16MB of these boards in your system.

A partially populated 128K board with the same characteristics is just $1095.

Sophisticated, reliable Hard Disk Controller. $550.

Designed around LSI technology for greater reliability. With its sophisticated Western Digital



1010 hard disk controller chip to control up to four 5¼″ drives, this unique board gives you automatic seek, automatic retries after error; comes with both CP/M 80 and CP/M 86 bios; and includes two serial ports and one Centronics-compatible parallel port.

Using the WD 1014, it detects and corrects single-bit errors and detects double-bit errors.

ECC generation and correction on data fields is user selectable.

And its phase lock loop lets you forget about critical timing adjustments. In fact, you never need to adjust it at all. There's no pot.

Octagon's Hard Disk Controller Board also gives you data rates up to 5MBits/sec., and CRC generation and verification on ID fields. Complete subsystem including board and 19.2 MB hard disk packaged with power supply: $2195.

1-408-262-7777

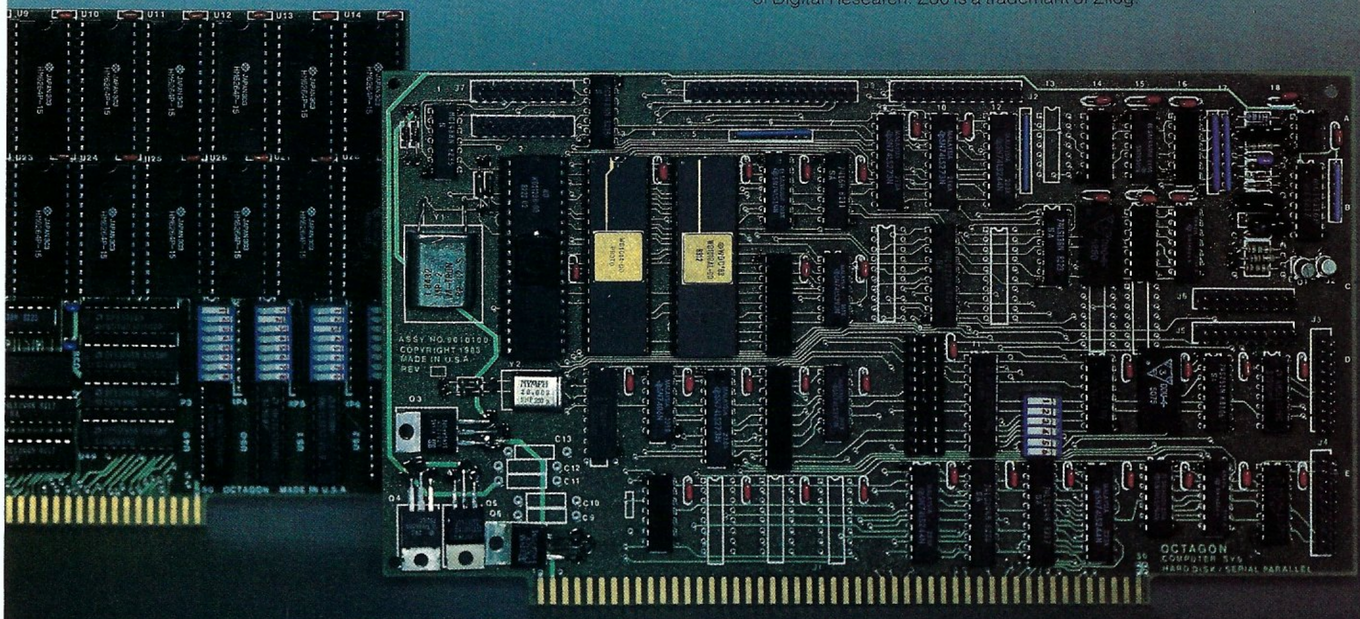
Call us now for more details. And take advantage of these introductory offers: Buy CPU Board 8/16 plus the Memory Board and get Concurrent CP/M 86 or CP/M 86 free. (Includes CP/M 80 emulator.) Or buy all three boards and get MP/M *plus* new Personal BASIC from Digital Research free.

You'll never need any other S-100 boards. It takes guts to say that, but try us out and you'll know: Octagon has what it takes.

octagon

2960 North First St., San Jose, CA 95134 408-262-7777

*CP/M 80, CP/M 86, MP/M 86 and Personal BASIC are trademarks of Digital Research. Z80 is a trademark of Zilog.



Hard Disk Controller

CIRCLE 189 ON READER SERVICE CARD

the ARBitration state, which was discussed in the last "S-100 Bus" (May 1983).

The complete bus transfer protocol (including bus transfer timing relationships) is a bit too involved to be completely covered here. However, this should give a general idea of what happens during TMA, and of the importance of TMA in the IEEE-696 bus. TMA is defined completely in the IEEE-696 standard, which should be available soon from the IEEE.

Reader feedback and disk drive woes

I have received a number of letters from readers concerning problems with 8″ floppy disk drives. Most of the problems involve slow step rates, but there are a few exceptions.

Although most new floppy disk controller cards can step the heads of their floppies at 3 ms, many of the older 8″ floppy drives can only step at a maximum of 8 ms or 10 ms. This is particularly true of many single-sided drives. Usually, the controller board's step rate can be adjusted for these slower drives, but not all manufacturers mention the procedure to do this. In addition, many people just don't know what the step rate of their drives actually is. Be sure to check before you buy, and make sure that your controller board and your drives will work together.

Now for the exceptions. At least one popular DMA controller board now being marketed is unable to work with most double-sided 8″ floppy drives because of a write-gate timing problem. Currently, it will work with Shugart 850s, but not much else. The problem will probably be fixed by the time this is printed (via an EPROM update), but be sure to check that your drives are compatible before you buy.

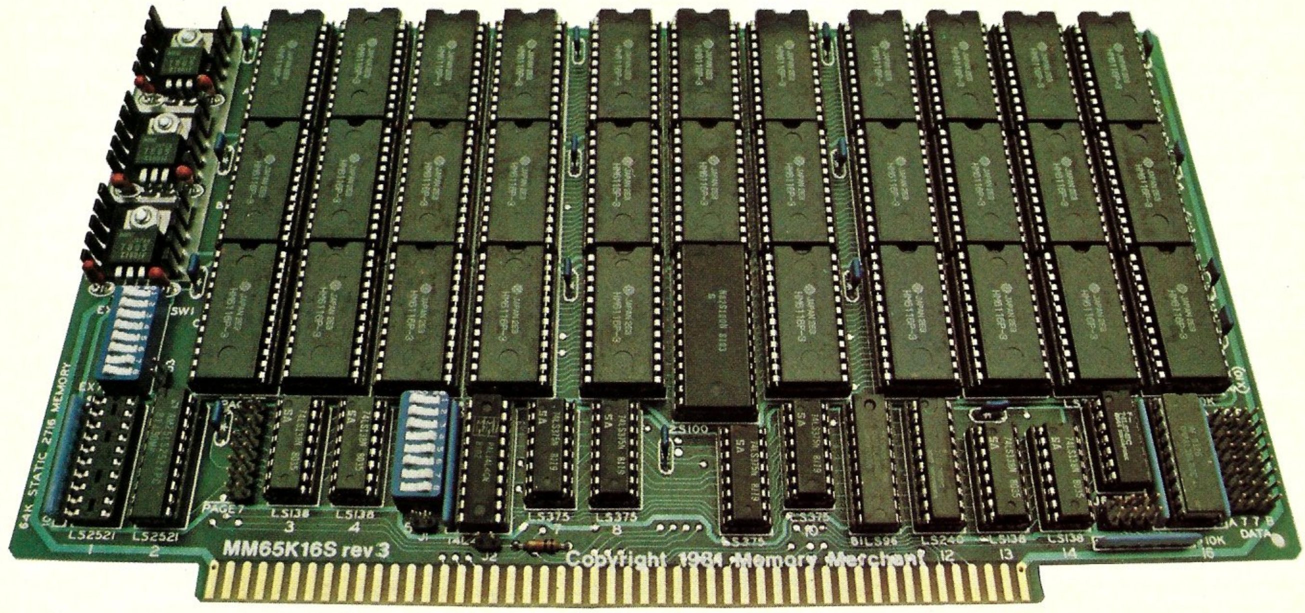
Another *very* popular controller board has similar problems with various brands of floppy drives. This trouble can be cured with just a few lines of code in the BIOS, but will cause infrequent BDOS errors during heavy disk I/O.

I'm not mentioning any manufacturers' names because these problems occur in more than one brand (they are "generic" problems); however, you may see a word or two about these troubles in a future "S-100 Bus."

Future topics

Dysan Corp. has introduced a new type of alignment disk, called the "Digital Diagnostic Diskette" (DDD), which takes a new approach to the (once analog) domain of floppy disk drive alignment. With the old alignment methods, you needed an expensive oscilloscope and a special (equally expensive) analog alignment disk. You also needed a skilled technician to figure out how to use them. With the DDD disk (which costs about $40), you will still

64K STATIC RAM MEMORY



S-100 STATIC MEMORY BREAKTHROUGH

Finally, you can buy state-of-the-art S-100/IEEE 696 static memory for your computer at an unprecedented savings.

Memory Merchant's memory boards provide the advanced features, quality and reliability you need for the kind of operational performance demanded by new high-speed processors.

Completely Assembled.

These memory boards are not kits, nor skeletons — but top-quality, high-performance memories that are shipped to you completely assembled, burned-in, socketed, tested and insured with one of the industry's best warranties.

Superior Design & Quality.

Memory Merchant's boards are created by a designer, well known for his proven ability in advanced, cost-efficient memory design. Innovative circuitry provides you with highly desired features and incredible versatility.

Only first-quality components are used throughout, and each board is rigorously tested to assure perfect and dependable performance.

No Risk Trial.

We are so convinced that you will be absolutely delighted with our boards that we extend a no-risk trial offer. After purchasing one of our boards, you may return it (intact) for any reason within 15 days after shipment and we will refund the purchase price (less shipping).

NEW S-100 PRODUCTS COMING SOON:

* DUAL 8/16 BIT CPU BOARD
* 128K 8/16 BIT STATIC RAM
* 256K 8/16 BIT DYNAMIC RAM

$629.

48K PARTIALLY POPULATED $519.
32K PARTIALLY POPULATED $409.

64K RAM, MODEL MM65K16S

- 64K × 8-bit
- Speed in excess of 6 MHz
- Uses 150ns 16K (2K × 8) static RAMS
- Ultra-low power (435 Ma. max. — loaded with 64K)
- Bank Select and Extended Addressing
- A 2K window which can be placed anywhere in the 64K memory map
- Four independently addressable 16K blocks organized as:
  - — Two independent 32K banks **or**
  - — One 64K Extended Address Page **or**
  - — One 48K and one 16K bank for use in MP/M[1] (option)
- Each 32K bank responds independently to phantom
- 2716 (5V) EPROMS may replace any or all of the RAM
- Field-proven operation in CROMEMCO CROMIX* and CDOS*.
- Compatible with latest IEEE 696 systems such as Northstar, CompuPro, Morrow, IMS, IMSAI front panel, Altair and many others.

OEM and DEALER inquiries invited.

Memory Merchant™

14666 Doolittle Drive
San Leandro, CA 94577
(415) 483-1008

CIRCLE 98 ON READER SERVICE CARD

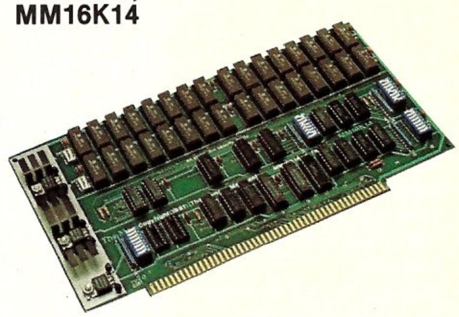
FULL TWO-YEAR WARRANTY.

The reliability of our boards, through quality-controlled production and proven performance, has enabled us to extend our warranty to a full two years. That's standard with us, not an option. This includes a 6-month exchange program for defective units.

Shipped direct from stock.

All Memory Merchant's boards are shipped direct from stock, normally within 48 hours of receipt of your order. Call us at (415) 483-1008 and we may be able to ship the same day.

16K RAM, Model MM16K14



16K STATIC RAM $169.

16K × 8 Bit
Bank Select & Extended Addressing
Four independently addressable 4K blocks
One 4K segment equipped with 1K windows
Uses field-proven 2114 (1K × 4) RAMS
Low Power (less than 1.2 Amps)
Runs on any S-100 8080, 4 MHz Z-80 or 5 MHz 8085 system.

Prices, terms, specifications subject to change without notice.

*Cromix and CDOS are trademarks of CROMEMCO.
[1] MP/M is a trademark of Digital Research

S-100 Bus continued. . .

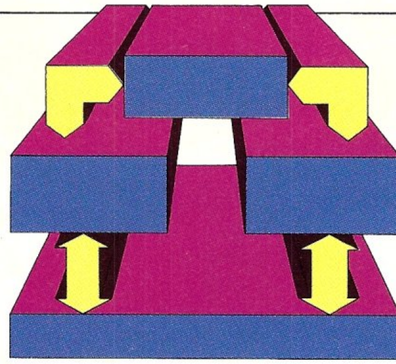
need some technical skill, but the alignment can now be done without the scope, and on your own (S-100) computer, with just a little kludging. I am working on my own version now (for a DJ2D), but if you can't wait, there are already external alignment units on the market (check with Dysan for details). A complete alignment check with the DDD takes about 3 minutes (it used to take about an hour).

Next time: another dose of IEEE-696 (the Temporary Master Interface), more reader feedback, and some interesting facts about floppy disk drives.

This column is intended as a forum on S-100 topics. I encourage readers to send in questions about the S-100 bus, which I will attempt to answer. Please write to Dave Hardy, 736 Notre Dame, Grosse Pointe, MI 48230.

The UNIX File

by Ian F. Darwin



The UNIX File is scheduled to appear every other month and will spotlight important aspects of UNIX. If you have any questions, send them in and I will attempt to answer them. Please write to Ian F. Darwin, University of Toronto Computing Services, 10 King's College Road, Room 4306, Toronto, Canada M5S 1A1.

On the influence of UNIX, and some books on UNIX

This column will explore the ways in which UNIX has influenced an entire generation of computer designers and users, many of whom don't even know about UNIX. Then I'll look at some of the books that talk about UNIX from various points of view, and claim to go beyond what the standard Bell Labs manuals offer.

Under the influence . . .

Imitation has been called the sincerest form of flattery. If this be true, then a large part of this little industry is rushing forward to prove itself adept at flattery. Everywhere people are scurrying to copy the ideas of UNIX into their favorite computer system. Thompson, Ritchie, and the gang have reason to be proud, for they have built a system whose influence has been and will be felt throughout the computer field. Here's looking at UNIX and some of the influences it has had on other computer systems.

One of the most obvious influences of UNIX is the growing use of the C language. C is the "native language" of UNIX and has been made widely available. There are numerous CP/M implementations, plus a number for the Intel 8086/8088 processor as used in the IBM PC and elsewhere, as well as for many other computers. Several of these were reviewed in *Microsystems* in November/December 1981 and November/December 1982. Many features of UNIX, such as command line argument passing and I/O redirection with the '<' and '>' operators, have been made available wherever C goes.

Some of the more useful utilities, or 'tools' have been exported from UNIX to other systems. The *Software Tools* books by Kernighan and Plauger provide commentary and source code for many tools which are like those of UNIX. This led to the Software Tools User Group (still active) and an article "A Virtual Operating System" in *CACM* (September 1980, Vol. 23, No. 9. p. 495). The Virtual Operating System and tools can be had for S-100 micros—see the review "Small-VOS and Small Tools" in the January '83 *Microsystems.* That issue also describes two packages which give CP/M some UNIX function—Microshell and UNICA. CP/M Plus has a number of features that have been present in UNIX (and other systems) for years, such as user-definable search list, date- and time-stamping of files, and a limited imitation of UNIX's facility for redirection of input/output files. CP/M-68K, the new version of CP/M for the MC68000, has a C compiler and related utilities (L068, SIZE68, NM68) that are clearly patterned after the corresponding UNIX tools (**ld, size, nm**). For CP/M 2.0 users, there is a public domain version of the UNIX text formatter **nroff** (called 'nro' because it is a subset) which I got from the local CP/M Bulletin Board System.

For digital Equipment PDP-11 mini- and microcomputers, UNIX has long been the operating system of choice for most applications. But some programs—and some organizations—remain wedded to DEC operating systems such as RT-11 and RSX-11. For both of these, a number of C and RATFOR compilers can be had, as well as various forms of the software tools, both commercially and through the DECUS user group. A UNIX-like shell was written in Whitesmiths' C for RT-11; this program implements pipes by using temporary files. For RSX-11M, Oregon Software has a series of "SourceTools" that appear to be crafted after the design of the UNIX **diff, make** and Source Code Control System tools.

Complete UNIX-like systems have been implemented and are being used. MARC is an 8080/Z80 UNIX-like system that boots up under CP/M, but replaces all of CP/M except the BIOS. Idris and Coherent are imitations of version 6 and version 7 UNIX respectively. TUNIS, described in the following section, is similar to version 7. And a great number of systems are derived from Bell Labs UNIX; a supplier will change a few pieces and adopt a name with an-IX ending to let you know that the system is derived from UNIX. Some of these systems include Spectrix, VENIX, and XENIX—but not all systems whose names end in -IX are derived from Bell UNIX.

More recently, Microsoft announced version 2.0 of MS-DOS (sold by IBM for their PC as PC-DOS). This includes a number of features copied directly from UNIX, in-

cluding the "redirection" operators ('<' and '>'), pipes (apparently implemented by use of temporary files) and even a tree-structured directory.

Many programmers and systems designers have adopted the "software tools" approach pioneered by UNIX, and many, many computer users will benefit from the UNIX approach to computer access. Perhaps equally important are some of the system's overall design strategies, such as the elimination of the concept of the 'record' from the OS, the provision of a minimal, efficient kernel free from the "creeping featuritis" that plagues systems such as MVS, and the simple but elegant file system design. It will take longer for these to become incorporated into the operating system development process, but I predict that this too shall come to pass.

By the book . . .
In addition to the standard Bell Labs manuals that come with UNIX, and the manual-like *C Programming Standards & Guidelines* from Plum Hall, I know of seven books currently in print describing the system, and several more coming. Here's a look at a few of them.

Rebecca Thomas and Jean Yates are sometimes credited with the first non-Bell-Labs book on UNIX, *A User Guide to the UNIX system* (reviewed in *Microsystems*, May 1983, page 136). In fact this honour belongs to Richard Gauthier, whose *Using the UNIX System* was first released in June of 1981. Most of the books I am discussing here are aimed at the novice. Like the others, Gauthier gives a description of the keyboard conventions, the command language, the line editor, file usage, and so on. The Gauthier book, *Using the UNIX System*, also includes coverage of a topic which most overlook, that is, the system administrator's role. For a single-user system, of course, the user is the system administrator, and for a large multiuser timesharing system, the system administrator is probably someone whose knowledge of UNIX goes beyond the information gleaned from reading this chapter. But it's one of the few concise, organised presentations of the system administrator's operations to be found. The book was "written for people with some knowledge of computer sciences, but with no specific knowledge of the UNIX system." One aesthetic weakness is that the author, who typeset the book himself using Knuth's TeX (nothing like DRI's TEX) did not seem to understand typography as distinct from computer printouts; all the examples in the book have little pointers (made of a dash and a greater-than sign). Use of bold face to distinguish user input from computer output would have produced a better looking book. As well as being more in line with conventions used in other UNIX documentation, this would also provide a more accurate portrayal of interactive prompting. A valuable por-

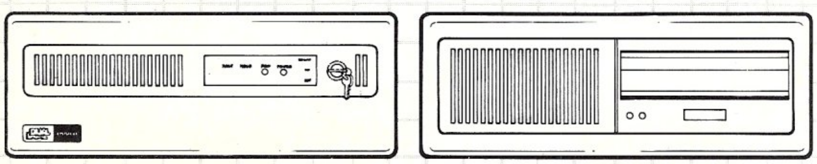
tion for the new user is a series of summaries and exercises to reinforce your understanding of particular skills.

One of the people who helped develop version 7 UNIX has written a book called *The UNIX System*. Beneath this simple title lies a very well-written exposition of the system's basic concepts and usage. This is a programmer's guide to UNIX; in the introduction S. R. Bourne states that "This book covers many aspects of the UNIX system from the user's point of view. It assumes that the reader has some familiarity with modern computing technology." There is sufficient information to enable someone with the specified background to learn the correct use of UNIX for text entry, writing both applications and 'systems' programs in the C language, use of the utilities and the command language, and text formatting with **nroff** and the '-ms' macro package. Unlike Thomas & Yates and unlike the Lomutos, Bourne describes *both* the line editor **ed** and the screen editor **vi,** which takes advantage of most video terminals using TERMCAP (see the March 1983 installment of this column). Bourne should know the command language very well—he wrote it. And he explains it very well here, with plenty of examples. This is the only one of the books I've seen so far to delve into systems programming in C, an area in which Bourne's familiarity breeds comfort, rather than contempt. All chapters of the book have review sections where appropriate.

The ultimate examples in the book include description and source for a C language cross-reference and a tennis ladder system which maintains a tennis ladder for the staff at a real computing centre. The C cross-referencer program uses the UNIX tools **lex, make, sort** and the C language. The tennis ladder uses **at, awk, cat, nroff, sort** and **uniq** working together so that one can see how a real-life system is constructed out of the UNIX tools. The appendices include descriptions of the most common commands, system calls, and summaries of the subcommands or 'requests' for **adb, ed, sh, troff** and **vi.** Also included are the source for a small formatter macro package (the one actually used in production of the book) and a summary of the '-ms' macro requests. As an aside, this book is current through UNIX System V as well as Berkeley 4.1BSD. With the resources of Bell Labs behind him, it's not surprising that Bourne would come up with a book that is both well-written and professionally typeset. S. R. Bourne's *The UNIX System* is highly recommended for anyone with computer experience who wishes to learn about UNIX.

For those without previous computer experience, and who will be using the system primarily for text processing or for running application programs, an interesting book is *A UNIX Primer* by Ann Lomuto and Nico Lomuto. This work is in the "Prentice-Hall Software Series" and carries the imprimatur of series editor Brian W. Kernighan, one of the original UNIX people in Bell Labs in the early days of the system, and one of the major publicists for the operating system and for the C language. The Lomutos have composed a book that seems to be quite well organised for the beginner who has no computer expertise. *A UNIX Primer* leads you through use of the system from logging in to text editing and formatting, and running other programs. There is no information on programming in C or other languages; such material would be outside the scope of this book. Little cartoon drawings of the key ideas adorn parts of the book. The examples used to illustrate text formatting are delightful nonsense, including memos which document the rise—and fall—of one J. J. Bullwinkle, section manager of a hypothetical corporation. In contrast with some other books mentioned here, the typography is well done, including use of typewriter (monospace) font where appropriate; the book looks professionally typeset.

I would only quibble with the Lomutos' order of presentation of material in a few places. First, the authors present the use of 'remembered filename' in the **ed** editor in the chapter on "Efficient Editing" rather than with the introduction to editing. To my mind this feature is so useful that it belongs in the beginning chapter on editing. Similarly, the information on the 'w' command (which saves your file on disk) is presented much later than the 'q' command, which (at least on very early versions of UNIX) would throw away the changes you had made to a file. More importantly, I would prefer a more sympathetic—and more informative—presentation on use of public 'formatter macro' packages such as 'mm' or 'ms', as well as use of



UNIX

the '-m' command line argument to the **nroff** text formatter. And of course there are a few technicalities one could quibble with. But one thing I really like about the book is its abundance of both *summaries* and *exercises*. Taken together, the summaries and exercises are very helpful in reinforcing one's learning of the material. Overall, I rate this book highly for novices, for those with no previous exposure to computers.

And finally there is a book that I can recommend for those who want to learn about the way UNIX operates internally. A previous book on the subject, *A Commentary on the UNIX Operating System* by John Lions of the University of New South Wales, cost $20,000, and you got a UNIX source license for version 6 with it. (Actually, I think it's the other way around—you bought the license and got the book.) It is distributed only by Western Electric to holders of V6 source licenses for UNIX. The "NSW Commentary" describes the operation of version 6 UNIX, which is not used that much anymore, and you probably will never see a copy anyway. So it's refreshing to see a major publishing house release a book that consolidates a lot of the previously published information on UNIX internals.

Ric Holt's *Concurrent Euclid, the UNIX system, and TUNIS* is a University-level textbook for use in operating systems courses, but it is accessible to anybody with knowledge of any modern block-structured language such as Algol, Pascal, or C. Despite some minor typographical problems, the book is entertaining and quite readable. Professor Ric Holt first discusses the problems of concurrency, which are part of any real operating system, then discusses Concurrent Euclid (ConEuc, or CE). CE has been in use for several years; it is a Pascal-like language with features for systems programming and concurrency as well as for program verification. One major project done in CE has been TUNIS, a UNIX-

The Compare II Story

as told by
L.L. "Bill" Packer
Director of Engineering
Solution Technology, Inc.

About a year ago, our company had a problem we solved by writing a utility program called COMPARE. I was asked to tell the story about how COMPARE evolved and why you might want to have your own copy. This engineering story spans about one year.

Early in 1982, Solution Technology, received a contract to implement a hard disk file server and database system using IEE-488 communications links and Digital Research's MPM-II operating system. Here's the kicker, the project had to be done between the first of February and the middle of April. In all, I had five programmers working on seven different computer systems; some with hard disks and some with floppies. With so many machines, programs and archive floppies involved, plus an around - the - clock work schedule, all normal source control methods were a joke. Our two biggest problems were; what was broken when a program used to work, and who changed what when two programmers "fixed" something? I tried a number of different conventional control systems that only held the problem at bay. In the end, the control problem became so acute that I assigned a couple of my best software engineers to come up with a way to compare two source files. Well, they designed a traditional file compare program called, naturally, COMPARE, and that program made our software control problem at least manageable. COMPARE 1.1 was effective but slow, so we ran it mostly on the hard disks.

COMPARE 1.1 was a useful program, but it needed work. First we had to remove the file restrictions so differences between files could be ANY number of lines long. Second, we needed the program to be able to compare the two input files in something less than an eternity. So back into the lab we went to see what software technology could be applied. We added a pinch of database philosophy and a dash of communications technique, and the result was COMPARE 1.2. . . a fast, line oriented file compare program which was very useful for programs and data files.

A short time later we were updating a technical manual for another contract and had to put change bars in each new issue to tell the end user where the changes were. Since text updates had been made by a number of people at different times, we attempted to use COMPARE 1.2 to locate all the places that had been changed. While somewhat successful, we still had to draw the change bars by hand. This was a huge waste of time because the process had to be repeated for every new issue. Enough! Time is money, so back into the lab. First to add word by word scanning and second, to be able to generate change bars automatically. Again success! We created COMPARE 1.3.

We have been selling COMPARE 1.3 for nine months now with hundreds of sales, excellent reviews, and customers telling us that we have a fine text analyzer. We've been pleasantly surprised at how many writers and lawyers have been using this package successfully. Even so, some said, "Why don't you have a version for the IBM-PC and CP/M-86?"

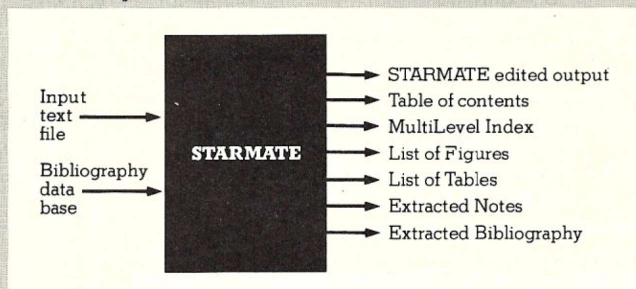
We listened. During the first months of 1983, we returned to the lab and added the most asked for features to the CP/M-80 version then did the code conversions for CP/M-86 and PC-DOS. What resulted was COMPARE II. In COMPARE II we give the user the ability to customize for his own default file name extensions, internal editor file formats, strings for change bars, strings, for difference highlight on and off, printer width, output to console, full document side by side listing, definition of comments (to ignore) and much, much more.

Order COMPARE II directly from your local dealer or Solution Technology, 1499 W. Palmetto Park Road, Suite 218, Boca Raton, Florida 33432; (305) 368-6226. Single unit price is $145.00.

Get more than WordIndex for less $!

STARMATE

Users get high speed document finishing for WordStar with STARMATE under PC-DOS, CP/M-86, or CP/M 2.2.



Writers! Researchers! Engineers! Cut document makeup time from hours to minutes!

- ▶ Multi-Level Table of Contents
- ▶ Mulit-Level Index
- ▶ Lists of Tables and Figures
- ▶ Numbers Paragraphs (1.2, 1.3, etc)
- ▶ Prepares Footnotes
- ▶ Prepares Bibliography
- ▶ Clear commands, Numerous formatting options
- ▶ Reads documents with nested files

Specify When Ordering; Operating System, Computer Type and Disk Format. Free brochure and *nearly* free demo disk available.

STARMATE
(Special Introductory Price). **$145.00**
Demo All Disk (credits to purchase). $12.95

SOLUTION
TECHNOLOGY, INC.

"We **Deliver** Productivity"

1499 Palmetto Park Road
Suite 218
Boca Raton, FL 33432
305/368-6228

WordStar, PC-DOS and CP/M are trademarks of MicroPro, IBM and Digital Research respectively.

Check or COD, Florida residents add 5% sales tax. Dealer and Distributor inquiries welcome.

UNIX

compatible operating system that is described after a look at the user interface and the internals of the original UNIX system. (There's not a full how-to-use-it description, but rather a cursory look around and instructions for compiling and running CE programs under UNIX. To get started with UNIX, you should consider one of the introductory books.) The final chapter discusses "Implementing a Kernel"—the lowest-level portion of an operating system. The book concludes with a specification of the CE language. CE runs on UNIX and produces code for machines such as the LSI-11, the NS16032, the VAX-11, the MC6809 and MC68000. If you have UNIX running, you can get the ConEuc system from the University of Toronto Computer Systems Research Group. But get and read this book first! If you know a bit about UNIX and want a look "under the hood" without getting your hands too dirty, then this is the book to get. If you want to get down close to the engine, then you should read the orginial UNIX articles, many of which appeared in the *Bell System Technical Journal,* Volume 57 Number 6, July–August 1978. And if you *really* want to know what's inside the engine, then you'll have to get a source license and read the code. But that's very expensive. Start with the Holt book.

That's all the books for this time. In the next column I'll look at *The UNIX Book* and *Introducing the UNIX System,* and discuss some other works on UNIX that can be expected to appear in the near future.

Notes

UNIX is a trademark of Bell Laboratories.

CP/M and CP/M-68K are trademarks of Digital Research, Inc.

SourceTools is a trademark of Oregon Minicomputer Software, 2340 SW Canyon Road, Portland, OR 97201.

MARC is a product of Vortex Technology, Box 2284, Culver City, CA 90230.

Move up to Microsystems

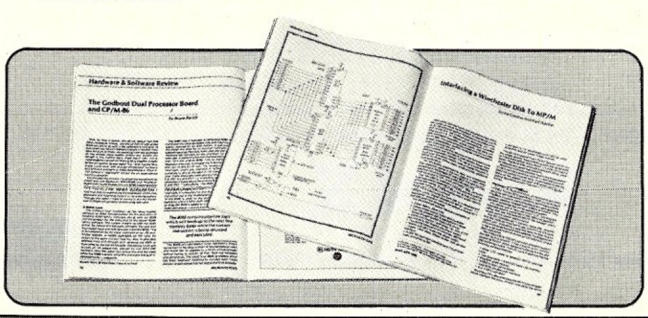
THE JOURNAL FOR SOPHISTICATED MICROCOMPUTER USERS.

Microsystems is not, in any sense, for beginners. Unlike "personal computer" magazines, *Microsystems* is written *exclusively* for expert users and designers in the software and support hardware field.

If you need a reliable, high-level, up-to-date journal devoted to CP/M,® MP/M,® MS-DOS,™ UNIX,® XENIX,™ OASIS and other operating systems, then *Microsystems* is for you. We feature tutorials on the modification and special use of systems, offer short-cut routines, show you debugging procedures, techniques for interfacing, and much more. Our one purpose is to keep the advanced microcomputer professional at the leading edge of DOS development and use.

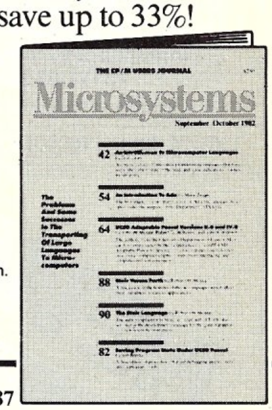
In a few recent issues of Microsystems you would have seen:

- Reviews of four C compilers: BDS C, Small C, tiny c TWO™ and Whitesmiths C.
- A programming style comparison: Digital Research PL/I-80™ vs. Microsoft BASIC.
- Twenty-seven 16-Bit DOS compared.
- Virtual segment procedures under UCSD Pascal.
- An in-depth look at MP/M-80 II.™
- A simple 6-byte hexadecimal ASCII conversion routine.
- Using CP/M's undocumented "Autoload" feature.
- An explanation of double density disk controllers.
- A comparison of five popular S-100 disk controller cards.
- OS-1—a UNIX-like DOS with a CP/M adapter.



Every issue of *Microsystems* keeps you at the forefront of the industry in customizing software and adapting hardware to meet new requirements and standards. You'll also find industry news, book and new product reviews, plus our regular *Software Directory* listing newly available programs.

To get the high-level systems information you want, join the club of serious microcomputer professionals who subscribe to *Microsystems.* Use the coupon below to order, and save up to 33%!



PROFESSIONALS PLEASE NOTE:

Your subscription to Microsystems may be tax deductible. Check with your accountant.

Trademarks:
XENIX: Microsoft; tiny c TWO: tiny c assoc.; PL/I-80, MS-DOS, MP/M-80 II: Digital Research.
Registered Trademarks:
CP/M, MP/M: Digital Research; UNIX: Bell Laboratories.

Microsystems CN 1987
Morristown, New Jersey 07960

YES, enter my subscription to *Microsystems* for:

☐ 12 issues at $19.97—I save 20%.

☐ 24 issues at $36.97—I save 26%.

☐ 36 issues at $49.97—I save 33%.

Savings based on full 12-issue subscription price of $24.97.

Mr.
Ms. ______________________ 4S22
(please print full name)

Company ______________________

Address ______________________

City ______________________

State ______________ Zip ______________

CHECK ONE: ☐ Payment enclosed. ☐ Bill me later.

Offer valid in U.S. and possessions only. Please allow 30 to 60 days for delivery of first issue. Satisfaction guaranteed or a prompt refund for unmailed issues.

NEW SUBSCRIBERS ONLY

UNIX

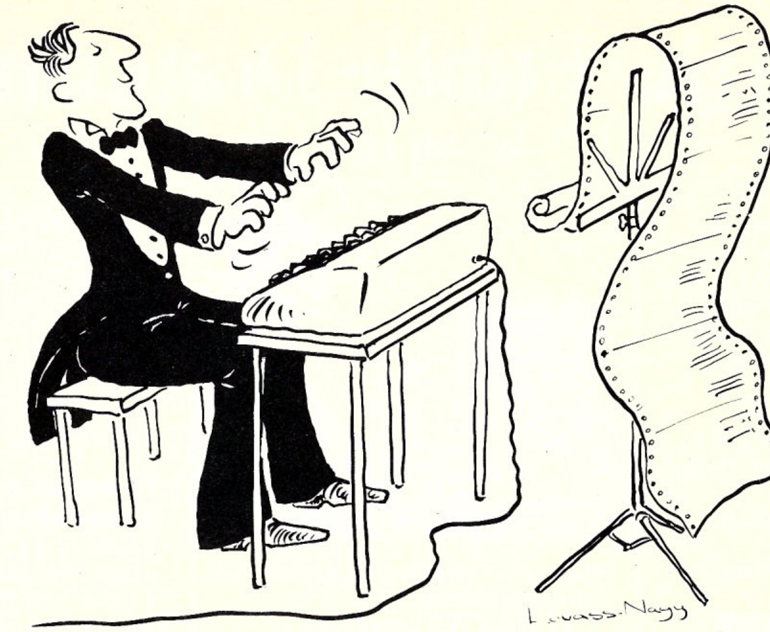
UNICA is a trademark of Knowlogy, Box 283A, Wilsonville, OR 97070.

Idris is a trademark of Whitesmiths.

Coherent is a trademark of Mark Williams.

Many of these books, and the Bell Labs UNIX manuals, can be ordered from Cucumber Bookshop, 5611 Kraft Drive, Rockville, MD 20852. For our Canadian readers, I have given my UNIX book shopping list to Exceltronix, 319 College Street, Toronto M5T 1S2, (416) 921-8941 and to the newly opened Computer Book & Supply Centre, 253 Eglinton West, Toronto M4R 1B1, (416) 489-3625, both of which said they would try to keep these books in stock.

Thanks to Geoff Collyer, Laura Creighton, and Mike Grossman at the University of Toronto for suggesting corrections to (and deletions from!) this month's column.



The Well-Tempered Cross-Assembler

Before Johann Sebastian Bach developed a new method of tuning, you had to change instruments practically every time you wanted to change keys. Very difficult.

Before Avocet introduced its family of cross-assemblers, developing micro-processor software was much the same. You needed a separate development system for practically every type of processor. Very difficult and very expensive.

But with Avocet's cross-assemblers, a single computer can develop software for virtually any microprocessor! Does that put us in a league with Bach? You decide

Development Tools That Work

Avocet cross-assemblers are fast, reliable and user-proven in over 3 years of actual use. Ask NASA, IBM, XEROX or the hundreds of other organizations that use them. Every time you see a new microprocessor-based product, there's a good chance it was developed with Avocet cross-assemblers.

Avocet cross-assemblers are easy to use. They run on any computer with CP/M* and process assembly language for the most popular microprocessor families.

XASM05	6805	$200 each
XASM09	6809	
XASM18	1802	
XASM48	8048/8041	
XASM51	8051	
XASM65	6502	
XASM68	6800/01	
XASMF8	F8/3870	
XASMZ8	Z8	
XASM400	COP400	
XASM75	NEC 7500	$500

(Coming soon: XASM68K .... 68000)

Turn Your Computer Into A Complete Development System

Of course, there's more. Avocet has the tools you need from start to finish to enter, assemble and test your software and finally cast it in EPROM:

Text Editor VEDIT -- full-screen text editor by CompuView. Makes source code entry a snap. Full-screen text editing, plus TECO-like macro facility for repetitive tasks. Pre-configured for over 40 terminals and personal computers as well as in user-configurable form.

CP/M-80 version	$150
CP/M-86 or MDOS version	$195

(when ordered with any Avocet product)

ROM Simulator -- ROMSIM by Inner Access eliminates need to erase and reprogram EPROM. Installed in an S-100 host, ROMSIM substitutes RAM for EPROM in external target system. 16K memory can be configured to simulate the 2708, 2758, 2716, 2516, 2732, 2532, 2764, 2564 in either byte or word organization. Avocet's configurable driver makes loading of HEX or COM files fast and easy.

From $495 depending on cabling and RAM installed.

EPROM Programmer -- Model 7128 EPROM Programmer by GTek programs most EPROMS without the need for personality modules. Self-contained power supply ... accepts ASCII commands and data from any computer through RS 232 serial interface. Cross-assembler hex object files can be down-loaded directly. Commands include verify and read, as well as partial programming.

PROM types supported: 2508, 2758, 2516, 2716, 2532, 2732, 2732A, 27C32, MCM8766, 2564, 2764, 27C64, 27128, 8748, 8741, 8749, 8742, 8751, 8755, plus Seeq and Xicor EEPROMS.

(Upgrade kits will be available for new PROM types as they are introduced.)

Programmer	$389
Options include:	
Software Driver Package	$ 30
RS 232 Cable	$ 30
8748 family socket adaptor	$ 98
8751 family socket adaptor	$174

Call Us

If you're thinking about development systems, call us for some straight talk. If we don't have what you need, we'll help you find out who does. If you like, we'll even talk about Bach.

VISA and Mastercard accepted. All popular disc formats now available -- please specify. Prices do not include shipping and handling -- call for exact quotes. OEM INQUIRIES INVITED.

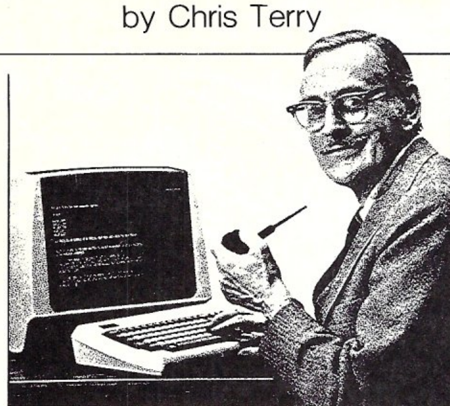
*Trademark of Digital Research.

AVOCET SYSTEMS INC.
DEPT. 783-M
804 SOUTH STATE STREET
DOVER, DELAWARE 19901
302-734-0151 TLX 467210

CIRCLE 201 ON READER SERVICE CARD

In the Public Domain

by Chris Terry



Quite a large number of telecommunication programs can be found scattered throughout the CPMUG and SIG/M libraries. However, they all fall into one of three main groups:

1. Ward Christensen's MODEM program for file transfer, and its many updates.
2. BYE and other remote-console programs.
3. Bulletin Board Software.

The MODEM program

The first release of this well-known program was in CPMUG Vol. 25. It was developed by Ward Christensen to allow CP/M users not only to communicate with each other and with mainframes in terminal mode, but also to create a memory log of incoming data that can be written to disk when main memory becomes full. Most important of all, this program allows binary files to be transferred in 128-byte blocks between the local disk drives and the line. A handshaking protocol detects transmission errors and automatically requests retransmission of an erroneous block.

MODEM has become the de facto standard for CP/M users because of its versatility, and commercial telecommunication software packages such as BSTMS, CROSSTALK, and the software for the Access Matrix portable computer have included the capability of using the MODEM file-transfer protocol as well as their own individual protocols.

A major breakthrough in utility and user friendliness came with the issue of **MODEM7**, extensively modified by Mark Zeiger and Jim Mills. This version was for PMMI only, but has since been updated to accommodate other plug-in and external modems. Three features make it one of the most sophisticated telecommunication packages available for microcomputers:

1. A command menu explains the meaning of each 3-letter command.
2. The CAL command makes use of the autodial capabilities of the PMMI and other intelligent modems. Any one of up to 26 numbers (displayed on the screen) can be dialed automatically by entering the letter assigned to it. Alternatively, you can enter a number from the keyboard; it will be dialed when the Return key is hit.
3. A Batch facility allows all of the files on a disk to be sent in directory sequence without operator intervention.

The latest version is **MODEM798** (CPMUG Vol. 93A). This includes many fixes of bugs or inconveniences found in previous versions. I/O overlays support the D.C. Hayes Smart Modem and the serial I/O ports of the Morrow Micro Decision, Xerox 820 and Kaypro computers. The .COM file is set up for a PMMI MM-103 modem with the base port at C0H (the standard PMMI port); if you have a different configuration, you will need Digital Research's MAC macroassembler and the special library file MODEM.LIB in order to reassemble the program for your system.

SMODEM37 (CPMUG Vol. 79) is a special version of MODEM7 set up for the D.C. Hayes SmartModem, an intelligent external 300-baud modem that can be completely controlled by sending ASCII command characters via a serial port. This version also supports the SmartModem 1200 (the D.C.Hayes 300/1200-baud, 212A-compatible modem).

APMODM22 (SIG/M Vol. 66) is another special version of MODEM (without the MODEM7 menu and batch facilities, but with a Help facility) for use in a 56K Apple computer with a Hayes Micromodem or CCS 7710A serial card and external modem.

Remote console programs

LINK is a CP/M transient program that allows a TRS-80 Model I or Model II, or Heath H8, or any S-100 computer using a PMMI or D.C. Hayes plug-in modem, to communicate with a remote computer that is not running a special telecommunications program. ASCII text received over the line can be selectively copied into memory and later written to a disk file. When sending an ASCII disk file to the remote computer, it sends one line at a time, suppressing the line feeds and waiting until it receives, a trigger character (e.g., a prompt) before sending the next line.The latest version is PLINK65 in SIG/M Vol. 68.

BYE67 (SIG/M Vol. 16) is a program that allows a computer equipped with a PMMI modem to answer an incoming call and to direct all console input and output to the remote instead of to the local console.

XMODEM41 (SIG/M Vol. 17 = CPMUG Vol. 69) permits RCPMs running BYE to download files to the caller. The RCPM must have XMODEM available; the caller runs it from the CP/M command level.

Bulletin board software

Volumes 7 and 8 of the SIG/M library contain MBasic programs and other utilities for setting up a Remote Bulletin Board System (RBBS), as well as documentation. Installing an RBBS is quite a complex project, and you may need help in getting started. It would probably be advisable to talk to the sysop of one of the RBBS or RCPM systems listed elsewhere in this issue.

Letters to the Editor

Dear Sir:

If I recall correctly, the quote on the cover of the March 1983 issue, "It's always scribble, scribble, scribble, Mr. Gibbon!" is a caption from a cartoon drawn by Burr Shafer circa 1950. It was one of a series entitled "Through History with J. Wesley Smith." Are there any prizes in this contest?

L. L. Kenan
4541 W. Bluefield Ave.
Glendale, AZ 85308

Chris Terry replies:
The original remark, quoted in Boswell's Life of Jonson, *is as follows: "Another damned, thick, square book! Always scribble, scribble, scribble! Eh! Mr. Gibbon?"*

It was attributed to the Duke of Gloucester upon being presented with the fourth volume of Gibbon's Decline and Fall of the Roman Empire.

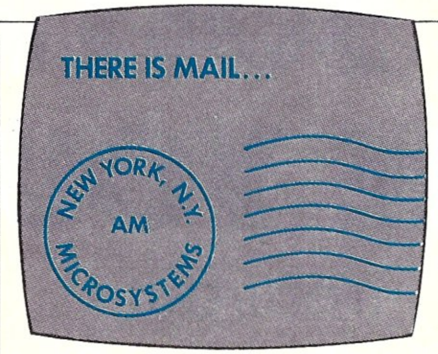
Gentlemen:

The following brief note is in response to a letter from G. A. Van Cott, published in "Letters to the Editor" in *Microsystems*, Jan. 83. Perhaps it will be of some assistance.

Few of the popular programming languages provide the intrinsic functions, ARCSIN and ARCCOS. One language which does provide these functions is Fortran. However, users who program in other languages can often write user-defined functions that are equivalent.

The adaptations of Basic implemented on many microcomputers lack several intrinsic functions, including ARCSIN and ARCCOS. However, many of these functions may be written in terms of intrinsic functions that are universal to even the most meager versions of Basic.

Such is the case of the inverse sine function and the inverse cosine function, which can be expressed in terms of the inverse tangent. (The inverse tangent function (ATN) exists as an intrinsic function in



most adaptations of Basic.) Therefore, the inverse sine (ARCSIN) expressed in terms of the inverse tangent is:

$$\text{ARCSIN } x = \text{ARCTAN } \frac{x}{\sqrt{1-x^2}}$$

Similarly, the inverse cosine expressed in terms of the inverse tangent is:

$$\text{ARCCOS } x = \frac{\pi}{2} - \text{ARCTAN } \frac{x}{\sqrt{1-x^2}}$$

For modularity and efficiency, some implementations of Basic provide the DEF FN statement, which is used to define a commonly referenced expression as a user-defined function. The inverse sine and the inverse cosine expressed as user defined functions are:

$$\text{DEF FNASN}(x) = \text{ATN } \frac{x}{\text{SQR } (1 - x * x)}$$

$$\text{DEF FNACS}(x) = \frac{\pi}{2 - \text{ATN}\left(\frac{x}{\text{SQR } (1 - x * x)}\right)}$$

R. J. Sadeck and Associates
Mini/Micro Systems Consultants
68 Charlotte White Rd.
Westport, MA 02790

Gentlemen:

In the January '83 issue, Andrew L. Bender of Neurological Services discusses the Memory Merchant 16K RAM board. He objects to phrases such as "pull-up-resistors" and "open-collector wired-OR" appearing in the manual. He implies that the manual should be restricted to instructions on how to insert the board and get it up and running.

Bender's thoughts are justified if the buyer can afford the downtime associated with returning the board to the manufacturer for repair—usually, two to four weeks. And after the warranty is out (six to 18 months)—throw the board away and buy new. Perhaps Neurological Services can afford this.

Computer Compatible Instruments cannot afford such "extravagance." Our systems must serve from 10 to 20 years. Downtime greater than the time it takes the user to get to a telephone, in my opinion, is intolerable. Nor can we afford to invest in spare boards or other parts.

We find that there is no substitute for an adequate service manual. We will not buy equipment until we have examined the manual and found it adequate. We are sick and tired of spending hours and days deciphering how a board works—when we could fix it in 20 minutes if an adequate manual were available.

Hewlett-Packard and Tektronix put out good manuals. But the best with which we are familiar are those put out by European companies. I recall a system built by Disa. We did not buy it, but we got the job of fixing it. "When can I get it back?"—came the usual owner query. I told him two months. I could see a week trying to figure out how it was built and how it operated, and six weeks getting parts. But the instrument came apart in two minutes rather than two hours. All components were marked and accessible. And the manual with it—beautiful! Voltages, waveforms, and explanations. Five minutes, and we knew where the trouble was. Twenty minutes, and we called the owner to come get the instrument.

It is true that solid-state cir-

Super Prices - Quick Delivery

MINIMUM SHIPPING $3.00 - NO SURCHARGE ON VISA/MASTERCHARGE

1-800-545-2633 - Continental U.S.

DISK DRIVES

SHUGART
5¼" SA400 (35) **150.00**
5¼" SA400L (40) **175.00**
5¼" SA410L (40) **200.00**
5¼" SA455L (40) **235.00**
8" SA801R (SS/DD) **355.00**

MITSUBISHI
5¼" M-4851 ½ size
DS/DD 40Tr **295.00**
5¼" M-4853 ½ size **410.00**
96 TPI same as TM100-4
5¼" M-4854 ½ size **439.00**
1.5 MG w/8" electronics formats same as IBM 1 & 2D 8" std
8" M-2894-63 .... **409.00**
Std 8" DS/DD 1.2 MG
8" M-2996-63 .... **439.00**
Thinline 8" DS/DD 1.2 MG

SHUGART 800/801R RECONDITIONED
90 Day Warranty **160.00**

TANDON
5¼" TM100-1 SS/DD **195.00**
5¼" TM100-2 (IBM)
DS/DD **239.00**

PERTEC
5¼" FD200-S
SS/DD 40TR **139.00**
5¼" FD250-5 *DS/DD 40TR* **195.00**

MPI
5¼" B-51 *40TR SS/DD* ... **230.00**
5¼" B-52 (IBM) *80TR DS/DD* **250.00**

All above drives SS/DD unless otherwise specified

DISK DRIVE CABINETS

***NOTE - Power supply will accommodate hard disk*

8" *DDC88V28 w/PS vertical - for 2-8" drives **300.00**
8" DDC88T-1 w/PS vertical - For 2 or 4-8" thinline drives **300.00**
8" DDC88T-2 w/PS vertical - inexpensive version for 2-8" thinline drives **225.00**
8" *DDC8V w/PS vertical - for 1-8" drive **249.00**
8" *DDC88H w/PS horizontal - for 2-8" drives **300.00**
5¼" DDC5H w/PS horizontal - for 1-5¼" drive ... **59.00**
5¼" DDC55H w/PS vertical - for 2-5¼" drives ... **89.00**

APPLE DISK DRIVES

"Thinline II" - ½ size 163K 40TR **235.00**
TEAC "Thinline" - ½ size 163K 40TR ... **275.00**
Micro-SCI **239.00**
Controller **79.00**
Rana Elite I **275.00**
Rana Elite II **439.00**
Rana Elite III **569.00**

DISKETTES

All diskettes have 1 yr. min. to lifetime warranty. Prices are for package of 10. Call for qty. pricing.

5¼" Sgl side/sgl den **19.00/10**
5¼" Sgl side/dbl den **20.00/10**
5¼" Dbl side/dbl den **26.00/10**
5¼" 10 sector **22.00/10**
5¼" 16 sector **22.00/10**
8" Sgl side/dbl den **29.00/10**
8" Dbl side/dbl den **36.00/10**

MONITORS/TERMINALS

Sanyo 2112 (15 MHZ) **80.00**
BMC 12A (15MHZ) 12" Grn **80.00**
BMC 9191 color 12" **269.00**
Amdek Color III **395.00**
USI 9" Grn (20 MHZ) **139.00**
USI 12" Grn (20 MHZ) **139.00**
USI 12" Amber (20 MHZ) **149.00**
Televideo 910 .... **560.00**
Televideo 925 .... **715.00**
Televideo 950 .... **899.00**
Televideo 970 .... **1095.00**

PRINTERS

C. ITOH
Pro-Writer I (8510A) 120 CPS **380.00**
Star-Writer F10 (Ltr quality) **1225.00**

STAR MICRO-TRONICS
Gemini 10 (100 CPS) 80 Col **310.00**
Gemini 15 (100 CPS) 132 Col **445.00**

OKI-DATA
82A Par + Ser 80 Col **399.00**
83A Par + Ser 132 Col **650.00**
84B Ber 200 CPS 132 Col **950.00**
92A 160 CPS - Par 80 Col **519.00**
93A 160 CPS - Par 132 Col **899.00**

NEC
NEC 8023 A **449.00**

TOSHIBA
P-1350 Ltr quality - 120 CPS **1500.00**
Draft mode 192 CPS Par or Serial
P-350 (100 CPS) Centronics Par-15" .. **429.00**

MPI
MPI 99G (9") **499.00**
MPI 150A1 (15") **935.00**

EPSON
MX80 w/Graphtrax **399.00**
M80FT w/Graphtrax **449.00**
FX80 .... **549.00**
MX100 .... **629.00**
MX100FX .... **CALL**

AC SURGE ELIMINATORS

Lemon (6AC outlets - 3 prong) **4600**
Lime (5' - 3 prong pwr cord w/on-off switch **69.00**
Orange - AC surge + EMI filter (6 outlets) **119.00**
Plum - EMI-RF filter **39.00**

MODEMS

Hayes Smart Modem 300
209.00

PARTS

TR1602B	**$2.50 ea.**	TMS2716(5+12V)	**$3.95ea.**
2114L2	**$1.50 ea.**	TMS2732	**$4.95 ea.**
TMS2532	**$5.95 ea.**	TMS4164(200NS)	**5.95 ea.**
2716 (5V)	**$3.95 ea.**	TMS6116-4(200NS)	**$4.75 ea.**

MANY OTHERS IN STOCK

DATA CABLES

8" DSC 88-2SKT - for 2-8" drvs w/skt. conn. **16.95**
5¼" DSC55-2SKT - for 1-5¼" drvs w/skt conn. **$15**
RS232MM-5' (male to male) **16.00**

E-PROM ERASERS

QUV-T8/1 (hobby) **49.95**
QUV-T8/2 (Industrial version) **68.95**
QUV-T8/2 (w/timer & safety switch) **97.50**

CONNECTORS

RS232 Connectors

SOLDER TYPE		FLAT RIBBON TYPE	
DB25P	**2.50**	IDC25P	**6.25**
DB25S	**3.00**	IDC25S	**6.60**
DB25 Hood	**1.00**	IDC25 Hood	**1.60**

S-100 Connectors .... **10 for 25.00**

S-100 PRODUCTS

GIGANTIC REDUCTIONS IN PRICE!

CARD CAGES/MOTHER BOARDS

**IEEE - 696 - No termination required*

Slots	Bare Bd	A+T	w/card cage
4	15.00	40.00	60.00
6	20.00	48.00	70.00
8	25.00	69.00	100.00
12	30.00	99.00	140.00
18	45.00	150.00	200.00
22	60.00	185.00	——

All card cages will accommodate a 4" fan
Add $20.00 for 1 fan - Add $30.00 for 2 fans

MAINFRAMES

For 2 Standard 8" Drives
MF+DD6 (6 slot M/B) **699.00**
MF+DD8 (8 slot M/B) **749.00**
MF+DD12 (12 slot M/B) **799.00**

For 2 Thinline 8" Drives
IMF+DD6C (Cadillac version) **500.00**
IMF+DD6F (Ford version) **350.00**

For 2 - 5¼" Disk Drives
MF+MD12 (12 slot M/B) **575.00**

Standard Plain Front
MF+12 (12 slot M/B) **500.00**
MF+22 (22 slot M/B) **600.00**

All mainframes except IMF+DD6F have EMI filter, 2 AC outlets, 15 ea. DB25, 2 ea. 50 pin, 2 ea. 34 pin, 1 ea. Centronic cutouts, power supply for 8" MF (−5V1A/+5V6A/+8V16A/±16V3A/+24V6A)

SIERRA DATA

SDS-SBC-100-Z80(4mhz)
master 2 serial 2 par/floppy controller/64k ram . **$855.00**
SDS-SBC-100S-4mhz slave/2 serial
2 par/64k ram **$565.00**
SDS-ZSIO/4 - 4 serial port I/O bd **$235.00**
SDS-MUX - RS232 multiplexer bd **$235.00**
SDS-HDI-M - Hard disk bd for micropolis ... **$129.00**
SDS-CPM/B105--cp/m for SBC100 w/BIOS **$150.00**
SDS-Turbodos - Multi-user for master & slaves ... **$845.00**

BOARD SETS

Best Bare Board Set Available
QTC-SBC 2/4 CPU
QTC-EXP+III 256K Memory bd.
QTC-FDC 5/8 Floppy disk controller
Bare Board Set $150.00

1) Includes manuals & assembly instructions
2) Parts available 3) Monitor & B10S available Add $30.00

QT CLOCK/CALENDAR

S-100 or Apple • Time in hrs, min, sec. • AM/PM or Military Format • Date in Mo., Day, Yr, Day of Week & Leap Year recognition • 4 hard interrupts (1024 Hz, 1Hz, 1 min, 1 hr) • On board battery (will last 14 mos. w/no power on)

QTC-CCS-BB (S-100) **$45.00**
QTC-CCS-K (kit) for S-100 .... **$100.00**
QTC-CCS-A (A+T) for S-100 .... **$125.00**
QTC-CCA-BB (for Apple) **$33.00**
QTC-CCA-A (for Apple) **$100.00**

COMPUPRO

S-100 - all assembled and tested

System 816A **4000.00**
20 Slot MB **210.00**
CPU 8085/88 .... **389.00**
Ram 17-64K **410.00**
Ram 21-128K **900.00**
Disk II **610.00**
System Support I **310.00**
Interfacer 4 .... **310.00**
Active Teammates **50.00**
Enclosure 2 (disk) **675.00**

Call for others

CPU / MEM / I/O

QTC-SBC 2/4 BB 1 ser 1 par **$50.00**
QTC-SBC 2/4 A A+T **$265.00**
QTC-Z+80 BB 1 serial **$28.00**

Teletek FDC-I A+T **$595.00**
Teletek Systemaster A+T **$695.00**

QTC-EXP+III Bare Bd. (dynamic) **$65.00**
QTC-EXP+III 64K A+T (64K/256K or 1 MEG) ... **$450.00**

S-RAM 64K BB (static) **$60.00**
S-RAM 64 A+T **$349.00**
QTC-ADA *ADA Converter* A+T **$375.00**
QTC-Dual GPIB-488 *IEEE 488 Interface bd.* A+T **$475.00**

MasterCard

Retail Sales
123 East 200 South
Salt Lake City, Utah 84111

The Great Salt Lake Computer Company, Inc.
1-801-363-3314

Mail Orders
P.O. Box 3150
Salt Lake City, Utah 84110

VISA

CIRCLE 39 ON READER SERVICE CARD

Letters continued . . .

cuitry gives little trouble. But things do go wrong. Lightning strikes. Fans stop and temperatures go out the top. We advise the buyer to make sure that the maintenance manual be such that he will be able to get that system back up and running when it does go down. We recommend having two manuals: an Operator's Manual for the Bender type, and a Maintenance Manual written so the board or device can be repaired with reasonable effort.

With Bender, we will agree that sales pitches have no place in a manual. The seller has already suckered us into buying his product. Needed now is info enabling us to fix it—rather than bring it back and bounce it off his head. Nothing is worse than trying to fix something, opening the manual, and finding:—"This instrument has been designed to give long and troublefree service." Grrrr!

R.O. Whitaker
Computer Compatible Inst. Co.
4719 Squire Drive
Indianapolis, IN 46241

FORTH for Z-80®, 8086, 68000, and IBM® PC

FORTH Application Development Systems include interpreter/compiler with virtual memory management and multi-tasking, assembler, full screen editor, decompiler, utilities, and 130+ page manual. Standard random access files used for screen storage, extensions provided for access to all operating system functions.

Z-80 FORTH for CP/M® 2.2 or MP/M II	$ 50.00
8080 FORTH for CP/M 2.2 or MP/M II	$ 50.00
8086 FORTH for CP/M-86 or MS-DOS	$100.00
PC/FORTH™ for PC-DOS, CP/M-86, or CCPM	$100.00
68000 FORTH for CP/M-68K	$250.00

FORTH+ Systems are 32 bit implementations that allow creation of programs as large as 1 megabyte. This is the only language that supports the entire memory space of the 8086/88 directly for programs and data!

PC/FORTH+ for PC-DOS or CP/M-86	$250.00
8086 FORTH+ for CP/M-86	$250.00

Extension Packages for FORTH systems

Software floating point (Z-80, 8086, PC only)	$100.00
Intel 8087 support (8086, PC only)	$100.00
AMD 9511 support (8086, Z-80 only)	$100.00
Color graphics (PC only)	$100.00
Symbolic interactive debugger (PC only)	$100.00
Cross reference utility	$ 25.00
PC/GEN™ (custom character sets, PC only)	$ 50.00
Hierarchical file manager	$ 50.00
B-tree index manager	$125.00
B-tree index and file manager	$200.00

QTF+ Screen editor and text formatter by Leo Brodie,
for IBM PC with IBM or Epson printer .... $ 50.00

Nautilus Cross Compiler allows you to expand or modify the FORTH nucleus, recompile on a host computer for a different target computer, generate headerless code, and generate ROMable code with initialized variables. Supports forward referencing to any word or label. Produces load map, list of unresolved symbols, and executable image in RAM or disk file. No license fee for applications created with the Cross-Compiler. Prerequisite: one of the application development systems above for your host computer.

Hosts: Z-80 (CP/M 2.2 or MP/M II), 8086/88 (CP/M-86 or MS-DOS), IBM PC (PC-DOS or CP/M-86), 68000 (CP/M-68K)
Targets: 8080, Z-80, 8086/88, 6502, LSI-11, 68000, 1802, Z-8

Cross-Compiler for one host and one target	$300.00
Each additional target	$100.00

AUGUSTA™, ADA subset compiler from Computer Linguistics, for Z-80 computers under CP/M 2.2 ...... $ 90.00

LEARNING FORTH computer-assisted tutorial by Laxen and Harris for CP/M, includes Brodie's "Starting FORTH" .... $ 95.00

Z-80 Machine Tests Memory, disk, printer, and console tests with all source code in standard Zilog mnemonics $50.00

DATA ACE, fully relational data base system from CSD, for the IBM Personal Computer. Faster and more powerful than dBASE II $595.00

FORTH application development systems require 48 kbytes RAM and 1 disk drive, Cross-Compilers require 64 kbytes. All software distributed on eight inch, single density, soft sectored diskettes except PC/FORTH on 5¼ inch single sided double density diskettes. Prices include shipping by UPS or first class mail within USA and Canada. California residents add appropriate sales tax. Purchase orders accepted at our discretion.

Laboratory Microsystems, Inc.

4147 Beethoven Street
Los Angeles, CA 90066
(213) 306-7412

CIRCLE 26 ON READER SERVICE CARD

Using RCPM Systems Effectively

A wealth of free software is available from RCPMs: Here's how to make best use of them

by Ben Bronson, Kelly Smith, and Keith Peterson

The following are guidelines for using Remote CP/M systems effectively. An up-to-date listing of these systems follows this article. These suggestions (and rules of etiquette) should be followed carefully to get the most from your long distance charges, and to respect the privacy of the SYSOPS (system operators) who are giving generously of their time and effort to provide this service.

1. RCPM systems carry programs that will run only under the CP/M operating system, most of them public domain software distributed by CPMUG or SIG/M, the two main CP/M users' groups. Some of the programs (especially those in Basic, Pascal, and C, but also a few in Z80 or 8080 assembly language) can be converted to run on other machines quite easily. Systems dedicated to downloading programs for the PET, APPLE, TRS-80, and NorthStar do exist, however. Ask experienced users in your area, or check through "other systems" listings on local CBBSs, ABBSs, Forum-80s, etc. Such systems are often listed in files named OTHERSYS.xxx.

2. XMODEM is a modification of Ward Christensen's original MODEM program designed to allow the transfer of files between the exchange system and yours. The exchange system needs the program, not you; you will probably be using one of the many versions of MODEM itself: e.g., MODEM2, MODEM221, MODEM 926, MODEM7, MODEM75, TRSMODEM, APMODEM, CMODEM, MBOOT3, etc. BSTAM and most of the other commercial terminal programs will not work properly with XMODEM's S or R file transfer modes, which use the Christensen protocol. BSTAM is also available on many remote systems to allow multiple file transfers, but it is a licensed program (i.e., *not* public domain), so it is only available to you if you buy it.

Certain programs, such as PLINK and some versions of MODEM, can allow files to be "captured" without the benefit of handshaking or error checking. However, if you use such methods, you should bear in mind the following:

a. Errors occurring during the transfer will go undetected unless you catch them (no handshaking/error checking) by visual means, or by transmitting two or more copies of the same file. You can then run COMPARE or DF against them until you get a pair that "agree."

b. You may not transfer .COM files. You may, however, use UNLOAD to create .HEX files from .COM files, and the .HEX files can be sent as in #1 above. The receiving end can then run Digital Research's LOAD to recreate the .COM file from the .HEX file.

Keith Peterson, Box 309, Clawson, MI 48017

c. You may not be able to transmit some files if they contain certain values, such as hex 1A, which indicates end of file.

3. Non-XMODEM program transfer systems are starting to appear: e.g., LICC's BBS on Long Island [at (516) 561-6590; 110-600+ baud; 24 hrs, no callback] has a few downloadable ASCII files, but at present these must be captured like CBBS messages and then saved to disk without any error-checking protocol. John Wood, the sysop of Santa Clara CBBS [at (408) 241-1956; 110-600+ baud; 24 hrs, no callback] has announced plans to implement late-night exchanges; whether these will use XMODEM is not yet clear. Chuck Forsberg's YAM (Yet Another Modem) program is written in BDS-C, and the source is available on several systems in RCPMLIST.

4. Some XMODEM systems also operate as real CBBSs, including those operated by Dick Mead and Steve Vinokouroff. Others, like Keith Petersen's and CCCC's, use a skeletal subprogram for handling messages called "MINICBBS". Their emphasis is on exchanging software, not bulletin-board-type information. Most RBBS systems focus on software, but there are variations.

Note that you cannot transfer a .COM file to any of these systems; the XMODEM program will not allow it. Some systems will automatically rename any incoming .COM files to .OBJ. In addition, some systems will not let you take .COM files from them. Most systems will also prevent you from taking any files that the SYSOP does not want to distribute. Many systems contain .OBJ files; these are really .COM files that you are allowed to download. By calling them .OBJ, the SYSOP can prevent them from being run on his system, but still make them available to you via the XMODEM program.

5. "CBBS", "MINICBBS", "RBBS" (and a variant, "MINIRBBS") are the user-logging/message-taking programs used by almost all of the systems on RCPMLIST. You will need one of them if you are interested in starting your own RCPM/XMODEM system. CBBS & MINICBBS are assembly-language programs and cost $50. RBBS & related programs are in MBasic and free, but not quite as good as CBBS, and, of course, you'll need the MicroSoft Basic compiler to use them ($250). Ask the your local sysop for details, or read RBBS-USE.DOC, RBBSNSTL.DOC, or CBBSFORM.DOC found on many systems.

6. A few CBBSs and RBBSs have software-exchange functions that require the use of a password. Some such systems are indicated in the RCPMLIST that follows. If you wish to be given a password, you can usually leave a message to the SYSOP on the Bulletin Board of that system.

7. Presently, many of these systems are running

with hard disks. Several are running MP/M. You should become familiar with the USER command in order to move around on these large-capacity disks effectively. You should also become familiar with the SQ and USQ programs. These compress files to 50–85% of their original size. Many such "squeezed" files will be seen on these RCPM systems, because squeezing files allows more files to be placed online, and they transfer faster. See the file named SQ-BRIEF.DOC, or SQUEEZER.DOC. Squeezed files are identified by the letter Q in the second position of the filetype; for example, MODEM7.DQC must be run through USQ to produce the readable MOD-EM7.DOC file. Be sure you download USQ-15.OBJ and rename it at your end to USQ.COM so that you can "unsqueeze" the files you get.

8. All of the RCPMs will accept 300 baud. Many will accept 450 or 600 baud if you can push your "300" baud modem that fast. You can change baud rates during a session by running NEWBAUD on the host system, resetting your rate, and then re-entering terminal mode and pressing a few returns until the host senses what your new baud rate is. Weekends or holidays seem to give better luck with pushing to higher speeds, but the quality of your modem is the biggest factor. Several systems now accept 1200 baud also.

9. You should set your modem into ORIGINATE mode, and run it at FULL DUPLEX while accessing these and most host systems. The normal characteristics are: 8 bits, no parity, 1 stop bit, 300 baud for initial log-in.

10. Some "300 baud" modems are capable of speeds greater than 300 baud. The Racal-Vadic 3451s (and the manual MODEMPHONES) have 600 baud capabilities (at ordinary 103 modem frequencies). The PMMI (Potomac Micro Magic Inc.) modem widely used on the S100 RCPMs is often run at 600 and even 710 baud. Two other brands of modem will run at 600: the IDS on-board S-100 modem and Novation's newly announced RV 3451-lookalike "triple modem." Neither are yet used by an RCPM system. The other systems on the RCPM list all run either D.C.Hayes or assorted brands of external modem. They usually do not send or receive faster than 300 baud.

11. Do not despise things that are free. It is a curious fact in the computer world that value bears almost no relation to price. Some of the best programs for personal computers are in the public domain. They often work better and are better supported than software costing hundreds of dollars. And, in terms of documentation, commercial programs do not even come close.

12. And please, most of these systems operate from private home telephones. Be courteous and call only during the posted operating hours. Note that system operating times are posted in *local* time, not necessarily in your time zone. Be sure to translate these times to your own before calling a remote system.

Note: If the system says a "call back" is needed, you must call the number, let the phone ring *exactly* once, then hang up and redial. On your second call, the modem will answer on the first to third ring. If you get no answer after three rings, hang up and try again later. The purpose of all this is that the "call back" systems are using telephones that are also used for personal voice matters. Be sure to respect the generosity and the request of the SYSOPS.

13. Many systems have time limits (usually one hour in prime time). You can make more effective use of your time by downloading the directory menus, usually called DISKMENU.DOC. You can then hang up, print it off, mark the files you want, and then call back. Another effective way to use these systems is to run WHATSNEW on each disk as you log into it; this will list files newly added and deleted since a specified date. A separate copy of WHATS-NEW is kept on *each* of the system's drives, so you'll have to execute WHATSNEW on each drive in order to see all of the new files. This can be easily done by just prefixing the drive name to the command. For example, to see what's new on a 3 drive system, type:

```
A:WHATSNEW
B:WHATSNEW
C:WHATSNEW
```

14. It would be a good idea to practice file transferring with someone in your local calling area before going to a distant RCPM. If you do not have a local RCPM, practice with another CP/M user.

15. Be sure to look at the .DOC file or the first part of the .ASM file to learn the purpose or restrictions of a program before you bother to download it. If the filetype is AQM or DQC, indicating the file is "squeezed," use TYPESQ to view it.

16. Be sure to type BYE before you log off to reset the system for the next caller. If you are hopelessly hung up in a loop, then just hang up and call back in 30 seconds or so; the systems will automatically reset after they sense that the remote caller has hung up the phone.

17. If you discover a problem with the host system, report it by leaving a message on that bulletin board, or by contacting the SYSOP. He may never know if someone does not tell him. This is an easy way to repay for the use of that system, and it will be appreciated by the SYSOP.

18. Do *not* use these systems for commercial messages, ethnic jokes, political commentary, heavy-breathing messages, and the like. Users are of all ages and persuasions, and restraint (or maturity) will be appreciated by all. If you know of anyone abusing a system, please discourage it, or report it to the SYSOP. This sort of abuse has resulted in the loss of several good RCPM systems, wider use of pass-worded systems, and other restrictive curtailments.

19. Most remote systems have far more files than are online at any one time. Most will accept requests on their bulletin boards to leave certain files online for you for a specified interval of time, and on a certain disk and user #. If you do not see it, look for a .DOC file indicating that the SYSOP offers such help.

20. Speaking of help, be sure to enter HELP at the A> prompt to learn more about CP/M and what the host system offers.

RCPM systems carry programs that run only under CP/M; most are public domain, distributed by CPMUG or SIG/M.

A Directory of Remote CP/M Software Exchange Systems

A summary of Remote CP/M Software Exchange Systems using XMODEM for program transfers

by Jud Newell and Kim Levitt

Northeast

Programmer's Anonymous RCP/M, (207) 839-2337. Ralph Trynor. NCB. 24 Hrs. B3;LD1;DSK: 180k. [Gorham, Maine] Interest in new software, modem programs, help and software for Osborne. (System runs on an Osborne 1.)

PROVIDENCE RCP/M, (401) 751-5025. Mark Rippe. CB. 1000 Sat-2200 Sun. B2;LD2,3;DSK:1.2mb [Providence RI]

TORONTO ONTARIO RCP/M SYSTEMS ONE & TWO, (416) 231-9538,231-1262, Jud Newell. NCB. 24 hrs. B5;LD3;DSK: 20mb Hard and 10mb Hard. [Toronto, Ontario, Canada] Interest in new releases of Software. Online programs exceed 1000, and online program catalog of 6000 on request programs available. (System formerly named MISSISSAGUA RCPM.) (SYSTEM ONE and SYSTEM TWO now require preregistration and payment of an annual $25 fee. System One will allow limited access without pre-registration, System Two will not allow any access. Call System One for Access details. Membership is limited.

MISSISSAUGA ONTARIO HUG-RCP/M, (416) 231-4174, Toronto Heath Users Group. NCB. 1800-0600 wkdys, 24 hrs wkends. B1;LD1;DSK:2+mb. [Toronto, Ontario, Canada]

Mid-Suffolk RCP/M and Data Exchange, (516) 751-5639, Al Klein, CB. 1700-0900 wkdys, 1700 Fri - 0900 Mon. B2;LD2,3; DSK:400k. [Long Island, NY] Sysop interested in new programs for all micros. Note Phone will be answered voice 0900-1700 Mon-Fri.

Johnson City, NY SJBBS, (607) 797-6416, Charles ---. NCB. Eves., etc. B2,LD1,DSK:2mb. [Upstate NY]

SuperBrain RCPM, (617) 862-0781, Paul Kelly. NCB. 1900-0700 wkdys, 24 hrs wkends. B5;LD2,3,4;DSK:300k. [Lexington, MA] Special interest in Superbrain-adapted CP/M programs.

Rochester RBBS, (716) 425-1785, Arnie McGall. NCB. 24 hrs. B5;LD2,3,4;DSK:2.4mb. [Upstate NY]

Bearsville Town SJBBS, (914) 679-6559, Hank Szyszka. NCB. 24 hrs. B1;LD1;DSK:4mb. [Upstate NY]

Woodstock RCP/M RBBS. (914) 679-8734. John Doak. NCB. 24 Hrs. (Machine answers after 3rd ring.) B3;LD2,3,4;DSK:2.8mb. [Woodstock, NY] Heath H8 System. Sysop interested in all CP/M software, plus ham radio software. CPMUG and RCPM library is available. Baud rates in addition to 212A 1200/300: 75, 110, 134, 150 and 450.

Brewster RBBS, (914) 279-5693, Paul Bosshold & Carl Erhorn. CB. 9pm-8am Wkdays, 24 hr wkends. B1;LD1;DSK:500k. [Downstate NY] (S-100 based. General CP/M software)

East Central

Flanders, NJ. (201) 584-9227, Ken Stritzel. NCB. 24 hrs, B3(0700 Mon-1700 Fri);B1(1700 Fri-0700 Mon); LD2;DSK: 26mb Hard. [No NJ] Emphasis on new programs and recent updates of standard programs

---

Paul Bogdanovich's RBBS, (201) 747-7301, Paul Bogdanovich, NCB. 1800-2300 wkdys, 0800-2300 wkends. B1;LD1;DSK:1mb. [NJ]

RIBBS of Cranford, New Jersey (201) 272-1874, Bruce Ratoff. NCB. 24 hrs. B1 (B3 on request); LD2,3; DSK: 3mb. [NJ] General CP/M software; Bulletin Board of SIG/M (Special Interest Group/Microcomputers, ACGNJ)

The C-Line, (201) 625-1797, David Fiedler. NCB. Mon-Fri 2000-0900, 24 hrs wkends. B1; LD2,3?,4?; DSK: 2 Mb. [NW NJ] Running CP/M under MicroShell. Special interest UNIX, UNIX-like systems, C software. On-line instruction in UNIX and C is planned.

Allentown RBBS/RCPM System, (215) 398-3937, Bill Earnest. NCB. 24 hrs. B1;LD2,4;DSK:10mb Hard. [E. Pa] General CP/M software. Bulletin board of the Lehigh Valley Computer Club & SIG/M

Laurel, MD. RCPM/RBBS, (301) 953-3753, Wayne Hammerly. NCB. 24hrs. B2;LD2;DSK: 64mb hard. [Washington DC Area] Now running on Molecular Super micro 32, with three phone lines for remote use. 953-3753,3754 are 300 baud, 953-3755 is 1200 baud.

Jud Newell, 4691 Dundas St. W., Islington, Ontario M9A1A7 Canada

---

BHEC RBBS/RCPM, (301) 661-4447, Walt Jung, Charlie Schnepf, Harry Barley. NCB. 6pm-9:30am Daily, 9pm Thu-9:30am Fri, 5pm Sat-9:30am Mon. B2;LD2,3,4;DSK:10mb Hard. [Baltimore, MD]

Arlington RCPM/DBBS of Virginia, (703) 536-3769, Eliot Ramey, NCB. 2200-1500 wkdays, random wkends, B1;LD2,3,4;DSK:800k. [Arlington, Va] Recent updates and new releases.

OxGate-007 Grafton VA, (804) 898-7493, Dave Holmes. NCB. 24 hrs. B2;LD2;DSK:200k. [Tidewater VA] Carries CP/M, TRS-80 & Apple software; plans for setting up dual system (on one line) with LNW-80 as well as CP/M computer. Active as bulletin board.

State College, PA. CUG-NODE, (814) 238-4857, Joe Shannon. NCB. 24 hrs. B2;LD1;DSK:3mb. [Pa]

Midwest

Logan Square RCPM, (312) 252-2136, Earl Bockenfeld. NCB. 24 hrs. B1;LD2,3,4;DSK:1mb. [Chicago] Special interest in recent releases and developing on-line databases, with daily change of software on B drive

---

Palatine RCPM, (312) 359-8080, Tim Cannon. NCB. 24 hrs. B3;LD2,3,4;DSK:4.8mb. [Chicago area] Emphasis on very recent releases, updates to existing programs and BDS C programs. Disks on B:;C:; and D: are changed daily.

IBM-PC BBS, (312) 647-7636, Gene Plantz. NCB. 1800-0700 wkdys & 24 hrs wkends. B5;LD2,3,4;DSK:200k. [Niles: Chicago area]

AIMS, Hinsdale, Ill. (312) 789-0499, Mark Pulver. NCB. 24 hrs. B1;LD2,3,4;DSK: 10mb Hard. [Chicago area]

NEI RCPM System, (312) 949-6189, Chuck Witbeck. NCB. 1800-0100 wkdys, 1200-0100 wkends. B1;LD2,3,4;DSK:2mb. [Chicago area] Main emphasis on communications programs, including versions adapted to non-standard CP/M systems.

Technical CBBS, (313) 846-6127, Dave Hardy. NCB. 24 hrs. B1;LD2,3,4;DSK:3mb. [Detroit area] Emphasis on very recent releases. RCPM sysops desiring access to passworded RCPM Clearing House system should leave msg on TCBBS. Active message system

---

Royal Oak CP/M, (313) 759-6569, Keith Petersen. CB. 24 hrs. B1 (B5 available on request);LD2,3,4;DSK:10 mb Hard. [Detroit area] Emphasis on new programs & recent updates of standard progs.

Southfield, MI, RBBS/RCPM, (313) 559-5326, Howard Booker. NCB. 24 hrs. B2;LD2,3,4;DSK: 2.7mb. Special interest in BDSC programs, doc. files and recent updates of standard programs.

MINICBBS/Sorcerer's Apprentice Group, (313) 535-9186, Bob Hageman. NCB. 24 hrs. B1;LD2,3,4;DSK: 500k. [Detroit,MI] Running on Exidy Sorcerer. Special interest in adapting CP/M software and assorted hardware to Sorcerer systems.

OZZY #1 - Osborne RBBS/RCPM of Milwaukee, (414) 342-4599. Jim Ryan. NCB. Mon-Thu 11pm-8am, Fri-Sat 8pm-12pm, Sun 8pm-8am. B2; LD2,3,4; DSK: 416k. [Milwaukee, WI] Special Interest in Osborne Software and Info. System software is first generation and does have some bugs.

Fort Fone File Folder, (414) 563-9932, Al Jewer, Shawn Everson, Ron Fowler. NCB. 24 hrs. B1;LD1;DSK: 20mb Hard. [Ft. Atkinson, WI]

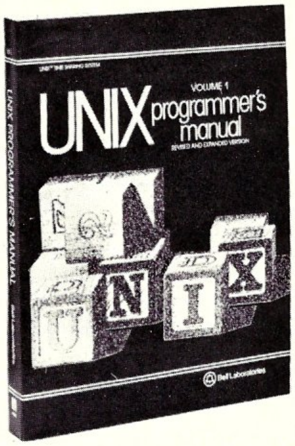
Mike's RCP/M System, (414) 647-0903, Mike Wesolowski. NCB. Mon-Fri 6pm-6am, 24 hrs wkends & holidays. B2;LD2,3,4;DSK: 700k. [Milwaukee, WI]

Cincinnatti RBBS, (513) 489-0149, Henry Deutsch. NCB. 1800-0600 daily. B1;LD2;DSK: 1.8mb. [Ohio] Specializes in Telecommunications.

West Carrolton RCP/M, (513) 435-5201, Rich Malafa & Bob Drake. NCB.24 hrs. B1;LD2;DSK:11mb Hard. [Dayton, OH]

Columbus CBBS, (614) 272-2227, (268-CBBS), John Walpole. NCB. 24 hrs. B1;LD2,3,4;DSK: 300k. [Ohio] Now running MP/M, on a Tarbell SD controller; occasional slow response means sysop is also using system; special interest in BDS-C programs. Also active as bulletin board.

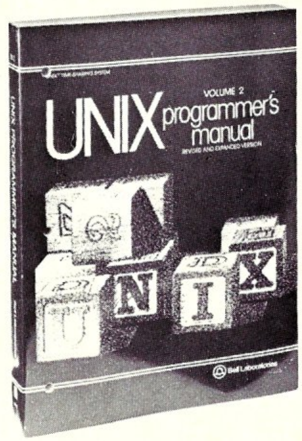
YOUR SOURCE OF REFERENCE FOR THE MANAGEMENT, USE AND PROGRAMMING OF CP/M AND UNIX OPERATING SYSTEMS



UNIX PROGRAMMER'S MANUAL Vol I
By Bell Laboratories

Volume 1 covers using the utilities, the comprehensive text processing facilities, the system calls, the library, file conventions, special languages, compiler-writers' tools and more All the building blocks you need for easy program development. Included are guides to the internal construction of the system and the language C, as well as descriptions of Fortran 77, the Awk report generator, and the unlimited precision desk calculator.

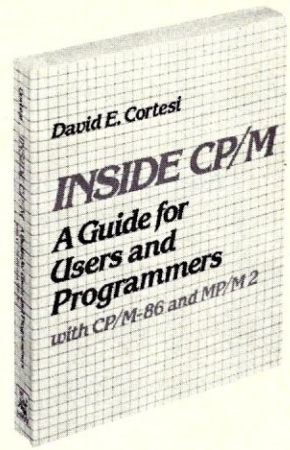
432pp., Paper $34.95* ISBN 0-03-061742-1



UNIX PROGRAMMER'S MANUAL Vol II
By Bell Laboratories

Volume II is comprised of general tutorial information and more detailed descriptions of major programming languages such as Fortran or the **troff** typesetting package.

611 pp., Paper $34.95* ISBN 0-03-061743-X



INSIDE CP/M: A GUIDE FOR USERS AND PROGRAMMERS with CP/M 86 and MP/M2
By David E. Cortesi

A complete guide to CP/M, the most popular — and fastest growing — microcomputer disk operating system. More than just a technical manual, this unique volume is equally valuable to users, managers, and programmers. The reference chapters are organized for practical day-to-day use. Information is included for the CP/M 86™ and MP/M 2™ systems.

571 pp., Paper $23.95* IBSN 0-03-059558-4

MAIL TODAY FOR 30 DAYS FREE TRIAL

To: Deborah Mitera
CBS EDUCATIONAL & PROFESSIONAL PUBLISHING
383 Madison Avenue, New York, NY 10017

Indicate Book By ISBN Code

☐ ISBN 0-03-061742-1 ☐ ISBN 0-03-061743-X ☐ ISBN 0-03-059558-4

Name ____________________

Company ____________________

Street ____________________

City ____________ State ______ Zip ______

☐ Bill me, including shipping and handling charges plus applicable sales tax.
☐ Check enclosed for...

Total Amount of Order	1.	______
Local Sales Tax (% x Line 1)	2.	______
Shipping (5% x Line 1)	3.	______
Handling		.75
Amount Due		______

VISA ________ Exp. Date ____ MasterCard ________ Exp. Date ____

*Prices subject to change without notice.

Let Me Know Your Book / Software Ideas...
Contact Brete Harrison, Editor

CBS
EDUCATIONAL & PROFESSIONAL PUBLISHING
383 Madison Avenue, New York, NY 10017
Nationwide: 800-227-1617 ext. 336
California: 800-772-3545 ext. 336

CIRCLE 32 ON READER SERVICE CARD

RCPM Directory continued . . .

Pickerington RBBS, (614) 837-3269. Greg Bridgewater. NCB. ??? Schedule. B2;LD2;DSK: 1mb. Running TRS-80 with Omikron. [Ohio]

Mission, KA, (913) 362-9583, Dave Kobets. NCB. 24 Hrs. B3;LD2;DSK: 2mb. [KS]

AlphaNet RCP/M RBBS, (913) 843-4259. Larry Miller. NCB. 1800-0900 daily. B2;LD3;DSK: 700k. B drv changes daily. [Lawrence, KS] Superbrain w/Hayes Smartmodem. General CP/M Software.

South

NACS/UAH RBBS/RCPM, (205) 895-6749, Don Wilkes. CB. 24 hrs. B1;LD1;DSK: 700k [Huntsville, AL] Run for N. Ala. Computer Soc. at U. of Ala.; general CP/M software.

REDSTICK RCPM, (504) 766-8962, Phil Cary, NCB. W'days 2200-1900, w'ends 2200-0900. B1;LD2;DSK: 2.3mb. [Baton Rouge, LA] Message system "REDSTICK" written by sysop. General software. Special Interest in CB-80.

Northern California

OxGate-002 Milpitas, (408) 263-2588, Mel Cruts, CB, 24 hrs. B1;LD2,,3,4;DSK 1.2mb. [South SF Bay Area].

dBASE II RCP/M, (408) 378-8733, Roger D. Brown. NCB. 24 Hours, B2;LD2,3,4; DSK 4mb Hard. [Campbell, Ca] This RCP/M is dedicated strictly to dBASE II users. dBASE II command files are available for downloading. Soon dBASE II will be available to use on system so as to generate software from independent software developers.

Cro'sNEST RCP/M -- DataTech node 004, (408) 732-2433, Robert Kuhman. NCB. 24 hrs. B1;LD2,3,4;DSK: 1mb. [South SF Bay Area] CROMEMCO system two based. Specializing in CP/M, CDOS, and CROMIX software. Many new CDOS programs (never before released to public domain) are available.

Silicon Valley Interchange -- OxGate-004. (408) 732-9190, Ed Svoboda. NCB. 7:45am-11:00pm 7days/week. B2;LD2,3,4;DSK: 2.7mb. [South SF Bay Area] APPLE][CP/M based. Specializing in very recent releases, telecommunications, and Apple CP/M prgms. Note: additional 256k installed.

RCP/M Sunnyvale, (408) 730-8733, Eric Sarti. NCB. 3:30pm-10:00pm 7days/wk. B2;LD2,3,4;DSK: 256k, APPLE][CP/M based. [South SF Bay Area]

OxGate-001 Saratoga, (408) 867-1243, Paul Traina, NCB. 24 hrs. B5;LD2,3,4;DSK: 2.4mb. [South SF Bay Area] Special Interest in latest releases, also functions as west coast "Sysop's Clearinghouse". (OxNet hub)

DataTech Network Headquarters System, (415) 595-0541, Edward Huang. NCB. 24 hrs. B5;LD2,3,4;DSK: 200k. [Box 290, San Carlos, CA 94070 S.F. Bay Area] Heath/Zenith based. Special interest in utilities and communications as well as general software.

Piconet RBBS-RCP/M, (415) 965-4097, Byron McKay. NCB. 24 hrs. B1;LD2,3,4;DSK 2.4mb. Sponsored by PicoNet CP/M group. [SF Bay Area]

RBBS of Marin County, (415) 383-0473, Jim Ayers. NCB. Eves & nites wkdys, 24 hrs wkends. B1;LD2,3,4;DSK: 1mb. [SF Bay Area]

Larkspur RBBS/RCPM, (415) 461-7726, Jim C. NCB. 24 hrs. B1;LD2,3,4;DSK: 2mb. [SF Bay Area]

Napa Valley RBBS/RCPM, (707) 257-6502, Dave Austin. NCB. 24 hrs. B1;LD1;DSK: 600k. [Napa, CA] Supports TRS, Apple, Osborne, Atari and CP/M systems. Also interested in amateur radio and net info.

Southern California

Los Angeles RCP/M, (213) 296-5927, Bob McCown. NCB. 24 hrs. B1;LD2,3,4;DSK: 2.5mb. [West. L.A.] System features catalog of latest CP/M, Apple, Atari, TRS-80 and IBM PC software.

PatVac, (213) 306-3611, 'Pavlov's Cat' (H. B. Edelman), NCB, hours irregular until futher notice. B2;LD2,3,4; DSK: 366kb. [Venice, Ca] RBBS/RCPM format becomes a magazine for computer literate, frequently changing text files on varying topics. Articles, as well as esoteric and arcane software, are encouraged. Running on Osborne 1.43.

Granada Engineering Group RCP/M, (213) 360-5053, Webber Hall. NCB. 24 Hrs. B2;LD2,3,4;DSK: 1mb. [Granada Hills, Ca.] Special interest in CP/M utilities, assembly language programs, hardware/software technical information.

The MOG-UR'S HBBS, (213) 366-1238, Tom Tcimpidis. NCB. 24 hrs. B3;LD2,3,4;DSK: 2mb. [San Fernando valley, LA Area] Disk cap. now 2MB. 450 Baud supported (Bell 103 std). CP/M and HDOS software is rotated on a monthly basis.

G.F.R.N. Data Exchange (RBBS), (213) 541-2503, Skip Hansen. NCB. 24 hrs. B5;LD2,3,4;DSK: 2.4mb. [Palos Verdes, CA] Standard CP/M s'ware with special interest in ham radio-related programs. Soon (with MP/M) will also be reachable thru 450 mhz radio.

Pasadena RBBS, (213) 577-9947, Rich Berg. NCB. 24 hrs. B1/B5;LD2,3,4;DSK: 3.98mb. Heath H89. [L.A. Area] General Interest CP/M Public Domain Software. Note System Power is off until Modem Carrier Lock. (does not recognize CR's for 15 seconds after lock, while System Auto Boots.

HOLLYWOOD RCPM/RBBS. (213) 653-6398. Kim Levitt. NCB. 24 hrs. (Phone not answered when in use, keep trying...) B2; LD2,3,4; DSK: 382k. (Drive B: changed frequently, leave requests for off-line software on RBBS.) [L.A. CA] Special Interests: CP/M utilities, data communications, videographics, applications for entertainment industry. System is Kaypro II with Hayes Smartmodem 300.

Pasadena CBBS, (213) 799-1632, Dick Mead. NCB. 24hrs. B1;LD2,3,4;DSK: 8.3mb Hard. [L.A. Area] Also active as bulletin board. General CP/M software.

BARSTOW RCP/M, (619) 256-3914, Bill Wood. NCB. 24 hrs Mon-Fri, off 0900-1800 Sat/Sun. B5;LD1;DSK: 4.9mb. H89 system also supports 450 and 600 Baud. [Barstow, Ca] General interest CP/M Public Domain Software. Note System Power is off until Modem Carrier Lock. (does not recognize CR's for 15 seconds after lock, while System auto boots.

San Diego RCPM, (619) 273-4354, Brian Kantor. NCB. 24 hrs. B5;LD2,3,4;DSK: 2.4mb. [San Diego CA]

G.F.R.N. Data Exchange (RBBS), Garden Grove, (714) 534-1547, Doug Laing, NCB. 24 hrs, B5;LD2;DSK: 5mb. [Garden Grove Ca.] Special interest in amateur radio and apple/cpm software, also general interest CP/M.

AnaHug RCPM/CBBS, (714) 774-7860, Bob Mathias, John Secor. NCB. 24 hrs. B2;LD2,3,4;DSK: 10mb Hard. [Ahaheim Ca] Special interest in hobby computing, ham, electronics hobbyists.

Thousand Oaks Technical RCP/M, (805) 492-5472, Trevor Marshall. NCB. 24 hrs. B3, (also 600 baud, baud rate set at log on or with NEWBAUD); LD1; DSK: 2mb. [Thousand Oaks CA]

Simi RCP/M, (805) 527-2219, Pete Mack, NCB. 1900-2300 PST, Mon-Fri, 24 hrs on wkends. (300-600 PMMI rates); LD1; DSK: 10mb. [Simi Valley, Ca.] Mostly BDS C programs and occasional new releases of general interest. Disk capacity 10 mb.

CP/M-Net(tm), (805) 527-9321, Kelly Smith. NCB. 1900-2300 (Pacific) Mon-Fri, 1900 Fri to 0700 Mon. B1;LD2;DSK: 20mb Hard. [Simi Valley, CA]

Southwest

Dallas RCP/M CBBS, (214) 931-8274, Dave Crane. NCB. 1800-0800 Mon-Fri, 24 Hrs Sat/Sun/holidays. B1;LD2,3,4;DSK: 2.4mb. [Dallas, Tx] Special interest in programs for and discussions of application of micros to science & engineering, especially earth sciences.

Boulder, Colorado RCPM, (303) 499-9169, Jack Riley. NCB. 1900-2230 wkdays, 1200-2230 weekends. B1;LD2,3;DSK: 32mb Hard. [Boulder, Co]

Colorado Springs RIBBS, (303) 634-1158, Richard Evers (Arvada Electronics). NCB. 24 hrs. B3;LD2,3;DSK: 2.4mb. [Colorado Springs, Co].

Pinecliffe RMP/M RBBS, (303) 642-3034, Craig Baker. NCB. Irregular hrs, 24 hrs. soon, (try anytime). B3;LD2,3;DSK: 16mb. [Pinecliffe, Co] Login by using "LOGIN" program. On-line databases on such topics as nuclear power, Retrieval system, MP/M-II mods, interest in active discussions.

Denver CUG-NODE, (303) 781-4937. ? Sysop. NCB. 24 hrs. B1;LD2,3;DSK: 1mb. [Denver, Co]

Lakewood RCPM/RBBS, (303) 985-1108. Gary Shaffstall. NCB. 24hrs. B3;LD2,3;DSK: 12.86 MB. on A:-D:. [Denver, Co] General interests in CP/M, MP/M, CP/NET, CP/NOS.

Satsuma RCP/M, (713) 469-8893. Charlie Sanborn. NCB. 1400-2400 CST. B3;LD2,3;DSK: 10mb hard. [Houston, TX]. No Message system, either active or planned. Software exchange only. Heath H-8 with Hayes Smartmodem 1200.

El Paso Texas Apple UG RBBS/RCPM, (915) 533-2202. NCB. 24 hrs. B2;LD2,3;DSK: 3.5mb. [El Paso, Tx] Runs on a 3.5Mb segment of North Star Hard Disk (multiuser system.) General, APPLE, and BDS C software.

El Paso Texas RCPM, (915) 598-1668, Sigi Kluger. NCB. 24 hrs. B1(Fri 5pm-Mon 5pm),B3(Mon 5pm-Fri 5pm);LD2,3;DSK:2.7mb. [El Paso, Tx] XMODEM DISKMENU.DQC for list of available files. Diskettes on B: rotated every 2 days.

Northwest

Olympia RCPM, (206) 357-7400. Tim Linehan. NCB. 24 hrs. B1;LD1;DSK: 16mb. Hard. [Olympia, Wa]

RCPM Directory continued . . .

Yelm RBBS & CP/M, (206) 458-3086, Dave Stanhope. CB. 24 hrs. B1;LD1;DSK: 250k. [Olympia, Wa]

Edmonton RCPM, (403) 454-6093, Dave McCrady, NCB. 24 hrs (somewhat sporadic .. not answered when system in use by SYSOP), B5;LD1;DSK: 3.8mb. [Edmonton, Alberta, Canada] General CP/M software;some HDOS, Apple and TRS80 stuff available as well.

Helena Valley RBBS/RCPM, (406) 443-2768. Marion Thompson. NCB. 8am-8pm Monday-Friday, intermittent on wkends. B3;LD1;DSK: 1.2mb. [Helena, Montana] Special Interest in CAI, S-100 and general CP/M software.

Chuck Forsberg's RCPM, (503) 621-3193, NCB. 24 hrs. B5;LD2;DSK: ?. [Oregon]

DOCTOR DOBB'S CP/M EXCHANGE RCPM, (503) 758-8408. Gene Head. CB. 2100-0900 wknites. B2;LD1;DSK: 336k. [Corvallis Or] Interested in helping get new modems up and running, and magazine (DDJ) input from readers.(Letters, articles, listings, etc). People phone (503) 758-0279 0900-2100 daily.

Beaverton, Oregon RCPM, (503) 641-7276, (641-RCPM), Dave Morgan. NCB. 24 hrs. B1;LD2;DSK: 26mb Hard. [Oregon] Interest in very recent releases and computer art.

Frog Hollow CBBS/RCPM, (604) 937-0906, David Bowerman. NCB. 24 hrs. B1;LD1;DSK: 1.2mb. [Vancouver, BC, Canada]

Anchorage RCPM, (AMS), (907) 337-1984, Thomas Hill. NCB. 11pm-9am 7 days/wk. B2;LD1;DSK:12.4mb. Hard. [Anchorage, Alaska] Sysop interested in "just about everything". Has text files on articles written for Lifelines on C: user 6. Voice contact at same phone, 9am to about 7pm.

Overseas

SOFTWARE TOOLS RCPM, 61-2-997-1018 (Australia), Bill Bolton. NCB. 24hrs. 300 baud CCITT V21 standard. LD1; DSK: 4.8Mb. [Sydney, Australia] Special interest in 'C' programs.

MICOM CBBS, 61-3-762-5088 (Australia), Peter Jetson. NCB. 24hrs. 300 baud CCITT V21 standard. LD1; DSK: 500k. [Melbourne, VIC, Australia] (as of Jan'83, CBBS only, no XMODEM)

General North America

CP-MIG. On MicroNet, type 'R CP-MIG' or GO PCS-47, Sysops Dave Kozinn, Tom Jorgenson & Charlie Strom are arranging to have MN carry much of new CPMUG and SIG/M software, plus newsletter and CP/M-oriented CBBS.

Notes

1. Call-back systems are those where computer and real people share same telephone line. To contact people dial & let phone ring until you get answer. To contact computer: (1) dial, (2) let phone ring once, (3) hang up just before 2nd ring, & (4) re-dial. Call back systems are noted as CB, systems not requiring call back are noted as NCB.

2. Baud rates are shown as Bx, where x is a code indicating:
1= PMMI rates (110-710 baud) ; 2 = 300 baud only
3= Bell 212A and 300 baud ; 4 = Vadic 3451 and 300 Baud
5= Bell 212A, Vadic 3451 and 300 Baud.
Note that 212A/Vadic 1200 baud modems may not be compatable with yours. Most of above systems are using Vadic 3451 Triple modems, compatable with both Bell and Vadic Standard. Sign on first time at 300 baud to determine system capabilities. Note also that PMMI's can sometimes be used over 300 baud with 1200 baud systems. PMMI baud rates are: 110, 300, 450, 600, 710.

3. Alternate Long Distance servies are shown as LDx, where x is a code indicating:
1= None 2=Sprint 3=MCI 4=ITT
Whether a program exchange system is accessible by an a.l.d.s. (=alternative long-distance service) should be considered when planning to modem over long programs. Charges on SPRINT, ITT/CITYCALL and MCI are 50-60% of Ma Bell's regular long distance rates.

4. Disk capacity shown for reference. Disks not noted as HARD may be any combination of floppies. Hard disks are generally divided into a number of logical disks. Check system documentation for exact details when logging on.

5. All times listed are local time.

PIP Data Between Computers

A simple way to begin system-to-system communication between two CP/M systems with noncompatible media

by Steven Fisher

PIPIO is a modification to the Peripheral Interchange Program that is supplied as a standard utility for the Digital Research CP/M operating system. This modification implements the special pseudo-devices INP: and OUT: (described in the "Introduction to CP/M" manual), providing a simple way to begin system-to-system data communication between two CP/M computers unable to read each other's media. Many excellent programs exist for this "downloading" purpose, but they must reside on both systems before they can be used (Catch-22).

When the PIP program transfers files, it reads as much as it can into the computer's memory, writes this memory-resident data to its destination, and then repeats the read-write process until the file has been transferred. Continuous transfer of data from one computer to another is not feasible, because when the receiving system pauses to write to its disks, any data still being sent is ignored and therefore lost. Very small files can sometimes be sent without "dropout," but none of the communications programs are that small. While exotic schemes of error detection and recovery exist, they are too complicated to be typed into two computer systems.

What is needed is a simple and effective method of enabling one computer to send data to another at a rate that the receiving computer can handle. This handshaking can be provided by a simple half-duplex protocol: the originating device sends a character and then waits until it receives a character, while the receiving device gets a character and then echoes it back. The operating system provides built-in functions to send a character to either the punch or the console, and to receive a character from either the reader or the console. The PIP program allows insertion of program code to perform the protocol through either the system's reader/punch or console.

To modify your PIP.COM to use this protocol, you must create a disk file named PIPIO.HEX by using an editor. Here is an example using CP/M's "ED" (all operator input—what you type—is underlined):

```
A>ED PIPIO.HEX

NEW FILE

*I

:10010300C30A01C31901000E03CD05003209015FC3

:100113000E04CD0500C9590E04CD05000E03CD050F

:0201230000C911

:0000000000
```

Steven Fisher CDP, Box 457, La Mesa, CA 92041

```
^Z

*E

A>
```

If you need to create a custom PIPIO interface, use the source code in Figure 5 as a guide. Replace the reader and punch system calls with whatever logic is appropriate, such as firmware calls or direct port I/O. Keep in mind that such specialized PIPIO versions are no longer universal or transportable.

Once the PIPIO.HEX file has been created, you must use the Dynamic Debugging Tool (DDT.COM) to insert the HEX file into the version of PIP.COM that is distributed with your operating system. If your computer system does *not* have a reader-punch logical device implemented, you must change four memory locations to use your console for communications as follows:

```
A>DDT PIP.COM

DDT VERS 2.2

NEXT  PC

1E00 0100

(insert reader/punch protocol)

-IPIPIO.HEX

-R

(if DDT responds with a question-mark,
  refer to Figure 1)

NEXT  PC

1E00 0100

(start of modifications to use console)

-S10B

- 03 01

- CD .

-S114

- 04 02

- CD .

-S11B

- 04 02

- CD .
```

```
-S12Ø

- Ø3 Ø1

- CD .

(end of modifications to use console)

-GØØØØ

A>SAVE 29 PIP.COM

A>
```

The modified PIP.COM is now ready to perform system-to-system communications. If you are using serial communications (RS-232C), it may be necessary to modify a cable so that pin 7 on one end connects to pin 7 on the other, while at both ends pin 2 goes to the other end's pin 3 (this is called a "null terminal"). See Figures 2 and 3 for more details.

When you are ready to receive 7-bit data, initiate the receiving computer with the command:

```
A>A:PIP d:filename.typ=INP:
```

(If you are receiving data via the console, disconnect your CRT and plug in the communications cable after PIP sends a linefeed to your terminal.)

This method sends 7-bit ASCII data, so you cannot send 8-bit files or programs without first turning them into HEX files (many word-processing programs, such as MicroPro's WordStar in "document" mode, use the eighth bit.). This conversion is done by the UNLOAD program from the CP/M Users' Group, which creates a file named "d:filename.HEX" from a file named "d:filename.COM". You can change the filetype of the file to be converted (if necessary) by renaming it:

```
A>REN d:filename.COM=d:filename.typ

A>
```

If you do not already have the UNLOAD program, you can create it by following the procedure outlined in Figure 4. When you are ready to create the HEX file from 8-bit data, enter the command:

```
A>UNLOAD d:filename Ø1ØØ

A>
```

When you are ready to transmit 7-bit data, activate the sending computer with the command:

```
A>A:PIP OUT:=d:filename.typ,EOF:
```

(If you are using the console connector to send data, disconnect your CRT and plug in the communications cable after PIP sends a linefeed to your terminal.)

Once the HEX file has been sent to the other computer, it is turned into a file of type COM with the command:

```
A>LOAD d:filename
FIRST ADDRESS Ø1ØØ

LAST  ADDRESS XXXX

BYTES READ    XXXX

RECORDS WRITTEN XX

A>
```

You can then change its type by renaming it:

```
A>REN d:filename.typ=d:filename.COM

A>
```

I first wrote this half-duplex method in March 1978, and have used it without modification on over 20 systems ranging from Apple to Zobex. Although it is rather tiresome if you want to make a career of shuttling data between computers, it is a simple, inexpensive way any CP/M user can "download."

Figure 1. Installing PIPIO by hand

Some early versions of Digital Research's Dynamic Debugging Tool (DDT.COM) do not allow reading a HEX file onto an existing program. If your DDT is one of these, it will display a question mark when you try to insert the PIPIO.HEX file into the version of PIP.COM that is distributed with your operating system. By following the procedure outlined here, you will be able to put the PIPIO protocol into PIP, but you will have to be very careful to avoid typographical errors. If your operating system does not have a reader-punch logical device implemented, you must also change four memory locations to use your console for communications:

```
A>DDT PIP.COM

DDT VERS 2.2

NEXT  PC

1EØØ Ø1ØØ

(insert PIPIO protocol)

- 5Ø C3

- 3A ØA

- 2F Ø1

- 4F C3

- 55 19

- 54 Ø1

- 3A ØØ

- 53 ØE

- 5Ø Ø3

- 41 CD

- 43 Ø5

- 45 ØØ
```

What is needed is a simple and effective method of enabling one computer to send data to another at a mutually acceptable rate. This handshaking can be provided by a simple half-duplex protocol.

```
- 29 32
- 28 Ø9
- 49 Ø1
- 4E 5F
- 5Ø ØE
- 3A Ø4
- 2F CD
- 4F Ø5
- 55 ØØ
- 54 C9
- 3A 59
- 53 ØE
- 5Ø Ø4
- 41 CD
- 43 Ø5
- 45 ØØ
- 29 ØE
- 28 Ø3
- 49 CD
- 4E Ø5
- 5Ø ØØ
- 3A C9
- 2F .
(start of modifications to use console)
-S1ØB
- Ø3 Ø1
- CD .
-S114
- Ø4 Ø2
- CD .
-S11B
- Ø4 Ø2
- CD .
-S12Ø
- Ø3 Ø1
- CD .
(end of modifications to use console)
-GØØØØ
A>SAVE 29 PIP.COM
A>
```

Figure 2. Null terminal wiring

The RS-232 standard for serial communications interfaces stipulates that on every line there are two devices, the Data Communications Equipment (DCE—computer or modem) and the Data Terminal Equipment (DTE—console or printer). Thus a "modem" end of the line only talks to a "terminal" end. The objective of the "null terminal" is to serve as a bridge between two modems. This is done by switching the two data lines. Some computers require other RS-232 lines to be active for "handshaking" control. The null terminal therefore loops the basic handshaking lines back to their source, causing each computer to keep active the serial lines it requires.

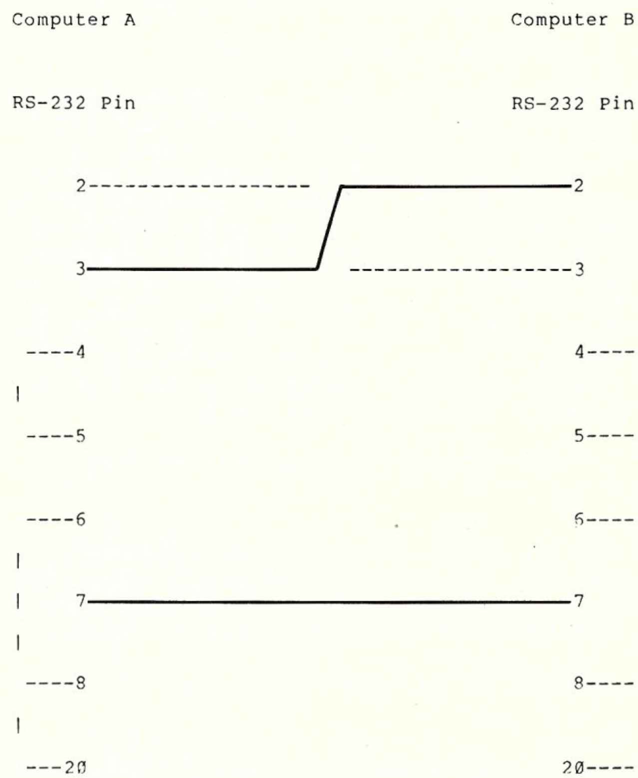


Figure 3. RS-232 configurations of microcomputers

The null terminal depicted in Figure 2 is necessary to allow communications between similarly configured ends of an RS-232 serial line. With dissimilar ends, only a normal cable is needed. If a computer plugs into a modem, it is wired as a terminal (DTE). Conversely, if you use your serial printer line for communications, it is wired as a modem (DCE). The list below represents a partial inventory of those systems on which I have used PIPIO successfully:

I have used this half-duplex method without modification on over 20 systems—from Apple to Zobex. It is a simple, inexpensive way any CP/M user can "download."

Computer Manufacturer	Type
Apple II with Apple Serial Card	DCE
Apple II with CCS Serial Card	DCE
California Computer Systems	DCE
CompuPro with Interfacer II	DCE/DTE
Cromemco System 3	DCE
Delta Products	DCE
Digital Group with CP/M	DCE
Epic Episode	DCE/DTE
IMSAI VDP-80	DCE
Jonos Escort	DCE
KayPro II	DTE
Morrow Discus II with TEI 3P&3S	DCE
MicroMation Mariner	DCE
NorthStar Advantage/Horizon	DCE
Osborne 1	DCE
Otrona Attache	DCE
TRS-80 Model II (special cable on A)	DTE
Vector Graphics MZ	DCE
Xerox 820 (using monitor firmware)	DTE
Zenith Z89	DCE
Zobex	DCE

Figure 4. Creating the UNLOAD program

The UNLOAD utility translates 8-bit binary data into a printable 7-bit form called the Intel HEX format. If you are unable to obtain a disk-resident copy of the UNLOAD utility from the CP/M Users' Group, you can create it by using your editor program to make the HEX version of the program:

```
A>ED UNLOAD.HEX

NEW FILE

*I

:100100002A06002BF9C311017E1223130DC2080128

:10011000C9215C00111F010E09CD0801C346010071

:100120000000000000000000434F4D0000000000F0
```

```
:1ØØ13ØØØØØØØØØØØØØØØØØØØØØØØØØØØØØØØØØØØØØØØBF
:1ØØ14ØØØEDØ3ØØØ4ØØØØC3B6Ø12A42Ø1EB2A44Ø17A
:1ØØ15ØØØ7D937C9ADA9FØ121ØØØØ2244Ø1EB2A422Ø
:1ØØ16ØØØØ17B957A9CD291Ø12A4ØØ119EBØE1ACDAØ
:1ØØ17ØØØØ5ØØ111FØ1ØE14CDØ5ØØB7C28BØ1118ØBF
:1ØØ18ØØØØØ2A44Ø119224401C35DØ12A44Ø122428C
:1ØØ19ØØØØ1118ØØØØE1ACDØ5ØØ21ØØØØ2244Ø1EB6Ø
:1ØØ1AØØØ2A4ØØ119EB2A42Ø17DB43E1AC81A2A449A
:1ØØ1BØØØØ123224401C9AF322BØ1323FØ121ØØØ447
:1ØØ1CØØØ2242Ø122440 1ØEØF111FØ1CDØ5ØØ3CC245
:1ØØ1DØØØEDØ1ØEØ911DDØ1CDØ5ØØC3ØØØØØDØA4E31
:1ØØ1EØØØ4F2Ø4946494C452Ø46494C4524215CØØ56
:1ØØ1FØØØ11FBØ1ØEØ9CDØ8Ø1C322Ø2ØØØØØØØØØØ1E
:1ØØ2ØØØØØØØØØØØØ484558ØØØØØØØØØØØØØØØØØØØ9
:1ØØ21ØØØØØØØØØØØØØØØØØØØØØØØØØØØØEDØ7ØØØ4E6
:1ØØ22ØØØØØØØC3A5Ø2F52A1EØ2EB2A2ØØ27D937C62
:1ØØ23ØØØ9ADA95Ø221ØØØØ222ØØ2EB2A1EØ27B95Ø9
:1ØØ24ØØØ7A9CD287Ø22A1CØ219EBØE1ACDØ5ØØ11E6
:1ØØ25ØØØFBØ1ØE15CDØ5ØØB7C268Ø2118ØØØ2A2ØEF
:1ØØ26ØØØØ219222ØØ2C33AØ2ØEØ91174Ø2CDØ5ØØCØ
:1ØØ27ØØØF1C3ØØØØØDØA4449534B2Ø46554C4C3AFB
:1ØØ28ØØØ2Ø4F46494C4524118ØØØØE1ACDØ5ØØ21ØF
:1ØØ29ØØØØØØØØ222ØØ2EB2A1CØ219EBF1122A2ØØ294
:1ØØ2AØØØ23222ØØ2C9AF32Ø7Ø2321BØ221ØØØ4229E
:1ØØ2BØØØ1EØ221ØØØØ222ØØ2ØE1311FBØ1CDØ5ØØB9
:1ØØ2CØØØØE1611FBØ1CDØ5ØØ3CC2EDØ2ØEØ911D73F
:1ØØ2DØØØØ2CDØ5ØØC3ØØØØØDØA4E4F2Ø4449522ØB4
:1ØØ2EØØØ535Ø4143453A2Ø4F46494C452421ØØØ193
:1ØØ2FØØØØ6ØØ116DØØ1A13D63ØDA14Ø3FEØADAØB69
:1ØØ3ØØØØ3D6Ø7DA14Ø3FE1ØD214Ø3292929294F32
:1ØØ31ØØØ9C3F5Ø222EAØ3CD49Ø1CA8AØ3F53E3A3Ø
:1ØØ32ØØØCD25Ø2AF32ECØ33E1ØCD4EØ33AEBØ3CDA8
:1ØØ33ØØØ4EØ33AEAØ3CD4EØ3AFCD4EØ3F1Ø61ØC58E
:1ØØ34ØØØCD4EØ3C1Ø5CA7ØØ3CD49Ø1C33FØ34F3AE7
```

```
:1ØØ35ØØØECØ39132ECØ379ØFØFØFØFCD5FØ379E6B9
:1ØØ36ØØØØFC63ØFE3ADA6AØ3C6Ø7C5CD25Ø2C1C9F9
:1ØØ37ØØØ3AECØ3CD4EØ33EØDCD25Ø23EØACD25Ø2BB
:1ØØ38ØØØ2AEAØ3111ØØØ19C314Ø33E3ACD25Ø2Ø6DØ
:1ØØ39ØØØØ5AFC5CD4EØ3C1Ø5C291Ø33EØDCD25Ø26B
:1ØØ3AØØØ3EØACD25Ø22A2ØØ27DE67FC2B1Ø3221E2D
:1ØØ3BØØØØ23E1AF5CD25Ø2F1C2A5Ø3ØE1Ø11FBØ174
:1ØØ3CØØØCDØ5ØØ3CC2E7Ø3ØEØ911D2Ø3CDØ5ØØC3E1
:1ØØ3DØØØE7Ø3ØDØA43414E4E4F542Ø434C4F5345C3
:1ØØ3EØØØ2Ø4F46494C4524C3ØØØØØØØØØØØØØØØØ97
:1ØØ3FØØØØØØØØØØØØØØØØØØØØØØØØØØØØØØØØØØØFD
:ØØØØØØØØØØ
^Z
*E
A>
```

Now you turn the HEX into a runnable program with the standard Digital Research LOAD utility program:

```
A>LOAD UNLOAD
FIRST ADDRESS Ø1ØØ
LAST  ADDRESS Ø3FF
BYTES READ    Ø3ØØ
RECORDS WRITTEN Ø6
A>
```

Figure 5. PIPIO assembly language source code.

```
;*                    I/O PATCH FOR PIP.COM
;*  THIS PATCH IS USED FOR IMPLEMENTING HALF-DUPLEX 7-BIT
;*  RDR:/PUN: HANDSHAKING IN COMPUTER-TO-COMPUTER COMM.
;*  DONATED TO PUBLIC DOMAIN FOR NON-COMMERCIAL USE
;*  STEVEN FISHER CONTROLLED INFORMATION ENVIRONMENTS 3/3/78
;*
;*      SOURCE SYNTAX IS:       PIP OUT:=d:filename.typ,EOF:
;*
;*      DESTINATION SYNTAX IS:  PIP d:filename.typ=INP:
;*
;*      INSTALLATION SYNTAX IS: DDT PIP.COM
```

```
;*                          IPIPIO.HEX
;*                          R
;*                          G0000
;*                          SAVE 29 PIP.COM

        ORG     103H        ;PUT THIS IN PIP PATCH AREA
;
RDRF    EQU     03H         ;READER RETURNS CHAR IN A
PUNF    EQU     04H         ;PUNCH SENDS CHAR FROM E
BDOS    EQU     0005H       ;SYSTEM ENTRY POINT
                            ;IF YOUR SYSTEM HAS NO
                            ;RDR: OR PUN:, YOU MAY NEED
                            ;TO USE FIRMWARE MONITOR OR
                            ;DO DIRECT PORT I/O.

RCVJP:  JMP     RCV         ;PIP INP: ENTRY POINT
XMTJP:  JMP     XMT         ;PIP OUT: ENTRY POINT
;
RCVDTA: DB      0           ;RECEIVED DATA
;
RCV:    MVI     C,RDRF
        CALL    BDOS        ;READ INTO A, MAYBE MONITOR
        STA     RCVDTA      ;SAVE IT
        MOV     E,A
        MVI     C,PUNF
        CALL    BDOS        ;PUNCH FROM E, MAYBE MONITOR
        RET

XMT:    MOV     E,C         ;CHAR TO SEND GOES FROM E
        MVI     C,PUNF
        CALL    BDOS        ;PUNCH FROM E, MAYBE MONITOR
        MVI     C,RDRF
        CALL    BDOS        ;READ INTO A, MAYBE MONITOR
        RET

        END                 ;END OF PIPIO SOURCE
```

Steven Fisher began his career in data processing in 1969 as a Cobol programmer in a computer service bureau. He formed his own consulting firm in 1977, creating custom software and performing systems integration for small businesses. In 1981, he designed and wrote SUPERVYZ, the first menu-driven "frontend" to the CP/M operating system for the nontechnical user. Steven Fisher is presently affiliated with a microelectronics research and development group, and also independently creates software aimed at putting people in control of their computers.

Toward Smarter Modem Programs

A simple language for writing protocols that interact with remote computers

by Walt Bilofsky

Programmers only do things twice. The third time, they write a program for it.

Dialing up and logging on to a remote computer is one such task—something that many programmers perform often. Existing programs can automate it to some degree. For example, some modem programs can transmit predefined strings containing pauses of specified lengths; this is often enough to let the program reply to the user name and password prompts, and so log you on automatically. With the help of an "intelligent modem" capable of accepting dialing commands, these programs can even dial the remote system.

But when I discovered several popular local bulletin boards, I found myself repeatedly dialing each system in turn in order to get past the almost constant busy signals. Clearly, a more sophisticated solution was needed.

The answer came as I recalled Ben Wegbreit's first step for solving programming assignments when he was a graduate computer science student. He would first design a programming language in which to write the solution. (The other steps were to write the solution in the new language, and then to hand-translate it into Fortran, which the instructor could understand. Ben later become a professor at Harvard and invented an extensible programming language, which eliminated the last step. But I digress.)

The solution to the dial-up problem was to design a simple language for writing little programs, called *protocols,* that interact with remote computers. For instance, I have written a protocol that repeatedly dials each of four bulletin boards in turn until it finds one which is not busy, and then logs me in.

This protocol language is built into REACH II, a modem program available for the Heath/Zenith H89/Z89/Z90 computers from The Software Toolworks. Hence, protocol examples given below relate to the capabilities of the REACH II program. But a similar facility could be installed in most any modem program, and the protocol approach can be applied to other tasks of a different nature.

The protocol language

The key to the power of the protocol language is that it can inspect incoming text for particular strings, and take action depending on whether or not that string has been received. This "test and branch" capability is quite simple, yet it is all you need to program fairly sophisticated remote dial-up tasks.

For example, modem programs that offer only a pause capability in automatic log-on require you to guess the longest length of time required for the remote computer to type out the next prompt. Using this protocol language, you simply wait for the prompt!

Walt Bilofsky, The Software Toolworks, 15233 Ventura Blvd., Suite 1118, Sherman Oaks, CA 91403

Now let's look at the protocol language in detail. It contains five statement types, each beginning with a unique character. The statement types are:

Label
Transmit
Match
Timeout
Comment

Each statement occupies a single line. The statements are entered into a file using any text editor.

The *label* statement defines a label in the file, so that other statements can "goto" that statement to continue execution. A label statement consists of a colon followed by the label name:

```
:retry
```

The *transmit* statement transmits a string over the modem. It consists of the character > followed by the string to be transmitted:

```
>Send this string.\r
```

The "/r" means the Return control character. Control characters are inserted in strings using the following conventions:

\n	means	Line Feed (0A hex)
\r	means	Return (0D hex)
\\	means	\
\:	means	:
\C	means	Ctrl-C (where C is any character but n, r, \ or :)

The *match* statement is the one with all the "moxie." It looks to see if a particular string has been received from the modem. If not, command execution continues with the next statement. If the string has been received, a jump to a specified label is executed. Also, the internal buffer is cleared so that no future match can be made to previously received data. The match statement consists of the character "=", the string to be matched, a ":", and the (optional) label "goto" if a match is found. For example,

```
=User\::sendname
```

will see if "User:" has been received, and transfer to the label "sendname" if it has.

A string containing no characters will always match. Thus, a simple "goto" consists of a match statement with no string:

```
=:gothere
```

The *timeout* statement initiates an action if nothing has happened for a while. It consists of the letter "T", a number, a ":", and an (optional) label. Its effect is to transfer to the label if nothing has been sent or received for the specified number of seconds. For example,

T30:hangup

will transfer to the label "hangup" if nothing has been received or sent in the past 30 seconds.

If the label is omitted in a match or timeout statement, the statement will simply wait until the specified condition has been satisfied, and then continue execution. Thus,

T20:

pauses 20 seconds, while

=OK:

waits until the string "OK" has been received from the modem.

The last statement type is the *comment*. This is simply a ";" followed by anything. When a comment is executed, the statement is displayed on the terminal, but nothing else happens.

A remote log-on protocol

As an example, here's a simple protocol for logging-on to MicroNet, a typical remote timesharing system. The statements are shown on the left, and the effect on the right.

>\c	Type ctrl-C to get host's attention
=User\::	Wait for "User:" prompt
>70000,100\r	Enter our user ID
=Password\::	Wait for "Password:" prompt
>jabberwocky\r	Enter our password

If this protocol has been typed into a file called LOGON, I can just dial up the remote system and give the CP/M command

REACH LOGON

and this file will be invoked and will log me in automatically.

The command file finishes its execution by "dropping off the end." This puts the modem program into its normal mode of operation, in which the computer acts like a dumb terminal connected to the remote host. In addition, data can be transmitted from a disk file, and received text can be stored in a disk file or printed directly on the printer. And even while the command file is being executed, anything typed on the keyboard is sent to the remote host, allowing you to intervene if, for example, line noise garbles the response that the command file is waiting for.

Automatic dialing

It helps to have a modem such as the Hayes SmartModem, which can dial automatically. It has many other features as well, but the dialing feature is the only one I need to describe here.

When you first turn it on, the SmartModem takes ASCII commands from the computer and sends back an ASCII string indicating the result. The command

"ATD", followed by a number, tells the SmartModem to dial that telephone number. When a remote host has been reached, the SmartModem drops out of command mode, and your computer is connected to the remote host. If no carrier is detected within about 30 seconds, the SmartModem hangs up and reports failure. For example, if the command

```
ATD (213) 555-1212
```

is sent to the SmartModem, it will take the phone line off the hook and dial (213) 555-1212. (It ignores all punctuation in the phone number.) If it detects a carrier from an answering modem, the SmartModem sends back

```
CONNECT
```

If the line is busy, or for some other reason there is no carrier detected in 30 seconds, the SmartModem sends back

```
NO CARRIER
```

The following protocol uses this autodial feature to continue dialing a number until a carrier is detected.

```
:redial
>ATD5551212             Dial the number.
:wait
=CONNECT:logon          If it answers, go log on.
=NO CARRIER:redial      If it doesn't, go try
                          again.
=:wait                  Loop, waiting for either
                          result.
:logon
>\c                     . . . and continue with
  :                       the logon protocol for
                          that system.
  :
```

"BBoard Cruiser Special" protocol

It's easy to expand the previous example into my BBoard Cruiser Special. I crank it up and walk away, while it keeps dialing each of my local bulletin boards in turn until it gets through to one. Then it beeps a few times to call me back to the machine. Listing 1 is a simplified example for dialing two BBoards.

This protocol can be expanded to dial any number of bulletin boards. I use it for up to four. Using the techniques presented previously, it can even be made to log into each one with a different user ID and password before it calls me back from whatever else I am doing.

Summary

Dialing and logging into remote systems is a repetitive task that can be automated given the right tools: the protocol language described above and an optional autodial modem. The language is powerful enough to program redialing until it gets through, and even to dial multiple systems until one is reached. This capability is available to owners of the Heath/Zenith H89/Z89 as part of the REACH II modem program

Modem Programs continued . . .

from The Software Toolworks, and a similar language could be incorporated into almost any modem program. ▣

Listing 1

```
:board1
;Dialing Board 1.        Tell user what's happening.
>AT D5551212             Dial the number.
:wait1
=CONNECT:callWalt
                         If it answers, go log on.
=NO CARRIER:try2
                         If it doesn't, go try Board 2.
=:wait1                  Loop, waiting for either
                            result.

:board2
;Dialing Board 2.        Tell user what's happening.
>ATD9361212              Dial the number.
:wait2
=CONNECT:callWalt
                         If it answers, go log on.
=NO CARRIER:try1
                         If not, go try Board 1 again.
=:wait2                  Loop, waiting for either
                            result.

:callWalt                Comes here if one is reached.
;\g                      Send a bell (ctrl-G).
T1:                      Pause 1 second.
;\g                      Send another and drop into
                            terminal mode.
```

Put Your Printer on a Parallel Port

Interface the Epson MX-80 printer to the North Star Horizon and other computers via a parallel port

by Ted Carnevale

North Star's most recent version of CP/M 2.2 (revision 1.1.0) offers some very useful features. First of all, disk files are allocated in multiples of 1K instead of 2K. As a result, all your short programs and subroutines take up less disk space, freeing up room for handy utilities. The COPY utility can be used to duplicate damaged disks, using ^A to force reading of bad sectors, thus allowing retrieval of whatever data remains intact. And DIRDUMP will list the sectors occupied by all files on a disk, so you can go back with North Star's DOS and read individual sectors belonging to any CP/M file!

What really convinced me to get North Star's CP/M was the fact that it can relocate itself "around" the disk controller and floating-point board, which are memory-mapped into 0E800-0EFFFH. Previously, North Star users had to settle for 58K CP/M. This restricted the program and data areas available to memory-hungry applications like the MUMPS or Pascal interpreters.

The problem

If you get North Star's CP/M, you may want to make one change in the BIOS. Certain parallel printers, such as the Epson MX-80 with Graftrax option, require data transmission along eight lines for full utilization of special features such as graphic or alternate character sets. Any handshaking must be relegated to separate control lines. However, documentation for North Star's DOS and their version of CP/M made it clear that their parallel printer routines used only seven of the data lines for data transmission, toggling the eighth line as a strobe to tell the printer that the data was valid. While seven-bit ASCII code will do for alphanumeric communication, using the Epson's dot-addressable graphics would be restricted and awkward.

The MX-80's parallel port

Although this discussion refers specifically to the MX-80, it applies equally well to any other printer that has a Centronics-compatible parallel interface. The pinout table at the end of the Epson's manual lists 21 different signals appearing on the printer's 36-pin connector. The printer is "enabled" (i.e., allowed to receive data) by tying pin 36 (SLCT IN*) to logic ground (pin 16). Pins 2–9 are data lines for bits 1–8.

Various control signals appear on several pins, but simple "handshaking" requires only two of these. The computer has to pull pin 1 (STROBE*) low for at least 0.5 μsec to let the printer know that valid data is present. The printer in turn acknowledges receipt of each character by pulling pin 10 (ACKNLG*) low.

Ted Carnevale, M.D., Neurology Dept., SUNY, HSC T12, Rm. 020, Stony Brook, NY 11794

The Horizon's parallel output port

The schematics in the back of the North Star *Computer System Manual* show that the Horizon's motherboard has a parallel output port with eight latched data lines and several control lines. Depending on the configuration of the parallel port header, one of these lines (pin 8, labelled SPARE*) is pulled low each time data is output. This line can be used as a strobe to signal the printer that new data is ready.

A pulse applied to pin 7 (ACK*) sets the output of a flip-flop high (the "parallel output flag," P-O-FLG). This flag can be set and reset under software control. Its status can be checked to determine whether the printer is ready for data to be sent.

The hardware solution

I built a cable with the connections listed in Table 1.: Horizon's SPARE* drives the MX-80's STROBE* input, and the MX-80's ACKNLG* output is tied back to the Horizon's ACK* pin. Now, how should

Table 1. Cable connections

Function	Horizon	MX-80 signal/return (see notes)
Data bit 1	5	2/20
2	12	3/21
3	4	4/22
4	11	5/23
5	10	6/24
6	2	7/25
7	9	8/26
8	1	9/27
STROBE* or SPARE*	8	1/19
ACKNLG*	7	10/28
SLCT IN*	—	36—tie to LOGIC 0
LOGIC 0	—	16
GND	3	33

1. a/b means a twisted pair with signal wire to pin a, ground wire ("return") to pin b.

2. Connectors are available from many hardware vendors. INMAC (2465 Augustine Drive, Santa Clara, CA 95051) sells these as #865NB for the Horizon (15-pin male D subminiature) and #504NB for the Epson MX-80 printer (36-pin female Centronics style).

Z80 Software

DEBUG QED LINKER MODEM MACRO TPM BASIC I ZDDT ZTEL QSAL ZEDIT TOP I 6502X TPM II

SOFTWARE DESCRIPTIONS

TPM (TPM I) - $80 A Z80 only operating system which is capable of running CP/M programs. Includes many features not found in CP/M such as independent disk directory partitioning for up to 255 user partitions, space, time and version commands, date and time, create FCB, chain program, direct disk I/O, abbreviated commands and more! Available for North Star (either single or double density), TRS—80 Model I (offset 4200H) or II, Versafloppy I, or Tarbell I.

TPM-II - $125 An expanded version of TPM which is fully CP/M 2.2 compatible but still retains the extra features our customers have come to depend on. This version is super FAST. Extended density capability allows over 600K per side on an 8" disk. Available preconfigured for Versafloppy II (8" or 5"), Epson QX-10, Osborne II or TRS-80 Model II.

CONFIGURATOR I

This package provides all the necessary programs for customizing TPM for a floppy controller which we do not support. We suggest ordering this on single density (8SD).
Includes: TPM-II ($125), Sample PIOS (BIOS) SOURCE ($FREE), MACRO II ($100), LINKER ($80), DEBUG I ($80), QED ($150), ZEDIT ($50), TOP I ($80), BASIC I ($50) and BASIC II ($100)
$815 Value — **NOW $250**

CONFIGURATOR II

Includes: TPM-II ($125), Sample PIOS (BIOS) SOURCE ($FREE), MACRO II ($100), MACRO III ($150), LINKER ($80), DEBUG I ($80), DEBUG II ($100), QSAL ($200), QED ($150), ZTEL ($80), TOP II ($100), BUSINESS BASIC ($200) and MODEM SOURCE ($40) and DISASSEMBLER ($80)
$1485 Value — **NOW $400**

MODEL I PROGRAMMER

This package is only for the TRS-80 Model I. Note: These are the ONLY CDL programs available for the Model I. It includes: TPM I ($80), BUSINESS BASIC ($200), MACRO I ($80), DEBUG I ($80), ZDDT ($40), ZTEL ($80), TOP I ($80) and MODEM ($40)
$680 Value — **NOW $175**

MODEL II PROGRAMMER

This package is only for the TRS-80 Model II. It includes: TPM-II ($125), BUSINESS BASIC ($200), MACRO II ($100), MACRO III ($150), LINKER ($80), DEBUG I ($80), DEBUG II ($100), QED ($150), ZTEL ($80), TOP II ($100), ZDDT ($40), ZAPPLE SOURCE ($80), MODEM ($40), MODEM SOURCE ($40) and DISASSEMBLER ($80)
$1445 Value — **NOW $375**

BASIC I - $50, a 12K+ basic interpreter with 7 digit precision.

BASIC II - $100, A 12 digit precision version of Basic I.

BUSINESS BASIC - $200, A full disk extended basic with random or sequential disk file handling and 12 digit precision (even for TRIG functions). Also includes PRIVACY command to protect source code, fixed and variable record lengths, simultaneous access to multiple disk files, global editing, and more!

ACCOUNTING PACKAGE - $300, Written in Business Basic. Includes General Ledger, Accounts Receivable/Payable, and Payroll. Set up for Hazeltine 1500 terminal. Minor modifications needed for other terminals. Provided in unprotected source form.

MACRO I - $80, A Z80/8080 assembler which uses CDL/TDL mnemonics. Handles MACROs and generates relocateable code. Includes 14 conditionals, 16 listing controls, 54 pseudo-ops, 11 arithmetic/logical ops, local and global symbols, linkable module generation, and more!

MACRO II - $100, An improved version of Macro I with expanded linking capabilities and more listing options. Also internal code has been greatly improved for faster more reliable operation.

MACRO III - $150, An enchanced version of Macro II. Internal buffers have been increased to achieve a significant improvement in speed of assembly. Additional features include line numbers, cross reference, compressed PRN files, form feeds, page parity, additional pseudo-ops, internal setting of time and date, and expanded assembly-time data entry.

DEVELOPER I

Includes: MACRO I ($80), DEBUG I ($80), ZEDIT ($50), TOP I ($80), BASIC I ($50) and BASIC II ($100)
$440 Value — **NOW $150**

DEVELOPER II

Includes: MACRO II ($100), MACRO III ($150), LINKER ($80), DEBUG I ($80), DEBUG II ($100), BUSINESS BASIC ($200), QED ($150), TOP II ($100), ZDDT ($40), ZAPPLE SOURCE ($80), MODEM SOURCE ($40), ZTEL ($80), and DISASSEMBLER ($80).
$1280 Value — **NOW $350**

DEVELOPER III

Includes: QSAL ($200), QED ($150), BUSINESS BASIC ($200), ZTEL ($80) and TOP II ($100)
$730 Value — **NOW $300**

COMBO

Includes: DEVELOPER II ($1280), ACCOUNTING PACKAGE ($300), QSAL ($200) and 6502X ($150)
$1930 Value — **NOW $500**

LINKER - $80. A linking loader for handling the linkable modules created by the above assemblers.

DEBUG I - $80. A tool for debugging Z80 or 8080 code. Disassembles to CDL/TDL mnemonics compatible with above assemblers. Traces code even through ROM. Commands include Calculate, Display, Examine, Fill, Goto, List, Mode, Open File, Put, Set Wait, Trace, and Search.

DEBUG II - $100. A superset of Debug I. Adds Instruction Interpreter, Radix change, Set Trap/Conditional display, Trace options, and Zap FCB.

6502X - $150. A 6502 cross assembler. Runs on the Z80 but assembles 6502 instructions into 6502 object code! Similar features as our Macro assemblers.

QSAL - $200. A SUPER FAST Z80 assembler. Up to 10 times faster than conventional assemblers. Directly generates code into memory in one pass but also to offset for execution in its own memory space. Pascal like structures: repeat...until, if...then...else, while...do, begin...end, case...of. Multiple statements per line, special register handling expressions, long symbol names, auto and modular assembly, and more! This one uses ZILOG Mnemonics.

QED - $150. A screen editor which is both FAST and easy to learn. Commands include block delete, copy, and move to a named file or within text, repeat previous command, change, locate, find at start of line, and numerous cursor and window movement functions. Works with any CRT having clear screen, addressable cursor, clear to end of line, clear to end of screen, and 80X24.

ZTEL - $80. An extensive text editing language and editor modelled after DEC's TECO.

ZEDIT - $50. A mini text editor. Character/line oriented. Works well with hardcopy terminals and is easy to use. Includes macro command capability.

TOP I - $80. A Text Output Processor for formatting manuals, documents, etc. Interprets commands which are entered into the text by an editor. Commands include justify, page number, heading, subheading, centering, and more.

TOP II - $100. A superset of TOP I. Adds: embedded control characters in the file, page at a time printing, selected portion printing, include/merge files, form feed/CRLF option for paging, instant start up, and final page ejection.

ZDDT - $40. This is the disk version of our famous Zapple monitor. It will also load hex and relocatable files.

ZAPPLE SOURCE - $80. This is the source to the SMB ROM version of our famous Zapple monitor. It can be used to create your own custom version or as an example of the features of our assemblers. Must be assembled using one of our assemblers.

MODEM - A communication program for file transfer between systems or using a system as a terminal. Based on the user group version but modified to work with our SMB board or TRS-80 Models I or II. You must specify which version you want.

MODEM SOURCE - $40. For making your own custom version. Requires one of our Macro Assemblers.

DISASSEMBLER - $80. Does bulk disassembly of object files creating source files which can be assembled by one of our assemblers.

HARDWARE

S-100 — SMB II Bare Board $50. "System Monitor Board" for S-100 systems, 2 serial ports, 2 parallel ports, cassette interface, 4K memory (ROM, 2708 EPROM, 2114 RAM), and power on jump. When used with Zapple ROM below, it makes putting a S-100 system together a snap.
Zapple ROM $35. Properly initializes SMB I/II hardware, provides a powerful debug monitor.

IBM PC — Big Blue Z80 board $595. Add Z80 capability to your IBM Personal Computer. Runs CP/M programs but does not require CP/M or TPM. Complete with Z80 CPU, 64K add on memory, serial port, parallel port, time and date clock with battery backup, hard disk interface, and software to attach to PC DOS and transfer programs. Mfr'd by QCS.
50% Discount on all CDL software ordered at the same time as a Big Blue (and for the Big Blue).

APPLE II — Chairman Z80 $345. Add Z80 capability to your Apple II/II Plus computer. Runs CP/M programs with our more powerful TPM. Includes 64K memory add on (unlike the competition this is also useable by the 6502/DOS as well as the Z80), TPM, QSAL assembler, QED Screen Editor, and Business Basic. Mfr'd by AMT Research.
Apple Special $175. Buy the Apple Z80 Developer at the same time as the "Chairman" and pay only $175 instead of $325.

APPLE Z80 DEVELOPER

Includes: 6502X ($150), MACRO II ($100), MACRO III ($150), QSAL ($200), QED ($150), LINKER ($80), DEBUG I ($80), DEBUG II ($100), ZDDT ($40) and BUSINESS BASIC ($200)
VALUE: $1250 — **NOW $325**
$175 when purchased with AMT "Chairman" Board

DISK FORMATS

When ordering software specify which disk format you would like.

CODE	DESCRIPTION
8SD	8" IBM 3740 Single Density (128 bytes/26 sectors/77 tracks)
8DD	8" Double Density (256 bytes/26 sectors/77 tracks)
8XD	8" CDL Extended Density (1024 bytes/8 sector/77 traceks 616K)
5SD	5.25" Single Density (TRS80 Model I, Versafloppy I, Tarbell I)
5EP	5.25" Epson Double Density
5PC	5.25" IBM PC Double Density
5XE	5.25" Xerox 820 Single Density
5OS	5.25" Osborne Single Density
5ZA	5.25" Z80 Apple (Softcard compatible)

TPM INFO When ordering TPM I or II, in addition to Disk Format, please specify one of the following codes:

	CODE	DESCRIPTION
TPM I:	NSSD/H	North Star Single Density for Horizon I/O
	NSSD/Z	North Star Single Density for Zapple I/O
	NSDD/H	North Star Double Density for Horizon I/O
	NSDD/Z	North Star Double Density for Zapple I/O
	TRS80-I	TRS-80 Model I (4200H Offset)
	TRS80II	TRS-80 Model II
	VI8	Versafloppy I 8"
	VI5	Versafloppy I 5.25"
TPM-II:	VII8	Versafloppy II 8" (XD)
	VII5	Versafloppy II 5.25"
	TRS80II	TRS-80 Model II (XD)

ORDERING INFORMATION:

VISA/MasterCard/C.O.D.
Call or Write With Ordering Information....

MasterCard VISA

OEMS:

Many CDL products are available for licensing to OEM's. Write to Carl Galletti with your requirements.

Dealer Inquiries Invited.

For Phone Orders ONLY Call Toll Free...
1-(800) 458-3491 (Except Pa.)
Ask For Extension #15

For information and Tech Queries call
(609) 599-2146

Prices and Specifications subject to change without notice.
TPM, Z80, CP/M, TRS80 are trademarks of CDL, Zilog, DRI and Tandy respectively.

Computer Design Labs

342 Columbus Avenue/Trenton, NJ 08629

CIRCLE 84 ON READER SERVICE CARD

```
                   ;******************************************
                   ;
                   ;PARALLEL OUTPUT DRIVERS FOR THE NORTHSTAR HORIZON
                   ;          N.T.CARNEVALE 6/82
                   ;
                   ;******************************************
                   ;
                   ;THESE EQUATES ARE FOR BOTH CP/M AND NS DOS
                   ;
0000 =             PAROUT  EQU  0      ;PARALLEL OUTPUT PORT (FOR DATA OUTPUT)
0006 =             MOTHER  EQU  6      ;MOTHERBOARD PORT (FOR POFLG STATUS & RESET)
                   ;
0020 =             PORST   EQU  20H    ;COMMAND TO RESET THE PARALLEL OUTPUT FLAG
0001 =             OUTRDY  EQU  1      ;MASK FOR POFLG STATUS BIT
                   ;
                   ;FIRST ROUTINE IS FOR NS DOS 5.2DQ WITH ORIGIN AT 0100H
                   ;
0AC5                       ORG 0AC5H
                   ;
                   ;NO NEED TO CHECK STATUS OF PO FLAG--DOS DOES THIS ELSEWHERE
                   ;
0AC5 78            COUT2   MOV A,B     ;GET CHARACTER
0AC6 D300                  OUT PAROUT  ;SEND TO PRINTER
0AC8 3E20                  MVI A, PORST
0ACA D306                  OUT MOTHER  ;RESET PO FLAG
0ACC 78                    MOV A,B     ;RESTORE CHARACTER
0ACD 00                    NOP         ;FILL BLANKS LEFT BY SHORTER ROUTINE
0ACE 00                    NOP
0ACF 00                    NOP
0AD0 00                    NOP
0AD1 00                    NOP
0AD2 00                    NOP
0AD3 00                    NOP
0AD4 00                    NOP
0AD5 00                    NOP
0AD6 00                    NOP
0AD7 C9                    RET
                   ;
                   ;
                   ;NEXT IS THE CP/M VERSION
                   ;
                   ;STARTING ADDRESS IS FOR THE SYSGEN IMAGE OF NORTHSTAR'S CP/M 2.2.
                   ;FOR OTHER CP/M'S, CHECK YOUR DOCUMENTATION
                   ;
```

the software be patched to make it all work?

The software solution

Reading through the Horizon's *Computer System Manual*, I found an example of an output routine for a "hypothetical parallel printer." As I carefully examined this routine, it dawned on me that "hypothetical parallel printer" was a synonym for "Epson MX-80."

I have modified and assembled two versions of this routine, which was intended for North Star's DOS 5.2DQ and the SYSGEN image of North Star's CP/M 2.2 (see listings). Since the new parallel output drivers are 10 bytes shorter than the original versions, NOPs are included. Note that DOS uses register B as the source of the next character to output, but CP/M uses register C.

These routines will work only if the parallel port header is wired according to the instructions and diagrams shown in the *Computer System Manual*. That is, the header's pin 1 must be jumpered to pin 14, pin 2 to pin 13, and pin 4 to pin 12. My Horizon came with the correct jumper configuration and required no changes, but you might want to check your own.

Up and away!

Once you've reached this stage, the only thing left to do is to let your operating system know where to send its printer output. Under DOS, substitute device #2 (for parallel output port) everywhere you would have used device #1 (second serial port). With North Star's CP/M, invoke CPMGEN, tell it that the printer is now on the parallel port, and exit with a warm boot. Then use DDT to load the .HEX or .COM file that contains the BIOS patch over the memory image of CP/M, and exit again with a warm boot (^C or G0). Finally, invoke CPMGEN to write the memory image of the reconfigured version back to the system tracks.

If you are using someone else's CP/M, check the BIOS listing for the parallel output driver. If it differs substantially, you can patch this routine into your BIOS. You then have two options. If your CP/M doesn't use the IOBYTE, you can use STAT LST:=LPT: every time you cold boot to direct printer output to the parallel port.

However, it would be much more convenient to change the IOBYTE so that output will be properly directed automatically. If the IOBYTE was originally set up to use the second serial port, its two high-order bits will be 00 (you should be able to find this somewhere in an initializing segment of the BIOS). Change these two bits to 10, get a SYSGEN image of the revised CP/M into memory, write it onto the system tracks, and you're ready to go! ▣

What really convinced me to get North Star's CP/M was the fact that it can relocate itself "around" the disk controller and floating-point board.

Parallel Port continued . . .

```
3270                 ORG 3270H
3270 DB06    COUTP   IN MOTHER    ;GET MOTHERBOARD STATUS BYTE
3272 E601            ANI OUTRDY   ;CHECK POFLG
3274 CA7032          JZ COUTP     ;LOOP BACK IF NOT READY
3277 79              MOV A,C      ;GET CHARACTER TO SEND
3278 D300            OUT PAROUT   ;SEND IT
327A 3E20            MVI A,PORST
327C D306            OUT MOTHER   ;RESET POFLG
327E 79              MOV A,C      ;RESTORE CHARACTER
327F 00              NOP          ;FILL GAP LEFT BY SHORTER ROUTINE
3280 00              NOP          ;NUMBER OF NOP'S MAY NEED TO BE CHANGED
3281 00              NOP          ;  FOR DIFFERENT VERSIONS OF CP/M
3282 00              NOP
3283 00              NOP
3284 00              NOP
3285 00              NOP
3286 00              NOP
3287 00              NOP
3288 00              NOP          ;END OF CP/M PATCH
3289 C9              RET
```

Digital Audio on a CP/M System

by Joseph W. Long

We have developed a very simple package that allows digital recording and playback of audio using a North Star Horizon, Cromemco AD/DA converter, small stereo amplifier, and a simple assembly language program. The package would be usable on an 8080- or Z80-based system that has an AD/DA converter and an assembler.

The program consists of three loops: record, playback, and set speed. The record and playback routines are nearly identical. The record routine loads a digitized audio value at the top of memory, waits through a delay loop (variable using "set speed" routine), decrements memory pointer, stores a new digitized audio value from the A/D converter, and continues until zero memory is reached, where the loop repeats. The playback routine is similar, except that values in memory are output to the D/A converter. The program takes a few bytes at the highest available memory. All memory below this point is used for storage of audio.

The small keyboard monitor routine in the playback and record segments allows you to jump between playback, record, speed set, and the operating system. In order to keep the loops as small as possible, we didn't use any subroutines in the monitor. When the set speed routine is entered, it waits for some key other than the 'S' key to be touched, then uses whatever value it finds in the accumulator as the new delay value.

Using our 56K North Star CP/M, B000 is as high as the program can be placed without overwriting CP/M. Using a 32K CP/M, the program can be put at DC00, with all space below it available for recording. This has the disadvantage that we have been unable to make a .COM file using Microsoft L80, so that the linker must be used each time the program is to be executed, which is a minor inconvenience, but does give a bit more record/playback time. Perhaps a reader could suggest a method that we could use to make this .COM file. We have been able to do this by loading the program at DC00, then booting up North Star DOS and saving the memory image in a North Star DOS file, but there ought to be a better way.

Joseph W. Long, Department of Chemistry, Broome Community College, Binghamton, NY 13902

This routine was developed initially because it seemed like an interesting project and because we had the equipment; everyone is fascinated with a "talking" computer. It has proven quite useful in a computer course taught to our Chemical Technology freshmen, where it is used to illustrate the use of a computer for data logging—a very common application for small computers in chemistry. The trade-off between record time and fidelity clearly illustrates the relationship between the frequency of the sampled signal, the sampling rate and distortion. At the highest quality and sampling rate, 64K of RAM allows about 5 seconds of sound. At lower rates the time is increased—30–40 seconds of recognizable speech can be stored. This is much more effective than, say, a discussion and demonstration using the signal from an infrared spectrophotometer or other piece of chemical instrumentation, of which our students, as freshmen, have little knowledge.

Whistling into the microphone using different record rates will produce some interesting effects. At all but the highest recording rates, distortion can be plainly heard as an additional tone that appears when you whistle at higher frequencies. Adjustment of the tone controls of the audio amplifier will prove instructive. At lower recording rates, much distortion can be removed by cutting the treble response, which has the effect of adding a low pass filter, attenuating the higher frequency distortion products. Recording and playback at different rates will produce a range of interesting sound effects, ranging from "chipmunk" to very sinister "Darth Vader" types of voices.

We have noticed that it would be easy to modify the software for stereo by adding extra input and output commands, together with an increment or two. This would unfortunately have the effect of cutting in half record/playback times. It would also be interesting to be able to save digitized audio in a disk file, and no doubt readers will have their own ideas for other modifications or extensions.

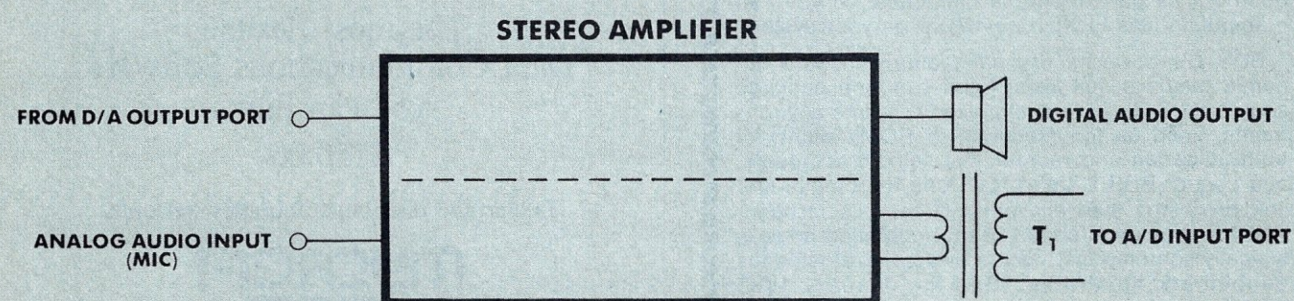


Figure 1. T_1 is an output transformer from an old 5-tube AC/DC radio. A 6V (or lower) filament transformer would also work.

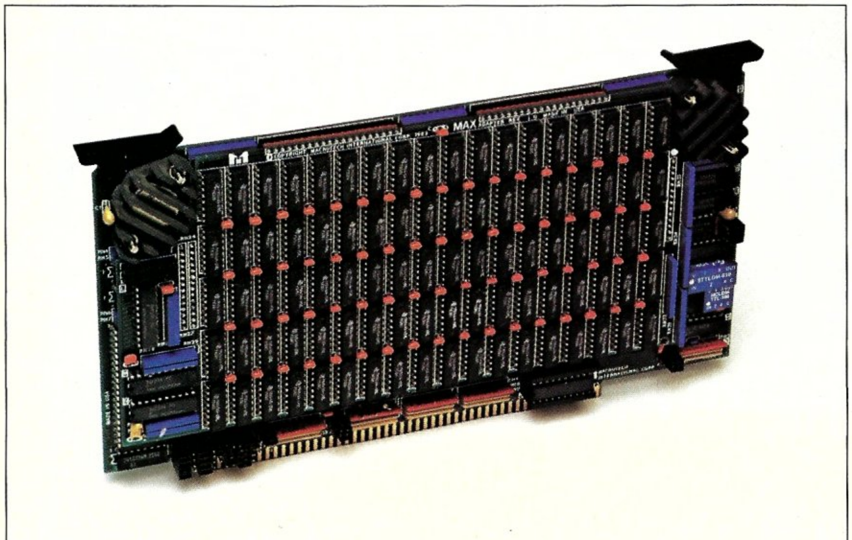
EXTRA

EXTRA

S-100 World News

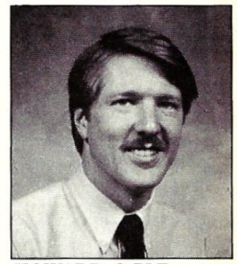
MACROTECH International Corporation **20630 Lassen Street, Chatsworth, California • 213-700-1501**

NOW 1 MEGABYTE *MAX* FOR ALPHA MICRO



CHATSWORTH—June 30, 1983—Mike Pelkey, Macrotech International President, announced today that a special version of ***MAX*** is now running in Alpha Micro Systems.

This special version is available only through Soft Machines of Champaign, IL. (217) 351-7199. Howard Ogle of Soft Machines stated, "The new **AM-*MAX*1** runs full speed with all three Alpha S100 machines." Ogle also said, "The **AM-*MAX*1** is not only the most economical memory for Alpha, but the most versatile as well. The system is even faster with Soft Machines' 'GO FAST' disk cache utilities."



HOWARD OGLE

Bob Rubendunst of Soft Machines reports, "Every ***MAX*** is shipped with software that greatly simplifies implementation on bank switched systems. Also included are detailed installation instructions and diagnostic programs."

Dealer inquiries and orders should be directed to Bob at Soft Machines.

VIRTUAL DISK NOW NONVOLATILE

CHATSWORTH—June 30, 1983—Mike Pelkey announced today the release of the latest addition to the Macrotech product family. The **B-Board** is a multifunction system support board, for use with ***MAX*** and **128ST** memories. Used with the **128ST**, this combination creates a complete disk emulation, including nonvolatility. The **B-Board** features include battery backup, power fail monitor, and charging circuitry for on or off board batteries.

The **B-Board** functions also include a time-of-day clock, using a National Semi device for hassle free operation. It also gets early warning at power down, so the time-of-day can't suddenly get creative. An interrupt is available which can be used to turn the system on or off at a preset time.

On board ROM space accepts the users' EPROM based program storage. It can be configured to accept one or a pair of any EPROM type from 2716 to 27256, in 8 or 16 bit format. It supports a wake up jump option with full or shadowed phantom overlay.

The ERROR TRAP feature is designed to support the parity error detection feature of the ***MAX*** series dynamic memories. Any activity on the system's ERROR line causes the trap to record the extended address and data busses and 20 bits of bus status information. Up to 16 events can be trapped; the trap issues an interrupt when it's full.

The **B-Board** is a logical addition to the growing family of Macrotech International's no-compromise S100 boards for no-compromise users.

MAX Split Personality

BURBANK—June 30, 1983—"Many current operating systems permit ***MAX*** to double as both virtual disk and system memory," stated Dan West of Westcom Systems. As an example, an MP/M 2.1* system using ***MAX*-M** could be configured as a 512K system memory and a 512K Vdisk. A typical CP/M 3.0* configuration could be 256K of system memory and up to 768K Vdisk. CP/M 2.2*, of course, only permits a 64K system memory, leaving the balance for a virtual disk. With ***MAX***, or the **128ST**, both functions can run simultaneously in a single memory board.

MACROTECH Moves

CHATSWORTH—June 30, 1983—Macrotech has moved to larger facilities located at 20630 Lassen St., Chatsworth, CA 91311. The new phone number is (213) 700-1501. "Due to a healthier marketplace and a phenomenal demand for the ***MAX*** series, larger facilities were necessary. This permits additional staffing, increased production, and customer support levels," said Mike Pelkey, President of Macrotech.

Virtual Disk for CP/M 86*

Dan West, Westcom Systems

BURBANK—June 30, 1983—Most of the CP/M 86* application programs available today fail to take advantage of the possible one megabyte address space. Virtual Disk for CP/M 86* will convert this unused space into RAM resident disk capacity for greatly improved disk access processing. The easily installed Virtual Disk 86 software module has been added to Macrotech's applications software available to owners of ***MAX*** series and **128ST** memory boards.

PRICE INDEX

	SIZE	P/N	PRICE
Static Memory	128K	128-ST	$1232
Dynamic Memory	256K	MAX-256	$1108
24-bit	384K	MAX-384	1292
Addressing	512K	MAX-512	1647
	640K	MAX-640	1737
	768K	MAX-768	1815
	896K	MAX-896	1899
	1M	MAX-M	1983
With 16-bit M^3 Addressing option, add $91			
Software (provided on 8″ disk)			
Virtual Disk for MP/M 2.1* and CP/M 2.2*, CP/M 3.0* Bios modules, CP/M memory tests			$ 25
Manuals (sold separately)			
128/ST			$ 15
MAX Technical Manual			15

*CP/M 2.2, CP/M 3.0, CP/M 86 and MP/M 2.1 are registered trademarks of Digital Research Inc.

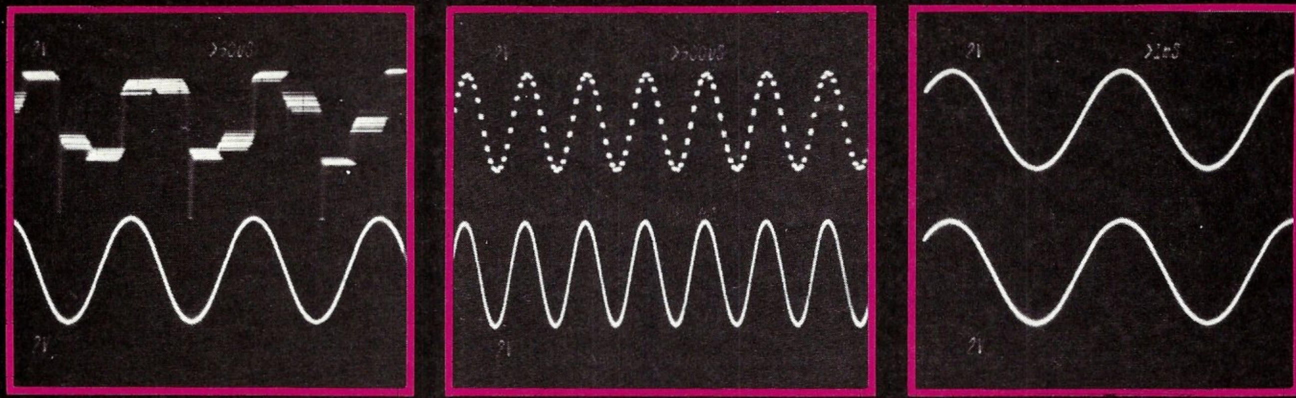


Figure 2. *In each case, the top trace is the digitized audio, and the bottom, the original audio. The frequencies are approximately 100 Hz, 1 KHz, and 8 KHz. These traces were done at the highest recording rate. Obviously, you will not achieve super high fidelity with this package!*

Joseph W. Long is a professor in the Department of Chemistry and Chemical Engineering Technology, Broome Community College, Binghamton, New York. His computer interests are laboratory applications of microcomputers, graphics, and computer-aided instruction. His other interests include R.C. modeling and ham radio (call WA2EJT). He is also a recipient of the State University of New York Chancellor's Award for Excellence in Teaching in 1982.

```
;
;               DIGITAL AUDIO
;               COPYRIGHT MAY 1982
;
;               J.W. LONG AND J. SIMON
;               CHEMISTRY DEPARTMENT
;               BROOME COMMUNITY COLLEGE
;               BINGHAMTON, NY 13902
;
;
;               607-722-5009
;
;
;       THIS PROGRAM WILL 'RECORD' AND 'PLAYBACK' AUDIO,
;       USING AN AREA OF MEMORY JUST BELOW THIS PROGRAM, AND
;       EXTENDING TO ZERO MEMORY. THE AUDIO IS MOVED INTO
;       (RECORD) AND OUT OF (PLAYBACK) MEMORY THROUGH AN 8 BIT AD/DA
;       CONVERTER (CROMEMCO D+7A I/O)
;
;       THE RECORD ROUTINE WORKS BY SAMPLING A VOLTAGE AT THE
;       ANALOG INPUT PORT, THEN SAVING IT IN MEMORY. AFTER A
;       DELAY, A MEMORY POINTER IS DECREMENTED, AND A NEW VOLTAGE
;       IS SAVED. THIS CONTINUES UNTIL THE BOTTOM OF MEMORY IS
;       REACHED, WHEN THE LOOP REPEATS. A KEYBOARD MONITOR
;       ALLOWS JUMPING BETWEEN 'RECORD', 'PLAYBACK' AND
;       'SET SPEED' ROUTINES.
;
;       THE PLAYBACK ROUTINE IS NEARLY IDENTICAL TO THE RECORD
;       ROUTINE, EXCEPT THAT DATA ARE MOVED FROM MEMORY TO THE
;       DIGITAL TO ANALOG CONVERTER
;
;       THE SET SPEED ROUTINE READS A VALUE FROM THE KEYBOARD
;       WHICH IS USED TO CONTROL THE RECORD/PLAYBACK SPEED.
;       THE LETTER 'A' GIVES THE SHORTEST TIME (ABOUT 2
;       SECONDS), 'Z' THE LONGEST (OVER A MINUTE)
;
;       HARDWARE:
;               NORTHSTAR HORIZON, 64K QD
;               CROMEMCO D+7A I/O AD/DA
;               LEAR SIEGLER ADM 3A+
;
;       SOFTWARE:
;               CP/M 2.2 56K
;               MICROSOFT M80 ASSEMBLER WAS USED TO ASSEMBLE
;               THIS PROGRAM. CP/M ASSEMBLER ALSO WORKS
;
;       HARDWARE SPECIFIC CODE PORTIONS: THESE SHOULD BE
;               ORG                      MODIFIED TO SUIT
;               KBD                      YOUR HARDWARE
;               MIC
;               SPKR
;               MONIT
;               DISP
;               CHAR M,R,P,S
;               MEMH , MEML
;
;
ASEG              ; MICROSOFT L80 LINKER SEEMS TO NEED THIS
ORG       0B000H
MEMH  EQU 0AFH  ; HI BYTE         MEMORY AREA FOR AUDIO STORAGE
MEML  EQU 0FFH  ; LO BYTE         IS AFFFH DOWN TO 0000
MONIT EQU 0E800H; NORTHSTAR BOOT PROM ADDRESS
DELAY EQU 01D   ; DEFAULT DELAY VALUE. MAY CHANGE VIA SET SPEED ROUTINE
KBD   EQU 02D   ; TERMINAL KEYBOARD INPUT PORT NUMBER
DISP  EQU 02D   ; VIDEO DISPLAY OUTPUT PORT NUMBER
CHARB EQU 66D   ; CHARACTER B
CHARM EQU 205D  ; CHARACTER M
CHARR EQU 210D  ; CHARACTER R
CHARP EQU 208D  ; CHARACTER P
CHARS EQU 211D  ; CHARACTER S
SPKR  EQU 31D   ; D+7A I/O  ANALOG OUTPUT PORT #7
MIC   EQU 31D   ; D+7A I/O  ANALOG INPUT  PORT #7
;
;
;               PLAYBACK ROUTINE
;
;
```

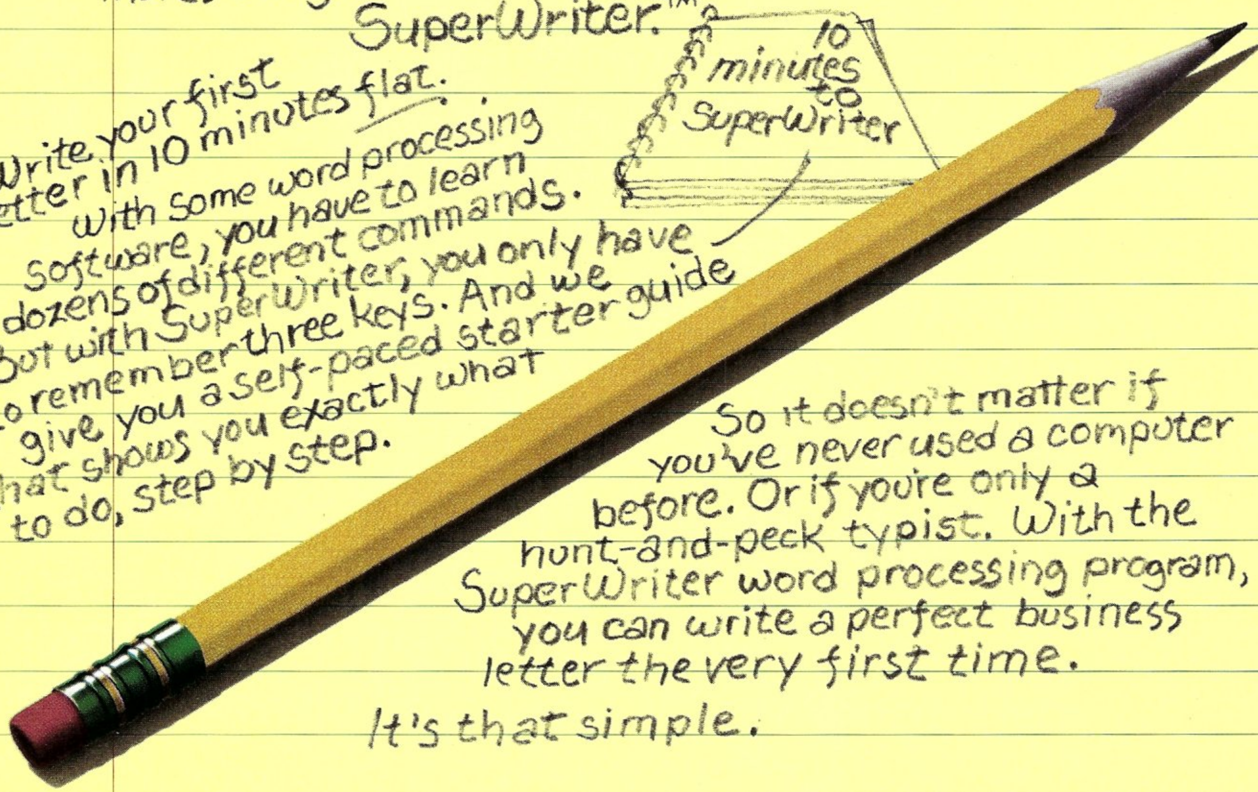
```
         MOV D , DELAY    ; LOAD DEFAULT DELAY VALUE
PLAYB:   MVI A , CHARP    ; SET UP A 'P'
         OUT     DISP     ; PRINT THE 'P' ON THE VIDEO DISPLAY
         MVI H , MEMH     ; SET POINTER TO THE TOP OF MEMORY
PLAB1:   MVI L , MEML
PLAB2:   MOV B , D        ; LOAD DELAY VALUE
PLAB3:   MOV A , M        ; GET AUDIO FROM MEMORY
         OUT     SPKR     ; PLAY IT
         OUT     SPKR     ; REPEAT TO KEEP REC AND PLAYB LOOPS SAME LENGTH
         IN      KBD
         CPI     CHARM    ; 'M' KEY PRESSED ?
         JZ      MONIT
         CPI     CHARR    ; 'R' KEY PRESSED ?
         JZ      REC
         CPI     CHARS    ; 'S' KEY PRESSED ?
         CZ      SPEED
         DCR     B        ; DELAY LOOP
         JNZ     PLAB3    ; KEEP WAITING
         DCR     L        ; BUMP MEMORY POINTER (LSB)
         JNZ     PLAB2
         DCR     H        ; BUMP MEMORY POINTER (MSB)
         JNZ     PLAB1
         JMP     PLAYB    ; ALL DONE GO START AT TOP OF MEMORY AGAIN
;
;
;                RECORD ROUTINE
;
;
REC:     MVI A , CHARR    ; SET UP AN 'R'
         OUT     DISP     ; PRINT THE 'R' ON VIDEO DISPLAY
         MVI H , MEMH     ; SET POINTER TO TOP OF MEMORY
REC1:    MVI L , MEML
REC2:    MOV B , D        ; LOAD DELAY VALUE
REC3:    IN      MIC      ; GET SOME AUDIO
         MOV M , A        ; SAVE IT IN MEMORY
         OUT     SPKR     ; OUTPUT IT, FOR MONITORING
         IN      KBD
         CPI     CHARM    ; 'M' KEY PRESSED ?
         JZ      MONIT
         CPI     CHARP    ; 'P' KEY PRESSED ?
         JZ      PLAYB
         CPI     CHARS    ; 'S' KEY PRESSED ?
         CZ      SPEED
         DCR     B        ; DELAY LOOP
         JNZ     REC3     ; KEEP WAITING
         DCR     L        ; BUMP MEMORY POINTER (LSB)
         JNZ     REC2
         DCR     H        ; BUMP MEMORY POINTER (MSB)
         JNZ     REC1
         JMP     REC      ; ALL DONE. GO START AT TOP OF MEMORY AGAIN
;
;
;                SET SPEED ROUTINE
;
;
SPEED:   IN      KBD      ; READ KEYBOARD
         CPI     CHARS    ; 'S' KEY PRESSED ?
         JZ      SPEED    ; WAIT TILL 'S' IS PRESSED
         ANI     01FH     ; MASKING INCREASES DELAY RANGE
         MOV     D,A      ; SAVE NEW DELAY VALUE
         RET
;
;
         END     0B000H

A>
```

You're looking at the only word processor that's as easy to use as SuperWriter.™

If you're like most people, you don't want to make a career out of word processing. When you have to write a simple letter or memo, all you want to do is roll up your sleeves and get down to business.

And if you're buying a personal computer, now there's a new software package that makes things just that simple.

SuperWriter.™



Write your first letter in 10 minutes flat.

With some word processing software, you have to learn dozens of different commands. But with SuperWriter, you only have to remember three keys. And we give you a self-paced starter guide that shows you exactly what to do, step by step.

So it doesn't matter if you've never used a computer before. Or if you're only a hunt-and-peck typist. With the SuperWriter word processing program, you can write a perfect business letter the very first time.

It's that simple.

Everything you need for only $295.00 - even a spelling checker.

We're talking about a really complete package here. You get a full-function word processor, our 20,000-word SuperSpellGuard™ spelling checker, and a form letter generator. All for $295.00 suggested retail. To get all these functions any other way, you might have to spend $900.00 or more.

You never knew your letters could look so good.

We've built all kinds of professional features into SuperWriter. Like proportional spacing. Boldface highlighting. Memo formatting. And more.

So even if you're a complete beginner, you'll wind up with completely professional results.

Help is closer than you think.

If you get stuck on a problem with any other word processing software, you'll probably have to go back to the store for help.

But if you have any questions at all about SuperWriter, all you do is pick up the phone. We've got a full-time crew of customer service experts on call, ready to help you out.

Works on lots of personal computers.

About the only thing it can't do is doodle.

Right now, SuperWriter is available for the IBM PC and the new IBM PC XT. Of course, it takes full advantage of their features—full color, hard disk and more.

By June 15, we'll have SuperWriter ready for CP/M, CP/M-86 and MS/DOS operating systems, which means you can use it on most of the popular personal computers.

So stop by your nearest computer or software store soon and ask about SuperWriter. In just 10 minutes, you can learn everything you ever need to know about word processing. And see why every other system is too complicated for words.

SuperWriter™

SORCIM®

2310 Lundy Ave.
San Jose, CA 95131
(408) 942-1727

The Custom CBIOS: Patching to Promote Peripheral Power

by Tom Wiens

Most microcomputer users take their operating system for granted, as something to be lived with for better or for worse. Often enough they have little choice— the routines may be installed in read-only memory (ROM), or the manufacturer supplies no intelligible documentation. Users of most CP/M operating systems have no such excuse, though, since there usually is an accessible area of BIOS (known as the CBIOS) open to user modification. Even so, it takes a bit of knowledge of assembly language to make more than trivial modifications of input-output routines, and it may not seem worth the investment of time and effort.

Why bother? Well, for one thing, today even low-end dot matrix printers are capable of extraordinary "intelligence," possessing a variety of alternative character sets that include special and graphic sysbols. Normally, these can be accessed only through control codes, which must be built into each program and can't be directly input through the keyboard. Wouldn't it be useful if special-function keys or unused control characters could be put to use turning on special printer functions directly from the keyboard, without modifying programs?

For another reason, many computers have a variety of serial and/or parallel ports. With a well-implemented commercial CBIOS, it is not difficult to access these for simple input or output (using the IOBYTE function), but more complex I/O patterns are not directly supported by CP/M. Today, when one doesn't have the only computer on the block, or with the advent of the "two-computer family," how does one get one computer talking to or controlling another, or exploiting common peripherals?

Custom modification of the CP/M CBIOS can add a new flexibility and power to your computer system, and this may be worth the bother of learning a little assembly language. This article takes my routines as examples of what can be done, how to go about it, and for what purpose. The CBIOS shown in Listing 1 is written in 8080 assembly language, sprinkled here and there with Z80 instructions that save considerable programming. The system for which it was written happens to include a North Star Horizon, a custom keyboard with parallel interface, an IMSAI VIO-C video board, and a NEC PC 8023A-C printer with parallel interface. On top of this, I use one serial port for a modem (run at 300 baud) and another to interface with another computer at 1200 baud. This kind of complexity forced me to become a novice assembly-language programmer rather early on. To keep the size of this article within reason, I'll assume that you have available a good tutorial or reference material, such as Kathe Spracklen's *Z80 and 8080 Assembly Language Programming* (Hayden Books, 1979), William Barden Jr.'s *The Z80 Microcomputer Handbook* (Howard W. Sams & Co., 1980), or Alan R. Miller's *8080/Z80 Assembly Language* (Wiley, 1982).

Tom Wiens, 2025 Westwood Terrace, Vienna, VA 22180

Design objectives

In writing the user area, I had these objectives in mind: I wanted to control as many printer functions as possible from the keyboard with no more than two-key output. Also, the printer has a Greek character set and a set of superscript numbers and special mathematical symbols, but the codes that access them are a meaningless jumble. I wanted to create a "function key" that would switch to an alternate character set, and then to recode the keyboard so that each special character would be assigned to the easiest-to-remember conventional character (for example, each Greek character would be accessed by the closest Latin character; superscript numbers by ordinary numbers, etc.). One character set on my printer appeared too crowded on paper, and I wished to increase the spacing by adding fractional spaces between each character output to the printer.

Aside from printer control, I wanted easy control over some special functions offered by the video board, such as reverse video. I wished to implement the IOBYTE control function (offered automatically by many CP/M systems) to control I/O device assignments, and more—I wanted to enable computer #2 (an Osborne I) to communicate with a parallel-interfaced printer via the serial input on computer #1 (the Horizon).

Space restrictions

Other things I might have done were precluded by lack of space. My implementation of CP/M has a liberal two pages (200H bytes) available for CBIOS, but much of this is absorbed by the conventional routines that every CBIOS must have, including the jump table at the beginning, hardware-specific initialization routines, and one input and output routine for each port to be used. For example, an earlier version included a "typewriter" routine that set printer tabs for letters and envelopes, allowing the keyboard to bypass CP/M and "type" directly to the printer whenever a particular control character was input. This go-round left insufficient space to include the typewriter routine. One of your first tasks, should you wish to implement a similar routine, is to figure out from your system's documentation how much space you can play with. Assuming you can locate the beginning of your CBIOS user area, run DDT and use the L(ist) command to see where the user area ends, if it isn't obvious from the literature. Look in particular for the series of NOP (no-operations) which may follow the existing user area, and see where normal code picks up again. Most likely the NOPs represent unused area in CBIOS that you can program, aside from replacing existing code.

Creating a new user area

The sequence of maneuvers required to create and save a new or modified user area is relatively simple (unlike the programming involved). It is:

1. Create the CBIOS routine using a text editor, starting from the routines supplied with your CP/M documentation (if any).

2. Assemble it into a .HEX file, using the ASM assembler supplied with CP/M or any commercial assembler (mine was assembled with Digital Research's MAC, using the libraries of Z80 macros that allow mixing Z80 and 8080 code).

3. Use the command MOVCPM ## * [CR] to create a memory image of the CP/M system of size ## appropriate to your system (this size is given on the screen when CP/M is loaded from disk), then save it as a file by typing SAVE 32 CPM##.COM.

4. With CPM##.COM. and USERAREA.HEX (as assembled) stored on disk A:, run DDT on disk B: by typing B:DDT CPM##.COM. Read in the new code by typing

```
-IUSERAREA.HEX
-R####
```

where #### is a four-digit hexadecimal offset that varies with the size of your system. Determining this number is a bit tricky; it is explained none too clearly in the *CP/M System Alteration Guide*. The idea is to find the address where the CBIOS is stored by DDT, which is considerably lower than the normal origin of CBIOS when CP/M is loaded. The difference is the offset value (in two's complement arithmetic), and can be computed by DDT using the command Hx,n where x is the CBIOS origin and n is its address in the CPM##.COM file as loaded by DDT (check for the series of JMP commands, which should be found at address n). The second number printed in response to this command is the desired offset.

5. Type Control-C or G0 to escape from DDT. Type B:SYSGEN, then type a carriage return when asked where to get the system from, and "A" when asked where to put the system.

6. When SYSGEN is through, do a reset and reboot the computer to load the modified CP/M. You can expect to go through this sequence many times before your modifications work correctly!

Anatomy of the CBIOS

In its basic form, the CBIOS is very recognizable. It always begins with exactly the same jump sequence shown in the listing. It always includes routines that perform the functions associated with INIT (initialization), CONST (console status), CONIN (console character input), CONOUT (console character output), LIST (list output), PUNCH (any other output), and READER (input from a port). All but the first of these routines commonly includes a loop that checks the status of an input or output port, and an I/O instruction that is executed when status indicates availability, sometimes a routine to reset the port, masks to strip off unwanted bits of data, and logical tests to set flags for the operating system to read.

When the computer can access any of several ports, it is common for CBIOS to use the IOBYTE (a byte of data stored at memory address 03H in CP/M) as the code or switching device determining which port is actually accessed to perform one of the major functions listed above. This involves initializing this byte to a default value (see routine INITR in the listing) and subsequently testing it for changes in device assignment (made with the STAT command). The form of this byte and use of the STAT command to change it should be studied carefully in your CP/M documentation or reference material before you exploit it in your user area programming.

What the CBIOS does with the IOBYTE is very much your decision. If the CBIOS supplied with your version of CP/M fully utilizes the IOBYTE conventions, more code may be wasted doing this than is required for your installation, and you can delete some of this (with care). Although it is best used for its intended purpose (to switch among devices), it may be regarded as an all-purpose switch that can be changed from the keyboard (with the STAT command). In my routines, it does the following: the console bits allow choice between the default console input (parallel input port) and a special routine that inputs from one serial port and immediately outputs the data received to a printer via the parallel output port. The list bits choose among two serial and one parallel output port and the associated routines. The reader bits choose input routines from among two serial ports. The punch bits are ignored (punch output goes to the console).

Keyboard control of input-output involves three devices in my system: *a)* My keyboard has some uncommitted keys, which output logic levels rather than character codes, while the Horizon has some correspondingly uncommitted lines on the serial input ports and user-programmable bits in the status byte. So I connected the keys via the uncommitted lines to the user-programmable status bits, and read those bits to control special functions. *b)* Some control codes are not normally used or meaningful to programs, yet can be output from a keyboard. If the hardwired special-function keys had not been available, I would use such codes (e.g., ^Q or ^]) to give the equivalent signals to the user area. *c)* My keyboard only sends a 7-bit code, whereas the computer accepts 8 bits. One uncommitted key was connected to the eighth bit of the serial interface, providing another switch.

Testing for conditions set from the keyboard is performed in the console input routine TTYIN. When CP/M has nothing better to do, it rests in the status loop of this routine, waiting for the signal to input a byte. Such a loop often consists of three operations: an IN operation from the status port, a masking of relevant bits (e.g., ANI 2), and a jump to the beginning of the loop if the bit is either zero or one

Custom modification of the CP/M CBIOS can add a new flexibility and power to your system, and this may be worth the bother of learning a little assembly language.

ANOTHER S-100 HARD DISK SOLUTION THAT PUTS YOU IN CONTROL.

5¼" Winchester control at a price that can't be beat!

$259 *large OEM quantities*

From the spectrum of Konan S-100 solutions comes the DGC-100. Offering full sector buffering, error correction, automatic flaw mapping...all standard. The DGC-100 supports the known spectrum of ST-506 compatible drives. And features what is possibly the best data clock separator on the market today.

Konan's DGC-100. Reliability. High performance. On-time deliveries. And...a low, low price.

For more information or to place an order, write or call:

KONAN T.M.
in control

1448 North 27th Avenue, Phoenix, Arizona 85009
(602) 257-1355, TWX/TELEX 9109511552

CIRCLE 242 ON READER SERVICE CARD

(through a JZ or JNZ operation). But this loop can be expanded to encompass all sorts of tests for special conditions that do not involve character input. In my TTYIN, the status loop includes a test of the IO-BYTE value, with a possible jump to a different routine. A test of a hardwired user-programmed bit from the status byte, which is "1" if a certain key is depressed, requires a code turning on reverse video to be sent to my video board. A test of another user bit determines my desire to send control codes to the printer, and jumps to a special routine written for this purpose.

Once past the status loop, a character is input, and the eighth bit is tested to see whether another key has been depressed. If so, the routine sets a flag to use an alternative (Greek plus special symbol) character set in printing, and also sends a control code to the video board to turn on character-by-character reverse video as a visual indication of this action. The eighth bit itself is stripped off (with ANI 7FH) before the byte is returned to CP/M, as is customary.

While TTYIN reads the signals from the keyboard, special routines must be added to implement them. These routines may serve as models for similar routines in your system. For examples:

PPASS is a routine allowing computer #2 to access the printer through computer #1 (serial-to-parallel conversion is involved). The routine inputs from one port and outputs to another. In between, it checks each incoming character to see if it is an ETX (end of transmission) control code; if so, it sends an ACK (acknowledge) back. There are two reasons for using an ETX/ACK protocol, as this is called: First, computer #2 (the Osborne I) uses such a protocol in communicating with another printer; second, computer #2 unfortunately has a "dumb" serial interface that does not test whether the recipient of this data is ready to receive. The only way to prevent computer #1 from missing parts of the data is to have computer #2 send an ETX after each byte and wait for the corresponding ACK before proceeding.

PFUNCT is jumped to whenever a certain key is pressed. It accepts without displaying one number, 0-9, which is treated as a function number. It then looks up the list of control codes which correspond to that function number, and sends them to the printer. The control codes are stored in DB (databyte) statements running from FUNCT to FUNCT9, and JUMPT lists the number of bytes for each function. The code is a good example of how to access economically an element of a table when the number of bytes in each element is not the same. However, note the warning included in the program comments.

If each element involves only one byte, processing may be simplified, as shown in the WGREEK routine. This is a special case, though, because the list is one of "key equivalents", arranged in order of the ASCII codes of all the characters on the keyboard, starting with '(' and running through the capital letters. Both PFUNCT and WGREEK use the ASCII codes that are input to the routine, in combination with the address of the first element in the list, to find the address of the desired element. This byte is moved into the C-register in place of the normal ASCII code, and is output to the printer. The results can be seen in Listing 2: an entire series of keys have a new interpretation, one that is easy to remember. To distinguish the alternate character set on the screen, the presence of a "1" in the eighth bit causes character-by-character inverse video to be turned on at the same time as it sets the switch for "Greek." The same signal is used to turn off "Greek" and return to the normal character set.

The above approach works from the keyboard, but is no good if text is output to the printer directly and I wish to insert "Greek" characters into text, which has otherwise normal ASCII characters. So I use an alternate switch, and if control-? (or ASCII 1CH) is found in the text during printing, "Greek" is turned on or off. I only need bracket text to be printed as the alternate character set with this control code.

One of the character sets that can be accessed as a special function through the PFUNCT routine is proportional spacing, which on my printer is very attractive but looks cramped and puts too many characters on one line. I wanted to have the LPT routine add an extra fractional space after (or before) each character output when this character set has been selected. PFUNCT therefore stashes the function code (0-9) in FUNCBYT, the LPT0 routine jumps to a routine FCHECK to see if it is "4" (indicating that proportional spacing has been selected), and if so the code to print a fractional space is output before the character is sent. This routine has its drawback: If a control string consisting of a sequence of codes is sent to the printer via the LPT0 routine, the spaces are added between codes, and the printer fails to recognize it. Thus programs which send control strings to the printer must do so directly rather than through LPT0 if proportional spacing is to be used.

Eliminating problems

The user area sometimes can also be used to finesse hardware problems. For example, a time delay may be required before or after some terminal or printer function, and a time delay loop that is set up to iterate a particular number of times can be added to the user area and jumped to at the appropriate time. My computer has a peculiar hardware bug that causes it to lose the least significant bit of any ASCII code coming in via parallel port whenever the least significant bit of the previous byte output via parallel port was not set. Rather than go to the expense of repairing the hardware, I finessed the bug by adding a rou-

Some functions are restricted by lack of space. One of your first tasks, should you wish to implement a similar routine, is to figure out from your system's documentation how much space you can play with.

S-100 DISK CONTROLLERS

VERSAFLOPPY II - SD Systems
Double density disk controller for any combination of 5¼" and 8" single or double sided, analog phase-locked loop data seperator, vectored interrupts. CP/M 2.2 & Oasis compatible control/diagnostic software PROM included.

IOD-1160A *A & T With Prom* — **$359.95**
SFC-55009047F *CP/M 3.0 With VF-II* — **$139.95**

DOUBLE D - Jade
High reliability double density disk controller with on-board Z-80A, auxilary printer port, IEEE S-100 can function in multi-user interrupt driven bus

IOD-1200B *Bare board & hdwr man* — **$59.95**
IOD-1200K *Kit w/hdwr & sftwr man* — **$299.95**
IOD-1200A *A & T w/hdwr & sftwr man* — **$325.95**
SFC-59002001F *CP/M 2.2 With Double D* — **$99.95**

DISK 1 - CompuPro
8" or 5¼" DMA disk controller, single or double density, single or double sided, 10MHz

IOD-1810A *A & T* — **$449.95**
IOD-1810C *CSC* — **$554.95**

2422 DISK CONTROLLER - C.C.S.
5¼" or 8" double density disk controller with on-board boot loader ROM. FREE CP/M 2.2 and manual set.

IOD-1300A *A & T with CP/M 2.2* — **$399.95**

WINCHESTER Disk Controller Monitor Dynamics
This S-100 hard disk controller board was designed to interface most 5¼" and 8" Winchester disks to your S-100 system. The controller will support either one or two drives, and with the INSTALL program provided, its easy to add an existing floppy-based system

IOD-5000A *Controller* — **$489.95**

Must Be Ordered With One Of The Following Data Seperator Boards:

IOD-5050A *5¼" Data Seperator* — **$75.00**
IOD-5080A *8" Data Seperator* — **$75.00**

See 1983 Jade Catalog For Further Information Concerning This Board

EPROM ERASERS

Ultra-Violet EPROM ERASERS
Inexpensive erasers for industry or home

XME-3100A *Spectronics w/o timer* — **$69.95**
XME-3101A *Spectronics with timer* — **$94.95**
XME-3200A *Logical Devices* — **$49.95**

8" DISK DRIVES

Siemens FDD 100-8 single-sided double-density
MSF-201120 — **$274.95 ea** 2 for **$249.95 ea**

Shugart SA801R single-sided double-density
MSF-10801R — **$394.95 ea** 2 for **$389.95 ea**

Shugart SA851R double-sided double density
MSF-10851R — **$554.95 ea** 2 for **$529.95 ea**

Qume DT-8 double-sided double-density
MSF-750080 — **$524.95 ea** 2 for **$498.95 ea**

Tandon TM848-1 single-sided double-den thin-line
MSF-558481 — **$379.96 ea** 2 for **$369.95 ea**

Tandon TM848-2 double-sided double-den thin-line
MSF-558482 — **$494.95 ea** 2 for **$484.95**

NEC FD1165 double-sided double-den thin-line
MSF-851165 — **$484.95 ea** 2 for **$449.95**

MODEMS

SMART BUY in Modems - Signalman
1200 and 300 Baud direct connect, automatic answer or originate selection, auto-answer, auto-dial on deluxe models 9v battery allows total portability, full one year warranty.

IOM-5600A *300 baud direct connect* — **$89.95**
IOM-5610A *300 baud Deluxe* — **$149.95**
IOM-5620A *1200/300 baud Deluxe* — **$369.95**
IOM-5650A *300 baud for Osborne* — **$119.95**
IOM-5630A *300 baud card for IBM* — **$269.95**

SMARTMODEM - Hayes
Sophisticated direct-connect auto-answer/auto-dial modem, touch-tone or pulse dialing, RS-232C interface programable

IOM-5500A *Smartmodem 1200* — **$574.95**
IOM-5400A *Smartmodem 300* — **$224.95**
IOK-1500A *Hayes Chronograph* — **$218.95**
IOM-1100A *Micromodem 100* — **$368.95**
IOM-2010A *Micromodem II w/term prgm* — **$329.95**
IOM-2012A *Terminal program for MMII* — **$89.95**

1200 Baud SMART CAT - Novation
103/212 Smart Cat & 103 Smart Cat 1200 & 300 baud, built-in dialer, auto re-dial if busy, auto answer disconnect direct conect, LED readout displays mode, analog digital loopback self tests, usaable with multi-line phones

IOM-5241A *300 baud 103 Smart Cat* — **$229.95**
IOM-5251A *1200 baud 212/103 Smart Cat* — **$549.95**

J-CAT™ MODEM - Novation
1/5 the size of ordinary modems, Bell 103, manual or auto-answer, automatic answer/orginate, direct connect, built-in self-test, two LEDs and audio beeps provide status information.

IOM-5261A *Novation* — **$149.95**

CP/M PLUS 3.0

NEW! CP/M Plus 3.0
CP/M 3.0 is Digital Research's latest version of the industry standard disk operating system. It features many performance improvements, such as intelligent record buffering, improved directory handling, "HELP" facility, time date stamping of files and many more improvements. A TREMENDOUS INCREASE IN SPEED!! it is fully CP/M 2.2 compatible and requires no changes to your existing application software. Available only to Versafloppy II owners with SBC-200 CPU's

SFC-55009059F *Unbanked, RS232* — **$250.00**
SFC-55009058F *Unbanked, VDB8024* — **$250.00**
SFC-55009057F *Banked, RS232* — **$250.00**
SFC-55009056F *Banked, VDB8024* — **$250.00**
SFC-55009057D *Manual* — **$50.00**

CP/M Plus On EPROM:

SFC-55009059E *Unbanked RS232* — **$400.00**
SFC-55009057E *Banked RS232* — **$400.00**

DISK TUBS

Diskette Storage Boxes
Clear Plexiglass storage boxes for up to 75 Diskettes

MMA-505 *Holds 75 5¼" Diskettes* — **$19.95**
MMA-508 *Holds 50 8" Diskettes* — **$24.95**

S-100 MEMORY BOARDS

64K STATIC RAM - Jade
Uses new 2K x 8 static RAMs, fully supports IEEE 696 24 bit extended addressing, 200ns RAMs, lower 32K or entire board phantomable, 2716 EPROMs may be subbed for RAMs, any 2K segment of upper 8K may be disabled, low power typically less than 500ma.

MEM-99152B *Bare board* — **$49.95**
MEM-99152K *Kit less RAM* — **$99.95**
MEM-32152K *32K kit* — **$199.95**
MEM-56152K *56K kit* — **$289.95**
MEM-64152K *64K kit* — **$299.95**
Assembled & Tested — **add $50.00**

EXPANDORAM III
SD Systems new ExpandoRAM III is a high density S—100 memory board utilizing the new 64K x 1 dynamic RAM chips. It allows memory sizes of 64K, 128K or 256K all on a single S-100 board

MEM-65064A *64K* — **$495.00**
MEM-65128A *128K* — **$595.00**
MEM-65192A *192K* — **$675.00**
MEM-65256A *256K* — **$755.00**
SFC-55009000F *RAMDISK sftwr CP/M 2.2* — **$44.95**
SFC-55009000F *RAMDISC with EXRAM III* — **$24.95**

RAM 17 - CompuPro
Meets or exceeds all IEEE 696/S-100 specifications including timing - works up to and including 10 MHz with 8088/86 CPUs Guaranteed to perform flawlessly with any IEEE 696/S-100 extended addressing specification

MEM-64180A *RAM 17, 64K A & T* — **$459.95**
MEM-64180C *RAM 17, 64K CSC* — **$549.95**

RAM 16 - CompuPro
There is an alternative to quirk-prone dynamic memories: thanks to low power consumption, high speed operation, fully static technology, and suitability for 8 or 16 bit systems, RAM 16 represents a new high in performance for sophisticated S-100 systems

MEM-32180A *RAM 16, A & T* — **$495.00**
MEM-32180C *RAM 16, CSC* — **$595.00**

RAM 21 - CompuPro
Operates at 12 MHz with 8088, 8086, 68000, 80286, and 16032 type CPUs, extremely low power consumption, meets or exceeds all IEEE 696/S-100 specifications, including timing

MEM-12810A *RAM 21 A & T* — **$995.00**
MEM-12810C *RAM 21 CSC* — **$1125.00**

8" SLIMLINE SUB-SYSTEMS

Dual Slimline Sub-Systems - Jade
Handsome vertical cabinet with scratch resistant baked enamel finish, proportionally balanced air flow system, quiet cooling fan, rugged dual drive power supply, power cables, power switch, line cord, fuse holder, colling fan, all necessary hardware to mount 2-8" slimline disk drives, does not include signal cable

Dual 8" Slimline Cabinet

END-000820 *Bare cabinet* — **$59.95**
END-000822 *A & T w/o drives* — **$179.95**

Dual 8" Slimline Sub-Systems

END-000823 *Kit w/2 SS DD* — **$919.95**
END-000824 *A & T w/2 SS DD* — **$949.95**
END-000833 *Kit w/2 DS DD* — **$1149.95**
END-000834 *A & T w/2 DS DD* — **$1179.95**

JADE

Computer Products

4901 West Rosecrans Ave. Hawthorne, California 90250

CIRCLE 16 ON READER SERVICE CARD

tine called TEST. Whenever something is output to the parallel port, the least significant bit of byte TRACK is set non-zero. Then when TTYIN is accessed for input, the routine jumps to TEST, which checks TRACK for its non-zero bit, and if it finds it, it sends a 01H to the parallel output port (which is ignored by my printer). So the last byte output before parallel input is performed is always 01H, and the bug disappears.

Some CP/M systems have a monitor implemented in ROM. In debugging programs, it is often handy to be able to jump out at any time to the monitor, check what's going on in memory or the registers, then return to program operation. This can be implemented in the CBIOS: on receipt of a special character or sensing a special function key, the TTYIN routine can call a routine which saves the stack (if required) and all registers, calls the monitor, then restores the stack and all registers on return from the monitor before returning to TTYIN.

There are numerous pitfalls in modifying the user area, as in assembly language programming in general, and the novice can expect to spend more time debugging his routines than writing them. You should examine carefully your existing routines to see what should be returned in each register (although CP/M is usually indifferent to what happens to registers not expected to contain particular input or output bytes). You should be careful about the distinction between jumps and calls to make sure that the last RET (return) in any sequence of operations takes you back to the calling program. The stack should be used liberally (but not excessively) to save bytes that could be lost in your routines, but you should make sure that for every PUSH there is a POP within the same subroutine, and that there is no condition which can bypass the POP. Be wary of the difference in effect of logical ANDs and comparisons (CPIs). Finally, work with a scratch disk until your user area has been fully tested, and only then copy the modified version of CP/M to the rest of your disk.

The reward for all this trouble is the control you obtain over all your peripherals, so that none of the intelligence you've paid for goes to waste. In addition, there are some intangibles: Once you have written your own user area, you are less likely to regard the workings of machine-language operating systems or programs as unfathomable mysteries, but rather as something to be deciphered and modified to suit your needs. You may even begin to have confidence that it is you rather than the computer who is in control of your system.

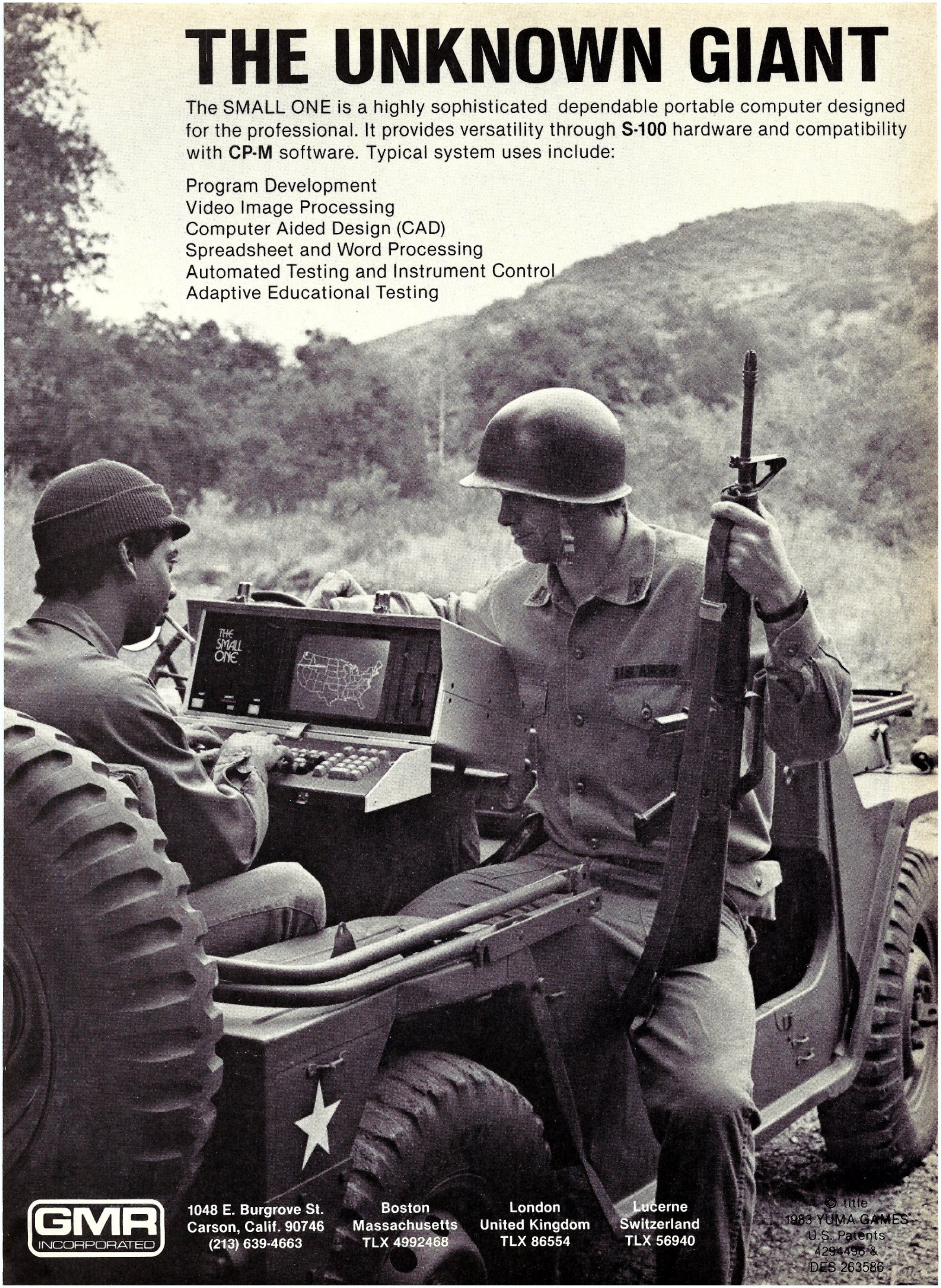
Listing 1.

```
;CUSTOMIZED USER AREA OF BIOS: I/O ROUTINES
;  NORTH STAR HORIZON + IMSAI VIO + NEC PC-8023A-C
; WITH SPECIAL CONSOLE-CONTROLLED FUNCTIONS
;
        ;MACLIB REFERS TO MACRO LIBRARY FOR DIGITAL RESEARCH "MAC" ASSEMBLER
        MACLIB  Z80

        ;EQUATES
MSIZE   EQU     44
BIAS    EQU     (MSIZE-16)*1024 ;DETERMINES BIAS AND ORG FOR SYSTEM
IOBYT   EQU     3
VIO     EQU     0F803H
VINIT   EQU     0F800H
USER    ORG     0B400H

        ;JUMP TABLE - CP/M JUMPS HERE FOR I/O
        ;JUMPS MUST REMAIN HERE, IN SAME ORDER

INIT    JMP     INITR   ;INITIALIZATION
CONST   JMP     TTYST   ;CONSOLE STATUS CHECK
CONIN   JMP     TTYIN   ;CONSOLE INPUT
CONOUT  JMP     TTYOUT  ;CONSOLE OUTPUT
LIST    JMP     LST     ;LIST OUTPUT
PUNCH   JMP     TTYOUT  ;PUNCH OUTPUT
READER  JMP     RDR     ;READER INPUT

        ;BEGIN USER DRIVER ROUTINES HERE

INITR   MVI     A,85H   ;SET UP IOBYT FOR DEFAULT CONDITION
        STA     IOBYT

IBYTE   DB      1       ;BEEN HERE BEFORE BYTE
        LDA     IBYTE   ;USED TO PREVENT REINITIALIZATION ON COLD BOOT
        BIT     0,A     ;IBYTE IS 1 IF HAVEN'T INITIALIZED
        RZ              ;NO INIT WANTED BECAUSE IBYTE=0
        RES     0,A     ;BEFORE INITIALIZING, TURN OFF IBYTE
        STA     IBYTE

INITS   XRA     A       ;STANDARD NORTH STAR HORIZON INITIALIZATION
        OUT     6
        OUT     6
        OUT     6
        OUT     6
        MVI     A,0CEH
        OUT     3
        MVI     A,0CEH
        OUT     5
        MVI     A,37H
        OUT     3
        MVI     A,37H
        OUT     5
        MVI     A,30H
        OUT     6
        CALL    VINIT   ;IMSAI VIO INITIALIZATION
        RET

                                1
```

The reward for all this trouble is the control you obtain over all your peripherals, so that none of the intelligence you've paid for goes to waste.

```
;CONSOLE STATUS ROUTINE
TTYST   IN      6       ;FOLLOWING IS STANDARD NORTH STAR ROUTINE
        ANI     2
        MVI     A,0
        RZ              ;A=0 IF NO CHAR WAITS
        CMA
        RET             ;A=0FFH IF CHAR

;CONSOLE INPUT DRIVER WITH SPECIAL FUNCTION TESTS
TTYIN   CALL    TEST    ;SOFTWARE FIX FOR HARDWARE BUG IN PARALLEL INPUT
        LDA     IOBYT   ; FIRST CHECK IOBYT
        ANI     3       ;CON:=CRT IS DEFAULT, CON:=TTY ALTERNATE
        JZ      PPASS   ;IF CON:=TTY THEN JMP TO SERIAL/PARALLEL PASSTHRU

        ;FOLLOWING TURNS EXTRA KEY INTO REVERSE VIDEO FUNCTION KEY
        IN      6       ;GET CONSOLE STATUS BYTE
        ANI     80H     ;J1 BIT ON? (BIT HARDWIRED FROM KEYBOARD)
        JNZ     NEXT
        MVI     C,1BH   ;ESCAPE
        CALL    TTYOUT
        MVI     C,56H   ;V FOR REVERSE VIDEO
        CALL    TTYOUT

        ;FOLLOWING TURNS ANOTHER KEY INTO PRINTER INITIALIZATION FLAG
NEXT    IN      6       ;CHECK AGAIN
        ANI     40H     ;J2 BIT ON? (BIT HARDWIRED FROM KEYBOARD)
        JZ      PFUNCT

        ;NOW WE ACTUALLY INPUT A CHARACTER FROM CONSOLE
        IN      6       ;CONSOLE STATUS ROUTINE
        ANI     2       ;PARALLEL INPUT FLAG?
        JZ      TTYIN   ;IF NOT,LOOP
        IN      0       ;GET CHARACTER

        ;KEYBOARD NORMALLY SENDS ONLY 7 BITS--8TH BIT IS HARDWIRED TO
        ;ANOTHER UNUSED KEY AS AN ALTERNATE CHARACTER SET FUNCTION SWITCH.
        ;FOLLOWING TESTS FOR SWITCH.
        BIT     7,A     ;PARITY BIT SET?
        JNZ     NEXT2   ;NOPE, SO DISREGARD NEXT ROUTINE
        PUSH    PSW     ;SAVE 'A' REGISTER
        LDA     GRKBIT  ;GET FLAG BYTE FOR GREEK CHAR. SET (PRINTER)
        CALL    SETGRK  ;SWITCH IT ON OR OFF
        MVI     C,16H   ;SEND CONTR.V TO TURN ON/OFF CHR-BY-CHR REVERSE VIDEO
                        ;SO CRT DISPLAY INDICATES CHARACTER SET
        CALL    TTYOUT
        POP     PSW

NEXT2   PUSH    PSW     ;SAVE 'A' REGISTER WITH INPUT CHARACTER
        MVI     A,30H   ;RESET PARALLEL INPUT FLAG, NORTH STAR ROUTINE
        OUT     6
        POP     PSW
        ANI     7FH     ;STRIP PARITY BIT
        RET

;CONSOLE OUTPUT TO IMSAI VIO VIDEO BOARD
TTYOUT  MOV     A,C     ;OUTPUT TO VIO
        CALL    VIO
        RET

;LIST OUTPUT--PRINTER(S) OR PORT(S)
LST     LDA     IOBYT   ;CHECK WHICH OUTPUT DEVICE CHOSEN
        ANI     0C0H
        CPI     0C0H
        JZ      UL1     ;LST:=UL1 SETUP FOR LEFT SERIAL (300 BAUD)
        CPI     80H
        JZ      LPT0    ;DEFAULT LST:=LPT IS PARALLEL PORT
        ;LST:=TTY OR CRT USE RIGHT SERIAL (1200 BAUD)

LST1    IN      5       ;RIGHT SERIAL OUTPUT (1200 BAUD) FOR LINE PRINTER
        ANI     1
        JZ      LST1
        MOV     A,C
        OUT     4
        RET

UL1     IN      3       ;LEFT SERIAL OUTPUT (300 BAUD) FOR MODEM
        ANI     1
        JZ      UL1
        MOV     A,C
        OUT     2
        RET

;PARALLEL PORT OUTPUT WITH SEVERAL SPECIAL FUNCTION CHECKS
LPT0    CALL    FCHECK  ;CHECK FOR FUNCTION #4 PROPORT.SPACING
        MOV     A,C     ;GET CHAR.
        CPI     1CH     ;IS IT CONTROL"\"?
        LDA     GRKBIT  ;GET GREEK CHAR. SET FLAG (00 OR FFH)
        JZ      SETGRK  ;YES, THEN SET OR CLEAR GREEK CHAR. FLAG
        ANA     A       ;NOW CHECK WHETHER BYTE IS 00 OR FFH
        CNZ     WGREEK  ;PUT GREEK CHAR IN C FOR LPT TO OUTPUT

LPT     IN      6       ;CHECK PARALLEL PORT STATUS
        ANI     1
        JZ      LPT
        STA     TRACK   ;LEAVE A FLAG TO SHOW A BYTE HAS BEEN OUTPUT
                        ;(SEE LABEL 'TEST' FOR EXPLANATION)
        MOV     A,C     ;GET OUTPUT BYTE
        OUT     0       ;SEND IT
        MVI     A,20H   ;RESET STATUS BYTE
        OUT     6
        MOV     A,C     ;RETURN IT IN A FOR GOOD MEASURE
        RET

;FOLLOWING INCREASES THE SPACING BETWEEN CHARACTERS WHICH WOULD OTHERWISE
;APPEAR TOO CROWDED BY SENDING A 3-DOT LENGTH SPACE BETWEEN EACH CHARACTER
;OUTPUT, PROVIDED THAT FUNCTION #4 HAS BEEN SET FROM KEYBOARD
;FUNCBYT IS FLAG CONTAINING FUNCTION NUMBER, USED ONLY BY THIS ROUTINE

FCHECK  LXI     H,FUNCBYT       ;GET THE FUNCTION NUMBER FLAG
```

```
ACK      MVI    C,86H    ;GET ACKNOWLEDGE (PARITY BIT SET)
         CALL   LST1     ;SEND IT
         JMP    PPASS

;FOLLOWING SETS PRINTER FUNCTIONS USING FUNCTION KEY + ASCII  0-9
;IF FUNCTION KEY HAS SET A BIT IN CONSOLE STATUS BYTE, ONE JUMPS HERE
PFUNCT   IN     6        ;INPUT A NUMBER 0-9
         ANI    2
         JZ     PFUNCT
         IN     0
         CALL   NEXT2    ;RESET PARALLEL INPUT
         SUI    30H      ;CONVERT FROM ASCII TO 0-9 BY SUBTRACTING ASCII '0'
         RC              ;TOO LOW TO BE A NUMBER IF CARRY FLAG SET
         CPI    0AH      ;GREATER THAN 9?
         RNC             ;TOO HIGH
         STA    FUNCBYT  ;STASH THE FUNCTION NUMBER FOR LATER REFERENCE

         ;NOW WANT TO LOOK UP THE SERIES OF BYTES TO OUTPUT TO PRINTER
         MOV    B,A      ;FUNCTION COUNT IN B
         LXI    H,JUMPT  ;ADDRESS OF JUMP TABLE CONTAINING COUNT OF BYTES
         ANA    A        ;SET ZERO BIT IF FUNCTION= 0
         JZ     BUTT1    ;NO LOOP IF FUNCTION=0
         XRA    A        ;CLEAR 'A'
BUTT     ADD    M        ;ADD UP NO. OF BYTES TO SKIP
         INX    H        ;MOVE POINTER TO NEXT BYTE
         DJNZ   BUTT     ;LOOP IF 'B' NOT YET ZERO

         ;HL NOW POINTS TO NUMBER OF BYTES TO OUTPUT FOR FUNCTION x WHILE
         ;'A' TELLS HOW FAR TO ADVANCE POINTER FROM START OF BYTE LIST 'FUNCT'
BUTT1    MOV    B,M      ;GET NO. OF BYTES TO OUTPUT IN 'B'
         LXI    H,FUNCT  ;GET START ADDRESS OF LIST OF BYTES TO OUTPUT
         ADD    L        ;ADD LOW ORDER BYTE TO NO. TO SKIP
         ;(NOTE: ADDRESS OF ALL BYTES IN LIST MUST HAVE SAME HIGH-ORDER BYTE
         ;ELSE  NEED 16-BIT ADDITION TO COMPUTE ADDRESS)
         MOV    L,A      ;HL NOW POINTS TO BEGINNING OF OUTPUT DATA
         CALL   PRINT    ;SEND DATA TO PRINTER
         JMP    TTYIN    ;GO BACK TO TTYIN TO WAIT FOR CONSOLE INPUT

;SIMPLE PRINT LOOP (Z-80 STYLE).  LOOP UNTIL 'B' GOES TO ZERO.
PRINT    MOV    C,M      ;GET BYTE TO OUTPUT INTO 'C'
         CALL   LPT      ;SEND IT
         INX    H        ;ADVANCE POINTER TO NEXT BYTE
         DJNZ   PRINT    ;CHECK 'B', DECREMENT IT, AND LOOP IF NONZERO
         RET

;JUMPT LISTS NO. OF BYTES TO OUTPUT FOR EACH FUNCTION 0-9
JUMPT    DB     12,2,2,2,2,2,1,2,5,1
;FOLLOWING IS LIST OF CONTROL BYTES FOR PRINTER
FUNCT    DB     1BH,21H,1BH,22H,1BH,41H,0FH,1BH,4CH,30H,30H,30H
                ;NO ENHANCEMENT,1/6" LINES,NORMAL SIZE,ZERO LEFT MARGIN
FUNCT1   DB     1BH,4EH ;10 CPI PRINT MODE
FUNCT2   DB     1BH,45H ;12 CPI PRINT MODE
FUNCT3   DB     1BH,51H ;17 CPI PRINT MODE
FUNCT4   DB     1BH,50H ;PROPORTIONAL SPACING
FUNCT5   DB     1BH,21H ;ENHANCED CHARACTERS
```



THE UNKNOWN GIANT
The SMALL ONE is a highly sophisticated dependable portable computer designed for the professional. It provides versatility through S-100 hardware and compatibility with CP-M software. Typical system uses include:
Program Development
Video Image Processing
Computer Aided Design (CAD)
Spreadsheet and Word Processing
Automated Testing and Instrument Control
Adaptive Educational Testing
THE SMALL ONE
GMR INCORPORATED
1048 E. Burgrove St.
Carson, Calif. 90746
(213) 639-4663
Boston
Massachusetts
TLX 4992468
London
United Kingdom
TLX 86554
Lucerne
Switzerland
TLX 56940
© title
1983 YUMA GAMES
U.S. Patents
4294496 &
DES 263586

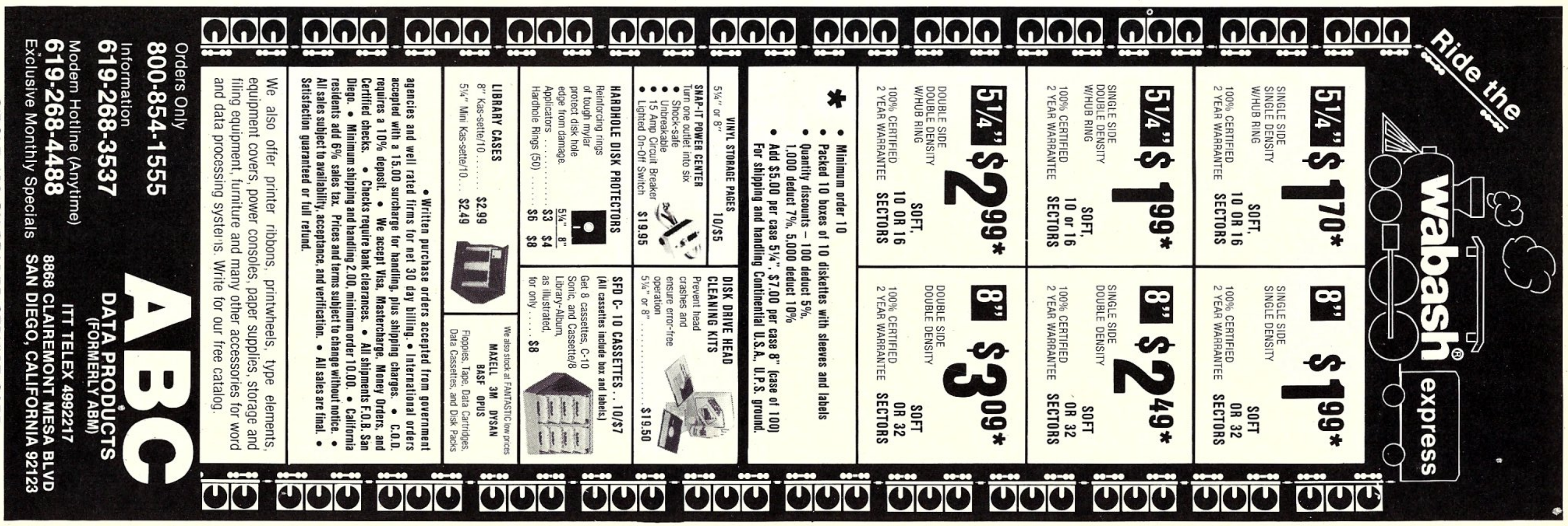
CBIOS continued . . .

```
        MOV     A,M
        CPI     4       ;IS IT FUNCTION #4?
        RNZ             ;NOT FUNCTION 4, THEN FORGET IT
        PUSH    B       ;SAVE C REGISTER WITH CHARACTER TO OUTPUT
        MVI     B,2     ;OUTPUT THE TWO BYTES FOLLOWING FUNCBYT
        INX     H       ;WHICH SET DOT-SPACING ON NEC PC-8023
        CALL    PRINT
        POP     B       ;GET CHARACTER BACK IN 'C'
        RET
FUNCBYT DB      0,1BH,3 ;LAST 2 BYTES SET 3-DOT PROPORT. SPACING


;FOLLOWING ROUTINE FIXES A HARDWARE BUG WHICH CAUSES A MISREAD OF PARALLEL
;INPUT PORT IF LAST BYTE SENT TO PARALLEL OUTPUT PORT HAS A '0' IN BIT 0
TRACK   DB      0       ;01H IF BEEN HERE LATELY, ELSE 00H
TEST    LDA     TRACK
        BIT     0,A     ;IS BIT 0 SET?
        RZ              ;NO, SO HAVEN'T USED THIS PORT SINCE LAST INIT
        MVI     C,A     ;SEND A 1 TO PARALLEL OUT TO PREVENT JAM UP
        CALL    LPT
        RES     0,A     ;CLEAR THE FLAG
        STA     TRACK   ;AND REPLACE IT
        RET

;READER INPUT ROUTINE
RDR     LDA     IOBYT
        ANI     08H
        CPI     08H
        JZ      UR1     ;IF RDR:=UR1 (USER READER), READ LEFT PORT

PTR     IN      5       ;DEFAULT IS PTR (RIGHT SERIAL PORT)
        ANI     2
        JZ      PTR
        IN      4
        MOV     C,A     ;PUT IN C FOR LPT TO FIND
        RET

UR1     IN      3
        ANI     2
        JZ      UR1
        IN      2
        ANI     127
        RET

;FOLLOWING GIVES ACCESS TO PARALLEL PRINTER TO DATA COMING FROM SERIAL PORT
;REQUIRING A COLD BOOT TO RETURN CONTROL TO CONSOLE, IF CON:=TTY
PPASS   CALL    PTR     ; GET CHAR. FROM RIGHT SERIAL PORT
        CPI     3       ;IS IT ETX? (BECAUSE OTHER COMPUTER SET UP FOR
                        ;ETX/ACK HANDSHAKING WITH PRINTER)
        JZ      ACK     ;YES, SO SEND ACKNOWLEDGMENT
        CALL    LPT     ;SEND INPUT CHARACTER TO PARALLEL OUTPUT PORT
        JMP     PPASS   ;GO BACK AND CHECK AGAIN FOR SERIAL BEFORE QUITING
```

```
FUNCT6  DB      0EH      ;ENLARGED CHARACTERS
FUNCT7  DB      1BH,42H        ;1/8" LINES
FUNCT8  DB      1BH,4CH,30H,31H,30H     ;SET 10-SPACE LEFT MARGIN
FUNCT9  DB      0BH      ;VERTICAL TAB 6 LINES

;FOLLOWING ARE ASCII CODES FOR SPECIAL CHARACTERS AND GREEK ALPHABET,
;ARRANGED IN ORDER OF ASCII CODES OF DESIRED KEYS
GREEK   DB      0ADH,0AEH,0A3H,0ACH,0DBH,0CEH,0A2H,0A5H,0DCH,0C7H,0CCH
        DB      0B1H,0B3H,0B4H,0B5H,0D4H,0D5H,0D6H,0ABH,0C8H,0A4H,0D2H
        DB      0A1H,0B0H,0AAH,0C0H,0C3H,0B2H,0BDH,0B6H,0DDH,0C5H,0C6H,0D7H,0C4H
        DB      0BEH,0DEH,0DFH,0C1H,0BCH,0CAH,0BFH,0B7H,0B8H,0CFH,0C9H,0D8H
        DB      0C2H,0DAH,0D9H,0D1H,0A7H,0A8H,0A9H,0CBH,0D9H

GRKBIT  DB      0        ;FLAG FOR GREEK-MATH SYMBOLS

;FOLLOWING IS JUMPED TO IF SWITCH GRKBIT IS SET (SEE TTYIN).  ROUTINE
;SUBSTITUTES 'GREEK' CODES FOR 'NORMAL' CODES AS INPUT FROM KEYBOARD
WGREEK  LXI     H,GREEK  ;GET POINTER TO SPECIAL CHARACTER LIST
        MOV     A,C      ;GET ASCII CHAR INPUT FROM KEYBOARD
        SUI     28H      ;FIRST CHAR IS '(' - SUBTRACT ITS CODE
        RC               ;FORGET IT IF INPUT CHAR. < 28H
        CPI     57       ;MUST BE LESS THAN THIS NUMBER
        RNC              ;NO SUBSTITUTION IF NOT
        ADD     L        ;ADD LOW ORDER BYTE OF ADDRESS TO FIND LOCATION IN LIST
        MOV     L,A      ;SET ADDRESS OF NEW CHARACTER CODE
        ;(NOTE: ADDRESS OF ALL BYTES IN LIST MUST HAVE SAME HIGH-ORDER BYTE
        ;ELSE NEED 16-BIT ADDITION TO COMPUTE ADDRESS)
        MOV     C,M      ;GET CHAR. CODE INTO PRINT REGISTER
        XRA     A        ;CLEAR 'A' FOR GOOD MEASURE
        RET

;A SIMPLE SWITCH WHICH RECEIVES A FLAG BYTE AND REVERSES IT
SETGRK  CMA              ;FFH BECOMES 00H, 00H BECOMES FFH
        STA     GRKBIT   ;SET OR RESET FLAG FOR GREEK
        RET
```

Listing 2. An alternate character set configured for easy access.

```
Normal ASCII:  ()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_
Alternate:     [illegible]
```



$1590

Full size smart terminal with detachable keyboard

4Mhz Z80A CPU
64K RAM Main Memory
200Kbyte 5¼" floppy disk
(Osborn, Xerox, IBM, CP/M-86 formats)
Two serial ports

Complete software package includes:
WordStar word processing
Correct-It spelling checker
LogiCalc electronic spreadsheet
Microsoft BASIC
NorthStar compatible BAZIC
CP/M 2.2 Operating System

At $1790, this computer was selling like hotcakes. So we dropped the price.

Crazy? No, not really. You see in order to meet the demand for the Micro Decision™ we increased our production. When we did that, our costs dropped. We're passing our savings on to you, because that's our philosophy.

More for less.

So now it only takes $1590 to buy a Micro Decision with 64K of memory, a 200K double-density floppy drive and a full-size Morrow smart terminal with detachable keyboard. Not bad. But there's more.

The Micro Decision also includes a package of business and professional software worth well over $2000. The WordStar® word processor. A 36,000 word spelling checker. The LogiCalc™ electronic spreadsheet. And both Microsoft BASIC-80® and NorthStar-compatible BAZIC® Plus, the CP/M® 2.2 Operating System that gives you access to thousands of other software programs. And, we take the mystery out of CP/M with plain English commands and single-key operation.

If you have your own terminal, you can buy the complete computer and software package for $995. That's the Micro Decision MD1™ The MD2™ includes another double-density disk drive, plus Personal PEARL™ the relational data base manager. Price? Only $1395.

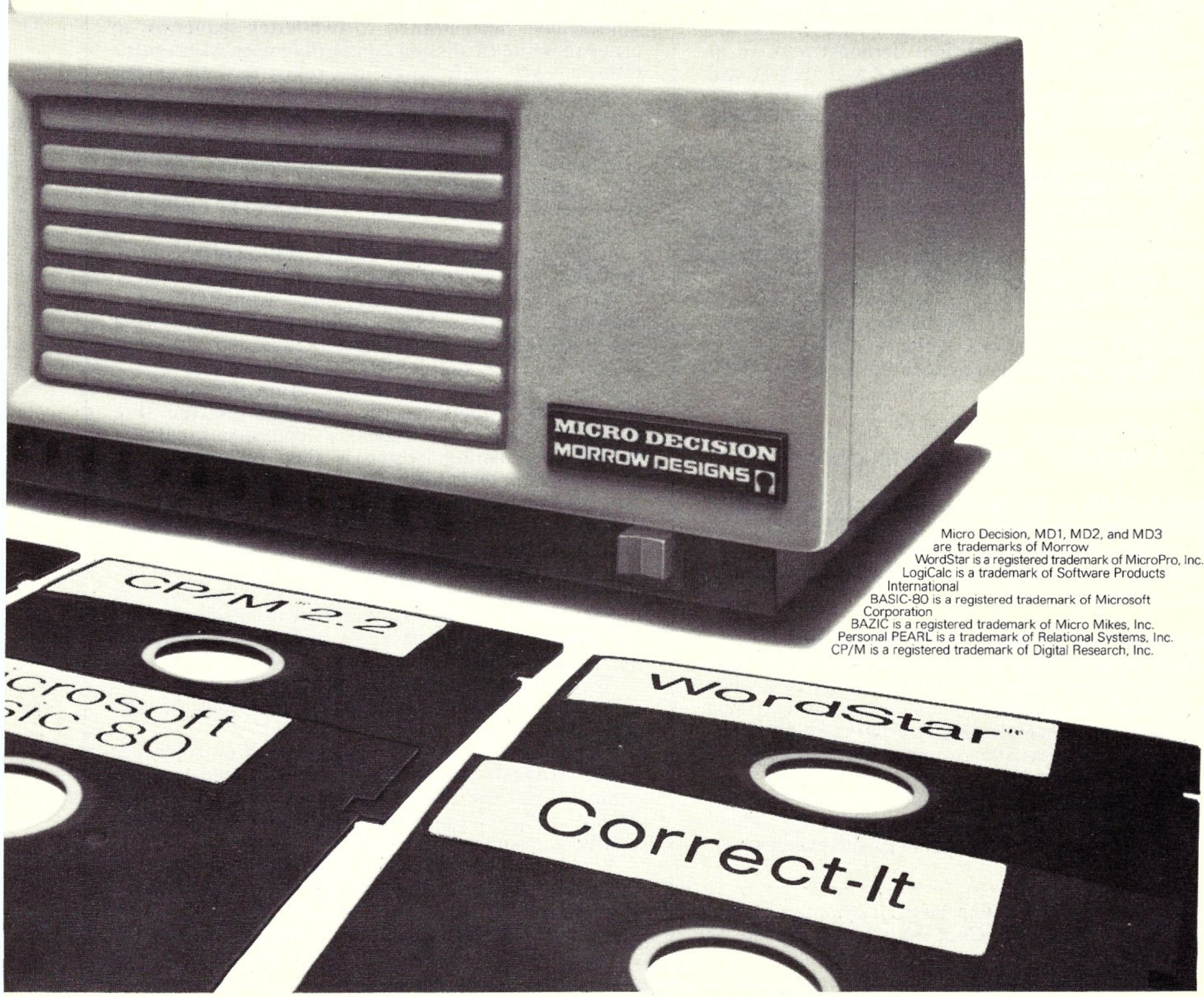
More memory? No problem. The Micro Decision MD3™ gives you two double-sided, double-density disk drives with 768K of storage and Personal PEARL for only $2490. Without the terminal, $1895.

Come in for a complete demonstration at your nearest Morrow dealer. If you don't know who that is, call us toll-free at (800) 521-3493. In California, call (415) 430-1970. At $1790, the Micro Decision sold like hotcakes. At $1590, we've just sweetened the deal.

MORROW

MORROW, 600 McCormick Street, San Leandro, CA 94577
(800) 521-3493, (415) 430-1970 In California

CIRCLE 280 ON READER SERVICE CARD



Micro Decision, MD1, MD2, and MD3 are trademarks of Morrow
WordStar is a registered trademark of MicroPro, Inc.
LogiCalc is a trademark of Software Products International
BASIC-80 is a registered trademark of Microsoft Corporation
BAZIC is a registered trademark of Micro Mikes, Inc.
Personal PEARL is a trademark of Relational Systems, Inc.
CP/M is a registered trademark of Digital Research, Inc.

A General-Purpose Graphic Plotting Package

Part 2: Installation and linking

by David H. Freese, Jr.

Extending high-level languages to dot-matrix printers is possible with a simple plotting interface, which was described in Part 1. In Part 2, I will discuss how to install the plotting package, as well as how to link it to Basic, Pascal/Z, and JRT Pascal. This article concludes the series.

Using the plot functions

I will illustrate how each plot function is invoked from both Basic and Pascal through two programs (listings for the Basic program are given here; the Pascal listings will be given next month). The first is simply a demonstration program that plots axes, grid marks, data points, lines, and circles. The second is more useful and in fact demonstrates an interesting attribute of both Pascal/Z and JRT Pascal. This program generates a Bode diagram (log amplitude vs. the log frequency) of an abitrary transfer function.

The Pascal/Z programs were compiled and linked using the utilities furnished by Ithaca Intersystems, Inc. The resulting .COM file is executed from the CP/M command line. Linkage to the plot functions is accomplished during the link phase, with the required code being obtained from PLOT.REL.

The JRT Pascal programs were compiled and executed using version 2.2. The listings should be compatible with later versions of the compiler.

The Basic programs are executed from the CP/M command line as follows:

```
A>PLOT<CR>
A>MBASIC NAME/M:&HNNNN<CR>
```

where NAME is the Basic program to be executed, and NNNN is the hex value of ORIGIN. This will save memory, starting at NNNN.

The graphic output from these two programs is shown in Figures 1 and 2.

Description of plot test program

Basic version. This program has formed the basis of several engineering analyses that required graphic output. Basic has the advantage of quick program development, but develops sloppy programming habits. I have tried to be as structured as the language will allow in this program.

Lines 100-190 define the USR function addresses. Note that the relative address of each of the individual plot functions is independent of the origin selected for PLOT.COM It is therefore necessary to change only the assignment of the variable BS on line 110 for a different origin.

The plotting variables are declared and initialized on line 340, and the x and y scale values evaluated by the subroutine call to line 710.

The two functions FNXP%() and FNYP%() are designed to convert a floating point number to a scaled and translated integer. These two functions provide a mapping between the x,y parameters and those required by the assembly code.

Plotting functions are defined in lines 440-630, which extend the Basic machine code USR call to a multiple parameter list. The variable and function names have been limited to two-letter sequences in order to make the listing compatible with Microsoft's OBasic. (Microsoft sometimes refers to their earlier version (4.5) of Basic as OBasic, as opposed to the current version (5.4), which they refer to as MBasic.) Running under OBasic provides the benefit of some additional memory. In a 64K environment, you will have approximately 7K of free memory for Basic code when both the interpreter and PLOT.COM are resident.

Line 660 defines some Heath H89 screen control functions that I generally put in all my programs.

Lines 10-750 are the meat and—from 1000 on—the potatoes of this program. The benefit of defining the plotting function calling sequences is readily apparent from the simplicity of the latter code.

Pascal versions. The test program demonstrates the differences between the Pascal/Z and the JRT Pascal versions. The Pascal/Z compiler creates a source code listing, which is then assembled and linked with external functions and library functions. During the link phase all unresolved externals are searched for in specified .REL files. During the assembly of the plotting package, each external function name is declared to be a global entry point. The linker therefore recognizes each function name to be unique. The array that contains the image is a part of the external plotting package, and no declaration of an image variable is necessary.

When linked the Pascal/Z program to the external functions, I suggest the following procedure to conserve disk space:

1. Set up the LINK command with no reference to the PLOT.REL file. The linker will load the program, search the library file for externals, and then declare a list of unresolved externals—all of which will be contained in the PLOT.REL file.

2. At this point give the LINK utility the name of the PLOT.REL file and specify the /E parameter. Forcing the LINK utility to search the .LIB file first and the PLOT.REL file last will result in a savings of 21.5K of file space on each .COM file created. The

David H. Freese, Jr., Clermont Computer Consultants, R.D. 1, Box 316, Cape May Court House, NJ 08210

image array is defined in the assembly listing with the DEFS (DS) equate at the end of the assembly code. Using the above sequence places the array at end of the link process. The utility LINK provided by Ithaca Intersystems is intelligent enough to recognize this and terminates the .COM file at the beginning of the array. If the PLOT.REL file had been searched before the .LIB file, then the array would be in the middle of the .COM file and would waste considerable disk space.

JRT Pascal is a P-code interpreter. The compile process creates an intermediate source code file with the extension .INT. The intermediate files may be either P-code or machine code. The contents of the .INT file are specified in a header. The header is defined by the compiler for Pascal source code, and by executing a program called CONVERTM for Microsoft-compatible .REL files. External procedures are brought into memory on a dynamic basis, and the interpreter will purge unused procedures when memory requirements so dictate. This would produce unpredictable results if the image array had been made a part of the external plotting package. That array is therefore declared in the main program. The external plot procedure gains access to the array by its position in the global variable structure.

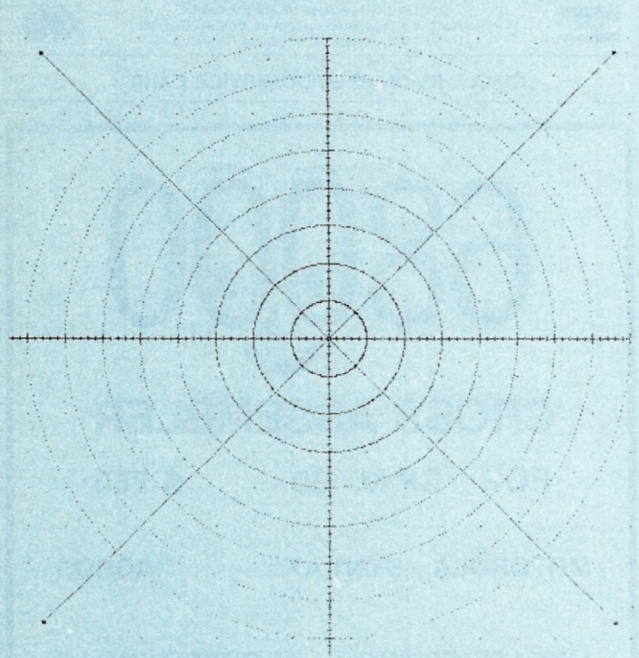


Figure 1. *Demonstration program that plots axes, grid marks, data points, lines, and circles.*

The type declaration of **pline** and **plt** produces an array of char that is 21,476 bytes long. Alternatively, a single type could have been declared to be: array [1..21476] of char. The variable, image, of type plt, is the first one to be declared in the program. It is therefore at a known position in the global variable allocation. The array will begin six bytes above the base address. The base address is contained in the HL register pair, and the pointer to the dynamic data stack, current, in the DE register pair. This is sufficient to gain access to both global and passed parameters.

Since the interpreter searches for external procedures/functions by file name, only a single external plot procedure is declared. Access to the individual subroutines is obtained by a command parameter (cmd) of type char. It is always necessary to pass eight parameters on every call to the external plot procedure—the command parameter and seven integer values. These are passed by value and not by reference.

Description of program: Bode

While the test program demonstrates the declaration and calling of the external plotting functions from the three high-level languages, it does not provide the insight needed to use them in an engineering application. The program, Bode, is a real electrical engineering application. It illustrates how the plotting package is used to provide sophisticated graphic representation of a circuit analysis.

The concept of the BODE plot is familiar to those of us educated in the electrical engineering field. If you are unfamiliar with the term, it can be thought of as the amplitude (and phase) response of any system as a function of frequency. The most common illustration is that of a high-fidelity amplifier. Quite often manufacturers will include a graph of the components amplitude versus frequency response. Since both amplitude and frequency values can range over several orders of magnitude, it is convenient to plot them both on a logarithmic scale. The amplitude scale is that of decibels, and the frequency scale that of the logarithm, base 10 of frequency.

When analyzing an electrical circuit that contains reactive components, inductors, and capacitors, it is usual to use either Fourier or Laplace transform representation of the network elements. I have elected to use the Laplace notation for the purpose of this program and the network description. In either case it is necessary to be able to describe complex numbers and perform complex number arithmetic. Both JRT Pascal and Pascal/Z are to be commended in their implementation of the language. Quite a few versions of the Pascal language allow only a function to return a scalar value. JRT Pascal and Pascal/Z both allow any declared variable type to be returned from a function call.

Both JRT Pascal and Pascal/Z are to be commended on their implementation of the language. They both allow any declared variable type to be returned from a function call.

Complex variables are created with a type complex that is a record consisting of two parts—the real and imaginary parts of a complex number—both of which are of type real. The following complex number arithmetic functions are included in the program:

- CSUM—sum of two complex numbers
- CDIFF—difference of two complex numbers
- CPROD—product of two complex numbers
- CMAG2—magnitude squared
- CMAG—magnitude
- CDIV—quotient of two complex numbers
- POLAR—rectangular-to-polar conversion of a complex number
- RECT—polar-to-rectangular conversion of a complex number

The system transfer function is declared as a procedure TRANSFER. This is the only code that needs to be altered for evaluation of a different system. In addition the procedure TRANSFER could be made an external procedure. If it is made an external procedure, it should be the first one declared. The program BODE can then be compiled, linked, executed, etc., only once. The new external TRANSFER function can be modified, compiled and linked, with a considerable saving in compile time for each new system analysis.

The Basic version of this program is considerably different from the Pascal versions. I have tried to retain as much similarity in structure as possible. Basic supports only the return of a scalar value from a function call. It is therefore necessary to make the complex number arithmetic into subroutine calls. The complex number representation is obtained with two floating point number arrays, R() and I(), the real and imaginary respectively. Return values are already contained in R(0) and I(0). R(1)-R(6) and I(1)-I(6) are temporary storage locations. S(0) and S(1) are the real and imaginary components of the Laplace complex frequency.

During execution, all the programs perform identically. Their speeds are indicative of the type of execution mode. Pascal/Z provides the fastest speed, since it is machine code after the compile and link process. Pascal/Z is rather unforgiving when errors are made in data entry from the keyboard. Be sure you observe datatype formats when entering the requested data. JRT Pascal is an interpretive language, as is MBasic. They both are considerably slower than the Pascal/Z version. The floating point numbers in MBasic are 4-byte binary mantissas and exponent values with a precision range of approximately 6½ digits. The JRT Pascal real numbers are all 14-digit binary coded decimal values. The transcendental functions of JRT Pascal are all .INT files compiled from Pascal source code. In spite of the increased range of precision and the nonmachine-code implementation of the math functions, JRT Pascal was still faster than the Basic interpreter. For $29.95, this has got to be the best software buy of the century.

How to install the plot package

The following utilities are required in order to assemble, load, and execute the plot package:

M80— Relocating macro assembler
L80 — Linking loader
DDT— Dynamic Debugging Tool

Proceed as follows:

1. Edit the source code equates to reflect your actual operating system, plotter, and plot size requirements. If you are short on disk space, I suggest you eliminate all remarks in the source code.
2. Assemble the source code using Microsofts's M80 or a compatible assembler. Either create a PRN file or assemble with output to a list device. You will need to have access to the value assigned to the symbol ORIGIN.
3. On a CP/M command line type:

```
M80 PLOT,PLOT=PLOT <CR>
```

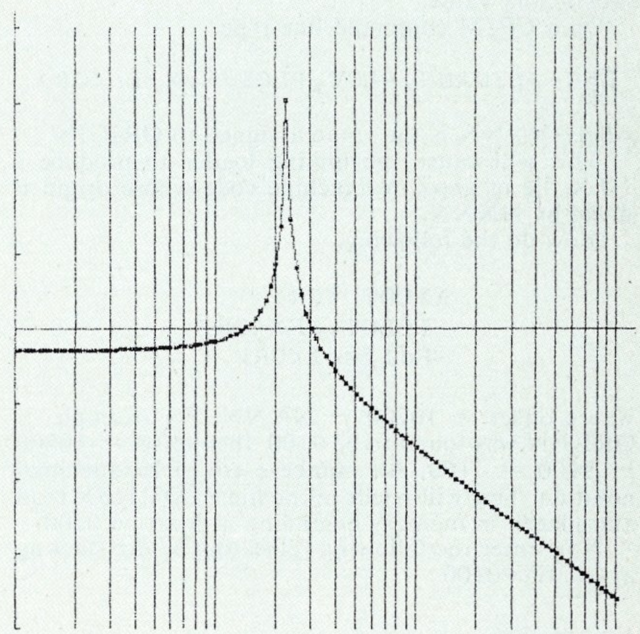


Figure 2. *Bode diagram (log amplitude vs. log frequency) if an arbitrary transfer function.*

The BODE program is an example of a real electrical engineering application. It illustrates how the plotting package is used for a sophisticated graphic representation of a circuit analysis.

Linking to Pascal/Z

If you are going to link the plot functions to Pascal/Z, you are finished. Pascal/Z expects to find external procedures in a REL file.

Linking to JRT Pascal

The JRT Pascal interpreter expects to find external procedures in .INT files with the proper header attached. The JRT package is furnished with its own assembler (which is actually a Pascal program) for assembling 8080 code. This assembler produces the .INT file directly. A program for converting .REL files to .INT files is also provided. This program, CONVERTM, is executed with the PLOT.REL as its object file. It will produce a new file called PLOT.INT. No further linkage is needed.

Linking to Basic

If you are going to link the plot functions to Basic, then several more steps are necessary. The first is to generate an absolute form of machine code that is located at the correct location in memory. Jot down the value assigned to the symbol ORIGIN as found in the PLOT.PRN file. This represents the highest location in memory to which the PLOT routines can be placed. You may elect to locate the plot package lower (if for example you had other code you did not want to overlay). I will assume that ORIGIN is a satisfactory value.

On a CP/M command line type:

```
L80 /p:NNNN,PLOT,PLOT/N/X/E <CR>
```

where NNNN is the value assigned to ORIGIN.

This will cause the linking loader to produce a HEX file of absolute machine code whose origin is given by NNNN.

Now do the following:

```
A>DDT <CR>
-IPLOT.HEX<CR>
-Roffset<CR>
```

where offset = 10200 − NNNN. For example, if ORIGIN was found to be 9100, then offset = 10200 − 9100 = 7100. All numbers are in hexadecimal notation. This will result in the file PLOT.HEX read and placed in memory beginning at location 0200.

Now enter the following relocating loader starting at location 0100.

```
LD    HL,0200
LD    DE,NNNN          ; where NNNN = ORIGIN
LD    BC,TMEM-NNNN     ; where TMEM = DF00 for Lifeboat
                       ;       TMEM = DC00 for others
LDIR
RET
```

Since DDT does not support Zilog Z80 mnemonics, use the following under DDT:

```
-A0100 <CR>
0100LXI H,0200<CR>
0103LXI D,NNNN<CR>
0106LXI B,TMEM-NNNN<CR>
0109<CR>
-S0109<CR>
0109xxED<CR>
010AxxB0<CR>
010BxxC9<CR>
010Cxx<CR>
-G0<CR>
A>SAVEqqPLOT.COM<CR>
```

where qq is the number of pages of memory occupied by PLOT.COM. The value of qq can be determined as follows:

```
qq = (TMEM - NNNN)/256 + 1
```

That completes the installation of PLOT.COM. When you execute PLOT on a CP/M command line, the plot functions will automatically load, be relocated to ORIGIN, and control will be passed back to the CP/M command control processor.

Source code for the complete plotting package is available from the author in the following formats:

8″ —SS, SD CP/M format
5¼″—SS, SD CP/M Osborne format
5¼″—SS, SD: SS, DD CP/M CDR Systems compatible
5¼″—SS, HS Heath/Zenith format

The cost is $15 for all formats, which includes shipping. Orders must be prepaid by check or money order.

Dave Freese is an electrical engineer, presently employed as a senior engineering analyst by Sanders & Thomas, Inc. He is the owner of Clermont Computer Consultants, a small company specializing in real-time control applications. He holds a B.S. and M.S. in electrical engineering. Dave recently retired from a 20-year career in the U.S. Coast Guard, where his duties included an assignment as head of the electrical engineering department at the Coast Guard Academy. Just prior to retirement he was chief project engineer at the Coast Guard's electronics engineering center in Wildwood, NJ.

During execution, all the programs perform identically. Pascal/Z provides the fastest speed, since it generates machine code. JRT Pascal and MBasic are interpretive languages and run slower.

```
10 '************************************************************************
12 '*    PLOT.BAS
13 '*
14 '*    AUTHOR: D. H. Freese Jr.
15 '*            Clermont Computer Consultants
16 '*
17 '************************************************************************
18 '
19 '
100 CLEAR 400
110 BS=&H8100               ' base address of plotting routines
120 DEF USR0=BS+&H152       ' draw line
130 DEF USR1=BS+&HE7        ' draw data mark
140 DEF USR2=BS+&H73        ' draw pixel
150 DEF USR3=BS+&H65        ' initialize plot image
160 DEF USR4=BS+&H633       ' transfer image to printer
170 DEF USR5=BS+&H45F       ' draw axis
180 DEF USR6=BS+&H24D       ' draw circle
190 '
200 ' Definition of variables used in PLOT
210 '    XA - x minimum
220 '    XB - x maximum
230 '    YA - y minimum
240 '    YB - y maximum
250 '    XT - minor tick interval on x axis
260 '    YT - minor tick interval on y axis
270 '    XM - # of minor ticks between major ticks on x axis
280 '    YM - # of minor ticks between major ticks on y axis
290 '    XØ,YØ - axis center values
300 '    X1,X2,Y1,Y2 - x and y data values
310 '    R   - radius of circle
320 '    GR - flag for plotting grid marks on image
330 '
340 XØ=Ø:YØ=Ø:XT=2:YT=2:XM=10:YM=10:XA=-200:XB=200:YA=-200:YB=200:GR=1:
^@    GOSUB 710:' default values
350 '
360 ' Scaling functions to convert (X,Y) values to integer
^@     X and Y may take on values in the following ranges:
^@               XA <= X <= XB
^@               YA <= Y <= YB.
370 DEF FNXP%(X)=SGN((X-XA)*XS)*INT(ABS((X-XA)*XS) + .5)
380 DEF FNYP%(Y)=SGN((Y-YA)*YS)*INT(ABS((Y-YA)*YS) + .5)
390 '
400 ' Functions used to extend the USR call with a list of passed
410 ' parameters contained in a string variable.  The string is formed
420 ' by concatenating bytes using the MKI$ function in a manner similar
430 ' to that used in random file data storage.
440 ' plot a pixel at the scaled position (x,y)
450 DEF FNPL(X,Y)=LEN(USR2(MKI$(FNXP%(X))+MKI$(FNYP%(Y))))
460 '
470 ' plot a data mark at the scaled position (x,y)
480 DEF FNCH(X,Y)= LEN(USR1(MKI$(FNXP%(X)) + MKI$(FNYP%(Y))))
490 '
500 ' draw a line between the scaled positions (x1,y1) and (x2,y2)
510 DEF FNLN(X1,Y1,X2,Y2)=
^@     LEN(USR0(MKI$(FNXP%(X1)) + MKI$(FNYP%(Y1)) +
^@              MKI$(FNXP%(X2)) + MKI$(FNYP%(Y2))))
520 '
530 ' clear the plotting image -- all pixels off
540 DEF FNCL=USR3(Ø)
550 '
560 ' transfer the image to the printer
570 DEF FNTR=USR4(Ø)
580 '
590 ' Draw axis centered at scaled positon (XØ,YØ)
^@     with minor ticks at intervals XT and YT
^@     and major ticks at intervals XM and YM.
^@     Plot grid marks if GR<>Ø
600 DEF FNAX(XØ,YØ,XT,YT,XM,YM,GR)=
^@      LEN(USR5(MKI$(FNXP%(XØ)) + MKI$(FNYP%(YØ)) +
^@               MKI$(CINT(XT*XS)) + MKI$(CINT(YT*YS)) +
^@               MKI$(XM*CINT(XT*XS)) + MKI$(YM*CINT(YT*YS)) +
^@               MKI$(GR)))
610 '
620 ' Plot a circle centered at (X1,Y1) with radius R
^@     using the X scale for both ordinated
630 DEF FNCR(X1,Y1,R)=
^@     LEN(USR6(MKI$(FNXP%(X1) + MKI$(FNXP%(Y1) + MKI$(FNXP%(R))))
640 '
650 ' Commonly used H89 screen escape sequences and functions
660 ES$=CHR$(27):  CL$=ES$+"E":  ER$=ES$+"K":
^@    DEF FNXY$(X,Y)=ES$+"Y"+CHR$(31+X)+CHR$(31+Y):
^@    DEF FNCN$(D$,L)=FNXY$(L,40-LEN(D$)/2)+D$
670 '
680 GOTO 1000
690 '
700 ' Subroutine to evaluate the scaling values XS and YS
710 XS=400/(XB-XA): YS=400/(YB-YA)
720 IF XT<>Ø THEN XS=SGN(XT*XS)*INT(ABS(XT*XS) + .5)/XT
730 IF YT<>Ø THEN YS=SGN(YT*YS)*INT(ABS(YT*YS) + .5)/YT
740 RETURN
750 '
1000 ' Plot a bullseye test pattern on the graphics printer.
1010 '
1020 PRINT "Clearing image"
1030    Z=FNCL
1040 PRINT "Plotting axis"
1050    Z=FNAX(XØ,YØ,XT,YT,XM,YM,GR)
1060 PRINT "Plotting cross hair lines"
1070    Z=FNLN(-190,-190,190,190)
1080    Z=FNLN(-190,190,190,-190)
1090 PRINT "Plotting data marks on line ends"
1100    Z=FNCH(-190,-190)
1110    Z=FNCH(-190,190)
1120    Z=FNCH(190,190)
1130    Z=FNCH(190,-190)
1140 PRINT "Plotting concentric circles"
1150 FOR I=1 TO 8
1160    Z=FNCR(Ø,Ø,25*I)
1170 NEXT I
1180 PRINT "Transfering image to printer"
1190    Z=FNTR
1200 END
```

```
10 '*********************************************************
11 '*
12 '*    BODE.BAS
13 '*
14 '*    AUTHOR: D. H. Freese Jr.
15 '*            Clermont Computer Consultants
16 '*
17 '*********************************************************
18 '
19 '
100 CLEAR 400
110 BS=&H8100
120 DEF USR0=BS+&H152
130 DEF USR1=BS+&HE7
140 DEF USR2=BS+&H73
150 DEF USR3=BS+&H65
160 DEF USR4=BS+&H633
170 DEF USR5=BS+&H45F
180 DEF USR6=BS+&H24D
190 '
200 DEF FNPL(X,Y)=LEN(USR2(MKI$(CINT(X))+MKI$(CINT(Y))))
210 '
220 DEF FNCH(X,Y)=LEN(USR1(MKI$(CINT(X))+MKI$(CINT(Y))))
230 '
240 DEF FNLN(X1,Y1,X2,Y2) =
```

```
^@      LEN(USRØ(
^@      MKI$(INT(X1)) + MKI$(INT(Y1)) +
^@      MKI$(INT(X2)) + MKI$(INT(Y2))))
25Ø '
26Ø DEF FNCL=USR3(Ø)
27Ø '
28Ø DEF FNTR=USR4(Ø)
29Ø '
3ØØ DEF FNAX(XØ,YØ,XT,YT,XM,YM,GR)=
^@      LEN(USR5(
^@      MKI$(INT(XØ)) + MKI$(INT(YØ)) +
^@      MKI$(INT(XT)) + MKI$(INT(YT)) +
^@      MKI$(INT(XM)) + MKI$(INT(YM)) +MKI$(GR)))
31Ø '
32Ø DEF FNCR(X1,Y1,R1)=
^@      LEN(USR6(
^@      MKI$(INT(X1)) +
^@      MKI$(INT(Y1)) +
^@      MKI$(INT(R1))))
33Ø '
34Ø     ES$=CHR$(27):  CL$=ES$+"E":  EP$=ES$+"K":
^@      DEF FNXY$(X,Y)=ES$+"Y"+CHR$(32+X)+CHR$(32+Y):
^@      DEF FNCN$(D$,L)=FNXY$(L,40-LEN(D$)/2)+D$
35Ø '
36Ø DEF FNLG(X)=LOG(X)/2.3Ø259
37Ø '
1ØØØ DIM S(1),T(1),R(1Ø),I(1Ø): PI=3.14159
1Ø1Ø GOTO 2ØØØ
1Ø2Ø 'csum
1Ø3Ø R(Ø)=R(1)+R(2): I(Ø)=I(1)+I(2): RETURN


1Ø4Ø 'cdiff
1Ø5Ø R(Ø)=R(1)-R(2): I(Ø)=I(1)-I(2): RETURN
1Ø6Ø 'cprod
1Ø7Ø R(Ø)=R(1)*R(2)-I(1)*I(2): I(Ø)=R(1)*I(2)+R(2)*I(1): RETURN
1Ø8Ø 'cmag2
1Ø9Ø R(Ø)=R(1)*R(1)+I(1)*I(1): I(Ø)=Ø: RETURN
11ØØ 'cmag
111Ø GOSUB 1Ø8Ø: R(Ø)=SQR(R(Ø)): RETURN
112Ø 'cdiv
113Ø R(3)=R(1): I(3)=I(1): R(4)=R(2): I(4)=I(2):
^@     R(1)=R(2): I(1)=I(2): GOSUB 1Ø8Ø: R(5)=R(Ø):
^@     R(1)=R(3): I(1)=I(3): R(2)=R(4): I(2)=-I(4): GOSUB 1Ø6Ø:
^@     R(Ø)=R(Ø)/R(5): I(Ø)=I(Ø)/R(5): RETURN


114Ø '
2ØØØ PRINT CL$;:
^@     INPUT"# of decades in frequency ..... ";NDEC
2Ø1Ø INPUT"# of points/decade in plot .... ";N: NPTS = N*NDEC
2Ø2Ø INPUT"minimum frequency to plot ..... ";F1
2Ø3Ø INPUT"maximum decibel value ......... ";DBMAX
2Ø4Ø INPUT"minimum decibel value ......... ";DBMIN
2Ø5Ø DECØ = FNLG(F1)
2Ø6Ø DELDEC = NDEC/NPTS
2Ø7Ø DBSCALE = 4ØØ/(DBMAX - DBMIN)
2Ø8Ø FSCALE = 4ØØ/NDEC
2Ø9Ø PRINT:PRINT"Clearing image"
21ØØ     Z = FNCL
211Ø PRINT"drawing axis"
212Ø     Z = FNAX(Ø,INT(DBSCALE*(DBAXIS-DBMIN)),
^@               Ø,Ø,
^@               Ø,INT(DBSCALE*2Ø),Ø)
213Ø PRINT"placing logarithmic grid"
214Ø FOR I = Ø TO NDEC -1
215Ø FOR J = 2 TO 1Ø
216Ø K = INT(FSCALE*(I + FNLG(J))):
^@     Z = FNLN(K,4ØØ,K,Ø)
217Ø NEXT J,I
218Ø F = 1Ø^DECØ
```

Graphic Plotting Package
continued . . .

```
2190 S(Ø)=Ø: S(1)=2*PI*F: GOSUB 3ØØØ
22ØØ R(1)=T(Ø): I(1)=T(1): GOSUB 11ØØ: DBØ=2Ø*FNLG(R(Ø))
221Ø IX1 = INT(FSCALE*(FNLG(F) - DECØ)):
^@      IY1 = INT(DBSCALE*(DBØ - DBMIN))
222Ø PRINT CL$;"Bode Computations"
223Ø FOR I = Ø TO NPTS
224Ø F = 1Ø^(I*DELDEC + DECØ)
225Ø S(Ø)=Ø: S(1)=2*PI*F
226Ø GOSUB 3ØØØ
227Ø R(1)=T(Ø): I(1)=T(1): GOSUB 11ØØ: DB=2Ø*FNLG(R(Ø))
228Ø PRINT FNXY$(2Ø,Ø);USING"Freq: ####.###   dB: ###.##";F,DB
229Ø IX2 = INT(FSCALE*(I*DELDEC)):
^@      IY2 = INT(DBSCALE*(DB - DBMIN))
23ØØ     Z = FNLN(IX1,IY1,IX2,IY2)
231Ø     Z = FNCH(IX2,IY2)


232Ø IX1 = IX2: IY1 = IY2
233Ø NEXT I
234Ø     Z = FNTR


235Ø '
236Ø END
3ØØØ ' transfer function definition
301Ø R(1)=S(Ø): R(2)=S(Ø): I(1)=S(1): I(2)=S(1): GOSUB 1Ø6Ø:
^@      R(1)=R(Ø): I(1)=I(Ø): R(2)=2: I(2)=Ø: GOSUB 1Ø2Ø
3Ø2Ø R(6)=R(Ø): I(6)=I(Ø):
^@      R(1)=S(Ø): I(1)=S(1): R(2)=.Ø2: I(2)=Ø: GOSUB 1Ø6Ø:
^@      R(1)=R(Ø): I(1)=I(Ø): R(2)=R(6): I(2)=I(6): GOSUB 1Ø2Ø
3Ø3Ø R(7)=R(Ø): I(7)=I(Ø):
^@      R(1)=S(Ø): I(1)=S(1): R(2)=1: I(2)=Ø: GOSUB 1Ø2Ø:
^@      R(1)=R(Ø): I(1)=I(Ø): R(2)=R(7): I(2)=I(7): GOSUB 112Ø:
^@      T(Ø)=R(Ø): T(1)=I(Ø)
3Ø4Ø RETURN
```

SOFTWARE REVIEW

PL/I-86 and PL/I-80 Version 1.4

by Bruce H. Hunter

Digital Research is about to release its newest version of PL/I-80 version 1.4, which will be followed shortly by the 16-bit implementation, PL/I-86. Both versions have the new features of double precision, and file and record locking. PL/I-80 is a reasonably full implementation of the ANSI general-purpose subset G.

I am completing a book on PL/I subset G, so I requested that Digital Research forward me an advance copy of the documentation of both releases. The documentation is a big surprise. It is well written and reasonably complete. In the last version (PL/I-80 1.3), Digital furnished three documents with their PL/I compiler, the Language Manual, the Applications Guide, and the Link-80 Manual. Digital is making a strong effort to rid itself of the reputation shared by so many software publishers of having obscure documentation. This effort really shows in the new Language Manual, which has been completely rewritten. The features of the language are described at length in great detail. The organization of the language, program structure, blocks, scope of variables, parameters, etc., are all defined and discussed, with user readability and clarity being stressed. The manual is also well organized. Besides the expected table of contents, it has the welcome additions of an index and a glossary. The manual has over 200 pages, and it is the most complete and readable document of its type I have in my library.

The Applications Guide has been substantially appended. The version 1.3 guide was 179 pages of dense technical prose. The new Applications Guide is nearly 300 pages of informative, much better written material. It parallels the Language Manual and explains in detail the features of this version of subset G. Numerous programming examples are included, most of which are carryovers from the old Applications Guide. An interesting feature is the addition of vertical ruler lines to the programming examples that clearly define the programs' block structures. Programmers who are new to Algol-like block-structured languages will appreciate this aid in the understanding of program organization in this kind of language. While the old Applications Guide programs were a bit difficult to understand in that they tended to cover too much ground for the beginning PL/I programmer, the programs are essentially the same, but more work has been put into explaining them. This should also prove helpful to users.

Subjects covered in the Applications Guide include:

- Introduction and discussion of PL/I as a programming language
- Declarations covering scalar data and data aggregates

Bruce H. Hunter, 1020 S. Jenifer, Glendora, CA 91740

- Executable statements incuding assignment, sequence control, I/O and file handling, condition processing, memory management, preprocessors, and nulls
- Programming style including case, indentation, and modular format
- Using the system, including compiler usage and operation, error messages and codes, and PL/I system files.
- Datatypes
- Stream and record file processing
- Label constants, variables, and parameters
- Condition processing
- Character string processing
- List processing
- Recursive processing
- Separate compilation
- Decimal computations
- Commercial processing
- Internal data representation
- Interface conventions
- Dynamic storage and stack routines

The documentation package is completed by the Link-80 or Link-86 Manual (depending on which PL/I).

PL/I-80 and PL/I-86 are subsets of the full set of PL/I. They are based on subset G, but they are not full implementations of that subset either. Nevertheless, they are extremely powerful implementations. The major divergence from the full set is the absence of ISAM (Indexed Sequential Access Method) and VSAM (Virtual Storage Access Method) files. ISAM and VSAM as a built-in feature provide the data-handling ability that has made PL/I one of the most powerful data-handling languages in existence. The absence of these two filetypes can be circumvented by the use of Digital Research's Access Manager, a B+ tree system for the creation of database-managed systems, in conjunction with the PL/I-80 and 86 compilers. (Access Manager is well implemented, and the database it creates can not only be accessed by PL/I-80 and PL/I-86, but also by CB-80 and Pascal MT+ as well.)

All datatypes are implemented in PL/I-80 version 1.4 and PL/I-86, with the exception of Float Decimal. A new feature is double precision. Version 1.3's major failing as a number cruncher was its lack of double precision float. Now this no longer need be a drawback. In anticipation of the usage of the 8087 math processor, PL/I-86 has incorporated simulation routines that will give a precision to 53 (binary) instead of the 24 maximum of version 1.3. The range of the exponent has increased to a range of −308 to +308. This actually exceeds the precision and range of Fortran 66. The eventual incorporation of the 8087 will expand precision to 64 (binary) and a 15-bit biased exponent. With or without the 8087, this is one of the most potent math processing languages available today for use on a micro.

Filetypes still include both stream (sequential) and record (random). All of PL/I's key and keyed features are coupled, making the use of file keys or indices available to the programmer. The READ with KEYTO statement allows the file key to be extracted from an input file and stored sequentially to form any sort of a data structure that the programmer is capable of creating.

Digital Research (CP/M and MP/M) has come to grips with the fact that PL/I will have to exist within an MP/M environment. The need for file locking and record locking in a multiuser environment has been met. Password protection is incorporated in the file open statement. The password becomes part of the filename as in

```
B:secret.fle;j/bond
```

File attributes are predefined within the open, and various levels of protection are offered. Protection levels are

```
read
write
delete
```

The environment attributes of the file have been expanded to have the file either locked, shared, or read-only. With the read-only attribute, more than one user can address the file. Locked (the file default) keeps the file local to one user. Shared or unlocked allows multiple access. Additionally, functions are provided for the locking and unlocking of records (as opposed to files).

It is noteworthy that Digital has created essentially one PL/I in two implementations. The manual applies to both the 8- and 16-bit versions. Total compatibility appears to be guaranteed. There are no inconsistencies with version 1.3, so upward compatibility should be no problem. The picture attribute is still supported, as are all the other features of 1.3.

The operation of the compiler PLI.COM remains essentially unchanged as well as the linker LINK.COM. The compiler is not the fastest compiler ever created, but considering the size of it and the size of the average PL/I program, it is not surprising. The linker and compiler are both easy to use and versatile. Command lines for simple programs are equally simple. Typical invocations are

```
pli b:myprog
link myprog
```

It's just that simple. Large programs take about three to four minutes to link on a Godbout 8085/8088, using Qume DT8's with direct memory access (DMA). Numerous switches and options are available on both the compiler and linker. Errors found during compilation are reasonably well explained. Like any compiler, errors are not always what the compiler says they are. It is not as friendly as CBasic or Pascal Mt+. No debugger is provided, although one would certainly be welcome.

There had been a bug in version 1.3 that involved the use of the GET EDIT statement. A sequence like

```
put skip list ('name : ');
get edit (name) (a);
put skip list ('id : ');
get edit (id) (a);
```

would cause the carriage return from "name" to be stripped off, then drop down to the next "get edit" and satisfy it without any other data being entered. The result was that "id" would be skipped. Worse yet, it was not a consistent error. The insertion of a "get skip" below the "get edit" cured the problem, but did nothing to enhance program readability. To the best of my knowledge, this has not been fixed. I will know better when I receive a distribution disk. This was the only bug I was able to find in version 1.3, even after a year of extensive programming in PL/I-80. Unfortunately, Digital does not publish reports of bugs found or fixed.

PL/I is well supported by DRI's support group. One of the members of the group is there to deal exclusively with PL/I problems. I found the staff available and ready to answer questions long before I ever started writing reviews. The staff has expanded in the last couple of years, and the phone(s) are answered 8 hours a day. On numerous occasions I have sent a wayward disk and request for help, and have always received an answer in a week or so. Perhaps it would be pertinent to mention here that DRI will allow software to be returned (if it was purchased directly from them) for only a restocking charge, if it is returned in a reasonable amount of time.

PL/I-80 fits comfortably on a single (3740) disk and will operate in any 64K CP/M or MP/M environment. When you think of the size of the language, this is rather amazing. I keep it and Access Manager on my A drive and still have all the room I need to accumulate programs and intermediate code on the same disk. Code generation is in relocatable 8080 assembly (8086 for the 86), and .REL files are normally generated for the linker. Microsoft M80 and L80 compatibility are maintained, and SID and MAC work well with generated PL/I code.

The only real shortcoming I have found yet with PL/I is that it is supported only by Digital in the micro version. This may be one reason the language does not get the serious consideration it deserves in the micro world. It is easily the most powerful programming language available today for micros.

Last year, *Info World* picked PL/I-80 as the software bargain of the year. These new and improved versions are even better than their predecessor, so I daresay they are the still the best software bargain. While PL/I is admittedly not a programming language for beginners, it is as good an applications language as is available today and should remain unchallenged until fuller sets of ADA are available for micros.

A fellow consultant and computer languages educator, Dr. William Hogan, has remarked (tongue-in-cheek) that the best time to buy a piece of software is when it is on the verge of obsolescence, because by then it is well documented and bug free. Far from obsolete, PL/I-80 is both comfortably old and virtually bug free, and these newest versions bring it even greater power. It is a job well done.

With its new double precision float feature, PL/I-80 1.4 exceeds the range and precision of Fortran 66.

The Tecmar Data Conversion Products

A/D and D/A converters with associated software

by Richard Newrock and Walter Knesel

Several data conversion products made by Tecmar are reviewed in this article: the TM-DA-100 digital-to-analog converter, the TM-AD-212 analog-to-digital converter and counter/timer, and the TM-G2-SSP software package.

In our laboratory, we perform a wide variety of measurements, from the very simple and slow (e.g., measurements of DC resistance) to the fairly complicated and fast (e.g., control of superconducting electronics and noise measurements). Therefore, two of the many important criteria we used to select our D/A and A/D converters were speed and flexibility. This led us to choose the three Tecmar products: they are available with a wide variety of options, many of which can be added later, and the top speed is reasonably fast (125 KHz.). In addition, a "complete" software package is available, compatible with the MicroAngelo graphics subsystem (Scion, Reston, VA 22091), which was already running on our system.

Getting the converters and the software up and running is not a simple task. The A/D card in particular is quite complex, with a maze of jumpers and switches. It also has more than its share of little quirks; we've been using it for some time and are still discovering new ones. In this review we discuss the boards and the software, delineating the important features and specifications and, insofar as possible, discussing the quirks and indicating possible problem areas.

Digital-to-analog converter

The TM-DA-100 is a high-speed, digital-to-analog S-100 board designed for real-time conversion; the board is sufficiently fast that most CPUs can operate at full speed. There are four independent 12-bit converters (DACs) that can be individually addressed and latched.

There is only one option available—a set of trimpots for adjusting the gain and range of the DACs. A companion board is available; it contains four 20 mA current drivers and four filters to smooth the analog output (the DA-100's outputs are unfiltered). We did not purchase this board.

A summary of our opinions: the board, although well made, doesn't have the fine details one expects from quality manufacturers. Electronically, it is well designed and well executed. It performs its functions properly, meets all of its specifications, and is very easy to use and to program.

The board is available from Tecmar, 23600 Mercantile Road, Cleveland, OH 44122 for $495.

The hardware. The circuit is constructed on a solder-masked glass-epoxy circuit board. It appears to be wave-soldered, the ICs are socketed, and the other components are carefully mounted. All the S-100 pins are present, even the unused ones; this can be very useful when one wishes to make on-board modifications or additions. We noted two problems. First, no attempt at silk-screening or other form of labeling is made. The manufacturer should at least consider labeling the address switches and the jumpers. Second, the external connection to the board is via a 16-pin DIP socket. While this is inexpensive, there is no reason why a standard header with connector ejectors can't be used. Labels and connectors might seem trivial, but this is an expensive and generally well-made product. The manufacturer should do it right.

The board power supply appears to be adequate, and a liberal number of by-pass capacitors are provided. Power is taken from both the positive and negative (nominal) 16V S-100 power lines. For the board to operate properly, these voltages must never drop below 17 volts. Be careful of excessive ripple when operating near the limits; it can drop the voltage below the limit momentarily, and cause an improper conversion.

No claims are made about compatibility with the IEEE/696 standard in the manual; however, such claims are made in the promotional literature. To be sure, we checked the pin assignments; no conflicts were found. The board uses none of the reserved lines (RFU), the undefined lines (NDEF), or the new ground lines. As mentioned, the supply voltages must remain above 17V; this is IEEE/696 acceptable, but it is not a requirement.

All of the important specifications of the DA-100 are displayed in Table 1; the more important ones are discussed below.

The specified conversion time is 3 microseconds but, if only the LSB is changed, it can be reduced to 1.5 microseconds. This is reasonably fast; as mentioned, the AD-100 will operate without delays with most CPUs.

The board is addressable as I/O ports or as memory locations (memory-mapped); the user selects the one he prefers with a DIP switch. Eight contiguous memory or I/O locations are used, two for each DAC (each DAC has two storage registers to hold the input word). If memory-mapped, the board sits at any eight consecutive locations at or above FE00. Only the lower 16 address lines are used; no provision has been made for extended addressing. If port addressing is used, the eight-port block can be located anywhere in the 256 port I/O space. No provision is made for the additional ports available in the new standard. The location of the board in the I/O or memory space is selected with DIP switches.

There are five output ranges for each DAC, three bipolar and two unipolar (see Table 1). The range is

Richard Newrock and Walter Knesel, Physics Dept., University of Cincinnati, Cincinnati OH 45221

selected by wire jumpers, a method we find to be poor. In our work we need to change ranges frequently. Constantly removing the computer from its rack to change jumpers is not only tiresome, but will eventually result in damage to the board. We would much prefer being able to change ranges with software or, at the very least, with DIP switches. The hardware necessary to implement software control of the range should not be very expensive, and the addition of it would be a great improvement.

The offset, drift, and slew rate specifications are excellent (Table 1) and, as far as we could determine, the board is stable and accurate. Two trimpots are needed for each DAC to adjust its range and gain. These are listed in the manual as options, although it is not clear who is to supply them. The fact that they are optional is not mentioned anywhere in Tecmar's advertising or in their promotional literature. We learned about them when we read the manual, after purchasing the board. This board is meant to be used for scientific research, where high precision is normally required and expected; trimmers are necessary and the board should come with them installed.

The input word is stored in two latches and is sent in two steps. First, the upper four bits of the new word are sent, and the DA-100 loads them into a four-bit register. Then the eight lower bits are sent. The DA-100 loads them, and the upper bits stored in the four-bit register, into a 12-bit register. The word then goes to the DAC and is converted to the analog voltage. The 12-bit register maintains the last word input, corresponding to the voltage currently being output.

Sending the most significant bits first, and storing them while the rest of the word is sent, ensures that no glitches will appear on the output during a word change. This method must also be used in 16-bit computers, as the DA-100 does not support 16-bit data transfers, and it is not possible to send all 12 bits at once.

The DA-100 has one major problem. When RESET is asserted, all latches are set to 0. (In one place in the manual, it is stated that RESET sets the latches to −1. This did not appear to be true.) If one of the bipolar ranges has been chosen, the output of the DAC will be zero and no particular problems result. However, if a unipolar range has been selected, the DAC's output will rise to one-half the full-scale value. This can be very bad, especially if the DA-100 is being used to control voltage-programmable instruments. The DAC's output can exceed the instrument's maximum programming voltage, resulting in damage to the instrument. With voltage-programmable power supplies, the output current can rise quickly to a point where external circuitry is damaged. We found no way to circumvent this; it was necessary for us to disconnect all instruments before we reset or restarted the computer.

The software. Writing programs for the DA-100 is quite simple; the manual shows that four lines of assembler code are all that is needed to convert a word to an output voltage. No complete (and simple) programs for this are provided in the manual or in the software package. Tecmar does provide several complicated programs in their software package, programs primarily designed to output arrays of data repetitively. For our purposes, controlling voltage-programmable instruments and chart recorders, we needed a very simple subroutine to accept an integer and send it to the DA-100, which would then output and hold a constant voltage. This was very simple to write and took less than 30 lines of code.

The manual. The manual is adequate. The technical details and explanations are nearly complete, although written in a "choppy" manner. Functional, schematic, and layout diagrams are provided, and diagrams and tables clearly indicate jumper layouts and switch settings. Some simple assembler code is shown for driving the board, and a few simple debugging instructions are given.

The manual needs a copy of the manufacturer's specifications for the DACs. It also needs to be reorganized in a more coherent way, and an index needs to be added. Although the manual is small and can be scanned rapidly, we still found the lack of an index irritating. There are a number of places in the manual where contradictory statements are made, and these ought to be corrected. Finally, in the diagram showing the DIP header for the output connector, the pin numbers are misplaced: the two rows of numbers need to be interchanged.

Analog-to-digital converter and counter/timer

Tecmar's AD-212 is a versatile analog-to-digital converter and counter/timer. It is a high-speed, high-accuracy device, intended for real-time applications. The board consists of a hybrid A/D converter module, an Advanced Micro Devices counter/timer, and the hardware necessary to connect the two, provide for addressing, interfacing, etc.

The basic AD-212 consists of two boards: The motherboard has all the timing circuits and control logic; the daughterboard has the data acquisition module (DAQ) and some minor control circuitry.

The A/D section of the AD-212 consists of a multiplexer (MUX), a differential input amplifier, a sample and hold circuit, and the A/D converter itself. It, of course, is used to convert analog data to digital for processing by the computer.

The counter/timer section contains a very versatile counter/timer and the circuitry necessary to address and buffer it. It is used to control and time A/D conversions, but can also be used for a variety of independent functions. These include time of day, event counting, frequency shift keying, coincidence alarms, complex pulse generation, programmable duty cycle waveforms, and more.

The AD-212 is available with a wide variety of options. These include:

- 12-, 14-, or 16-bit resolution

The offset, drift, and slew rate specifications are excellent and, as far as we could determine, the DA-100 board is stable and accurate. Writing programs for it is quite simple.

- 2.5, 10, 30, 40, 100 and 125 KHz conversion rates
- resistive-programmable gains, 1 to 1000 in seven steps
- software-programmable gains: 1, 2, 4, 8 or 1, 10, 100, 500
- ±250V isolation
- multiplexer expansion boards to 256 channels
- a cold junction signal panel for thermocouples
- an enclosure and cable for remote mounting of daughterboard

Not all of the options are compatible. Choosing one often excludes others, and there are some very real trade-offs that must be considered. One is conversion rate vs. resolution. The higher the resolution, the slower the maximum possible conversion rate. For instance, choosing 14-bit resolution precludes choosing conversion rates faster than 10 KHz. If 16-bit resolution is chosen, the fastest conversion rate is only 2.5 KHz. A similar trade-off must be considered if you wish to use the high-gain preamplifiers. The higher the gain of an amplifier, the longer the settling time and, therefore, the slower the sampling rate.

Note that there is no point in getting more speed than your processor can handle. For example, at the 125 KHz conversion rate, your processor must run at 8 MHz or better to take advantage of the maximum conversion speed. However, the people at Tecmar tell us that, using the various high-throughput options on the board, it is possible to run at 100 KHz with a 4 MHz processor.

The basic system comes with 16 single-ended or 8 differential inputs, allows 8 or 16-bit data transfers, has 12-bit resolution, a 30 KHz conversion rate, and a gain of 1.

A summary of our findings: Tecmar's AD-212 board is reasonably well made, but, as with the DA-100, it lacks the fine details one expects from quality manufacturers. It is mechanically well designed and executed. It does what it is supposed to do and does it well. It is, however, fairly difficult to learn to use it, and the manual increases the difficulty. Indeed, the manual needs very extensive rewriting.

We purchased and tested the basic board with two options: the 40 KHz conversion rate and the higher of the two programmable gains. It was purchased for $995. This includes the basic board for $765 and the two options for $230. Representative costs of other options: 16-bit, $1120; 125 KHz, $620; ±250V isolation, $350; enclosure for the daughter, $60.

The hardware. The circuit is constructed on a glass-epoxy solder-masked printed circuit board. There are no labels on the card. Given the plethora of jumpers and switches, labels are definitely needed. All of the ICs are socketed, and the other components are neatly mounted and waved-soldered. All of the S-100 pins are present.

There are headers for five ribbon cables on the two boards. One of these headers is for the analog inputs, a second is for the counter/timer I/O, the third is for multiplexer expansion, and the last two form a pin and socket interconnection between the mother and the daughterboards. The connectors for external cables are poor; they are simple pin headers rather than quality ribbon cable connectors. Furthermore, they are located at odd positions on the boards, rather than along the top edge, causing cables to be mounted down between the boards in the S-100 frame. This means that the entire A/D subsystem must be unplugged to change a cable.

The motherboard always plugs directly into the S-100 bus. The mother and daughterboards can be mounted together, with the daughterboard bolted to the back of the motherboard. The combination takes two slots. Electrical connection is made via the above-mentioned pin and socket connectors. These also accept standard ribbon connectors, allowing the boards to be separated and connected by up to 100 feet of ribbon cable.

Tecmar recommends that the daughterboard be mounted remotely if high gain or high accuracy (14 or 16 bit) is needed. An optional enclosure is available for that purpose. Remote mounting allows the analog signal-handling portion of the A/D converter to be located near the sources of low-level signals. This is very important when it is necessary to reduce noise and pick-up.

The mother and daughterboards have their own power supplies. As with the DA-100, these require at least ±17V on the (nominal) ±16V S-100 power lines; if the voltage levels fall below these values, improper conversions will result. A power connector is provided on the daughterboard, and it can be used to provide regulated +15V, 5V, and the analog and digital grounds, for a small amount of external circuitry. This could be useful when the daughterboard is in a remote location. In such cases, power is provided to the daughterboard itself via the ribbon cable. There is also a ±5V line available on the counter/timer ribbon cable for powering a small amount of external logic.

Complete compatibility with the IEEE/696 standard is claimed. We checked this, and, for the pin assignments at least, it is true. A few remarks are in order here. The AD-212 uses all of the new ground lines (20, 53, and 70). This can create problems in older systems, especially those with CPUs that use line 53 for SSWDSB*; if that line is grounded these CPUs will not run. Tecmar suggests cutting the necessary ground traces in that case. None of the NDEF or RFU lines are used. Both SIXTN and sXTRQ are implemented, as they must be since 8 and 16-bit data transfers are supported. All 23 address lines are decoded, so that if memory-mapping is used, extended memory locations can be used in an 8-bit system or the full address space in a 16-bit system. Again, the

The AD-212 is a high-speed, high-accuracy device intended for real-time applications. It is available with a wide variety of options.

STOK SOFTWARE, INC.

Humanizing the Computer (TM)

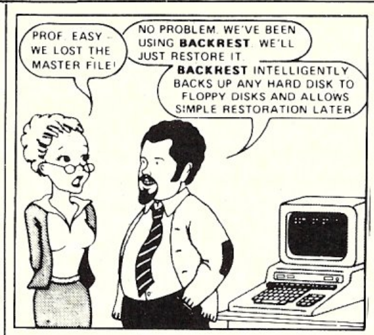
STOK PILOT™

Put your knowledge of your office environment into your computer so that your personnel will be properly guided in your absence.

STOK PILOT is a control language that allows easy development of a menu driven environment as well as an on line instructional utility for any CP/M or MP/M application. It can guide the user through an entire process without requiring the user to enter cumbersome system commands, hence making the system transparent to the user.

STOK PILOT can chain to any "COM" file program, or series of "COM" files, and regain control when the last program ends. This, and other unique features make it easy to design complete turnkey systems.

Disk and manual - $129.95. Manual alone - $14.95.



BackRest™

Hard Disk Backup, Restore and more!

- Incremental and Full backup.
- True copying of random files.
- Split large files if necessary.
- Migrate or delete selected files.
- Automatically restore bad files.
- Print Management reports.
- Requires CP/M 2.2, CP/M 3 or MP/M.

$99.95

VISA MasterCard

SSi

Stok Software Inc.
17 West 17th St.
New York, NY 10011
212 / 243-1444

Make CP/M Better
Global Command Processor

You can't destroy data if you change disks without control-C!
Make drive "A", User "O" public!
Read data files in user zero from any other user!
No system disk needed in drive "A"!
Redo and edit last command to reduce typing!
Auto installation under CP/M 2.2.
Just type GCP.
Send $29.95 and start smiling.

SuperDO & SuperSUB - $29.00

SuperDO allows the CP/M operator to type a string of commands that will execute one at a time. So you can work away for a while and let your computer do its thing. Example:

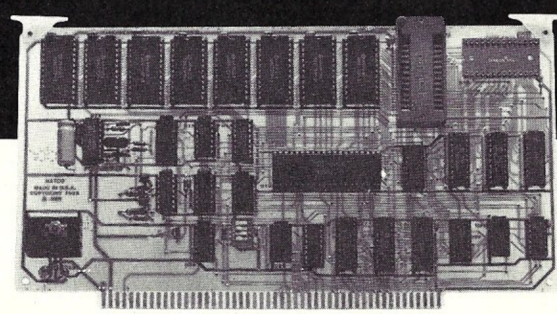
A > DO ASM PROG1; LOAD PROG1; ASM PROG2; LOAD PROG2;DIR

SuperSUB is an enhanced SUBMIT command that will run on any standard CP/M 2.2 system. It runs faster than SUBMIT because it buffers the commands in memory.

CP/M-MP/M are TM of Digital Research, Inc. Dealer Inquiries Invited.

CIRCLE 158 ON READER SERVICE CARD

EMPROM-1: Eprom Emulator/Programmer



INTRODUCING the EMPROM-1

The MATCO Eprom Emulator/Programmer EMPROM-1 is a sophisticated tool designed to support the software engineer who routinely develops Eprom-based applications. It will emulate and/or program any of the currently available five-volt only MOS Eproms from the 2758 at the low end to the 27128 at the high end. The primary features are:

- Compatible with IEEE 696 S-100 standard
- I/O mapped—consumes no system RAM space
- 16K bytes of on-board emulation RAM
- Zero insertion force programming socket
- Complete with software: ■ runs under CP/M*, MP/M*, or CDOS** ■ menu driven program supports both programming and emulation ■ limited debugger type functions are provided ■ supplied in source code on 8" single sided soft-sectored disk.
- Incorporates the Intel fast programming Algorithm
- NO personality modules necessary
- Hardware provisions have been incorporated to allow programming of future 32K Eproms

The board may be purchased with a zif socket mounted on the board, or with a flat cable socket to connect to an external module.

Board with zif socket	**$305.00**
Board with flat cable socket	**295.00**
External module + 4′ cable	**55.00**

64K STATIC RAM Board is only $395!

- Compatible with Proposed IEEE 696 standard
- Total RAM/EPROM interchange capability (R/W jumpered)
- Extended Address capability (A16-A23)
- Global Addressing possible in 16K groups
- Global/Extended allocations are 16K per group
- 0-8 wait states for EPROM with fast CPU's
- Wait states selectable on/off by 16K group
- Responds to PHANTOM* on pin 67 (jumper selected)
- MWRT generated on-board, or taken from bus
- Automatic deselect of empty sockets via FF detector

All boards are assembled, tested, burnt in and supplied with 150 nsec CMOS RAMS plus a 1 year limited warranty and owner's manual.

MASTERCARD VISA

MATCO

Manufacturing and Test Company, Inc.
427 Perrymont, San Jose, CA 95125 (408) 998-1655

*CP/M and MP/M are trademarks of Digital Research Inc. **CDOS is a trademark of Cromemco, Inc.

CIRCLE 46 ON READER SERVICE CARD

power lines must remain above ±17 volts; this is IEEE compatible, although not a requirement.

The detailed specifications are shown in Table 2; we will discuss only the important ones.

The A/D converter. The AD212 is designed to accept nine different A/D modules from two manufacturers. As these are not pin-for-pin compatible, an extensive jumper area is needed on the daughterboard to allow for the different variations. These jumpers are set at the factory. A variety of user-selectable options is available as well; these are selected in an additional jumper area.

All of the possible A/D modules used by Tecmar are able to convert signals from 0 to 10 volts and bipolar signals from −10 to +10 volts. Some of them have two additional ranges, 0 and 5 volts and −5 to +5 volts. The module you receive depends on which of the word size and throughput options you choose, as listed in the brochure. There is also a module, available as an option, which allows for true binary rather than decimal conversion (i.e., 0 to 10.24V, etc.). We have a decimal module, with two bipolar and two unipolar ranges, and feel it to be more than adequate for most needs.

There are several input amplifier options that can be purchased. First, a resistor-programmable amplifier is available. The user must change a resistor on the board whenever he wishes to vary the gain. Such an amplifier is only useful in situations where gain changes are rarely made and where one wants the same gain for every channel to be sampled.

The second option is a software-programmable amplifier. There are two ranges of gain available (Table 2). For us, this is the preferred option. It allows one to change gains quite simply, and to change the gain channel by channel, as one steps through a data-gathering sequence. The only reasons we can imagine for choosing the resistor-programmed amplifier is that there is more gain available (a maximum gain of 1000 as compared to 500), more gain selec-

Table 1. DA-100 specifications

General:	
Bus	IEEE-696
Resolution	12 bits
Channels	4
Output:	
Ranges	0 to +5V, 0 to +10V, ±2.5V, ±5V, ±10V
Current loop	4 to 20 mA (optional)
Impedance	0.05 ohms typ.
Connector	16-pin DIP header
Accuracy (0 to 70° C):	
Linearity	±¼ LSB typ. ±½ max.
Diff. linearity	±½ LSB typ. ±¾ LSB max.
Gain error	±0.01% typ. ±0.03% max.
Offset error	±0.05% FSR typ. ±0.15% FSR max.
Drift (0 to 70° C):	
Total bipolar	20 ppm FSR/°C max.
Total error—unipolar	±0.08% FSR typ. ±0.15% FSR max.
Total error—bipolar	±0.06% FSR typ. ±0.10% FSR max.
Gain	±15 ppm/°C typ. ±30 ppm/°C max.
Unipolar offset	±1 ppm FSR/°C typ. ±3 ppm FSR/°C max.
Bipolar offset	±5 ppm FSR/°C typ. ±10 ppm FSR/°C max.
Speed:	
Settling time to ±.01% FSR	3μs FSR 10V 5μs FSR 20V 1.5 μs 1 LSB change
Slew rate:	10V/μs min. 15V/μs typ.

tions in the range and the fact that such amplifiers usually have slightly better gain stability than programmable ones.

It should also be remembered that there is a penalty to be paid for using high gain: the higher the gain, the longer the settling time of the input amplifiers.This, quite naturally, results in longer conversion times, slowing down the sampling rate. If great amplification and high-speed sampling are both necessary, it would be better to use external amplifiers with fast slew rates and short settling times.

The input impedance, over 100 megohms, is quite

Table 2. AD-212 specifications and options

Specifications	
General:	
Bus	IEEE-696
Resolution	12 bit
Channels	16 single-ended, 8 differential
Inputs:	
Range (FSR)	±5V, ±10V, 0 to 5V, 0 to 10V
Max. input	10.24V
Current	1 nA at 25°C typ. 40 nA at 70°C
Input impedance	greater then 100 megohms
Source impedance	less than 1 kilohm
Input capacitance	
"off" channel	less than 10 pF
"on" channel	less than 100 pF
Input fault current	limited to less than 20 mA
Gain	1
Connector	40-pin ribbon cable
Accuracy:	
Relative	±0.025% at 30 KHz throughput
Absolute	same
Inherent quantizing error	±½ LSB
3-sigma noise	+0.01% FSR referred to input
Monotonicity	guaranteed 0 to 70°C
Stability:	
Linearity tempco	less than 3 ppm FSR/°C
Gain tempco	less than 15 ppm FSR/°C
Offset tempco	less than 10 ppm FSR/°C
Power supply sensitivity	0.003% FSR/%voltage change
Signal dynamics:	
Maximum throughput	30K chan./second, single-ended (5 μs delay for settling; 28 μs for A/D)
Sample-hold aperture uncertainty	5 ns
Crosstalk	80 dB at 1 KHz, "off" to "on"
Diff. Amp. CMRR	greater than 70 dB, DC to 1 KHz
Sample-hold feedthrough	80 dB down at 1 KHz
Max. error FS transition between channels	1 LSB
Options	
Resolution	14 and 16 bits
Conversion rates (Note: 2.5 KHz for 16 bits; 10 KHz for 14 bits)	2.5, 10, 40, 100 and 125 KHz
Gain (resistor programmable)	1 to 1000
Gain (software programmable)	1, 2, 4, 8 and 1, 10, 100, 500
Range	±5.12V, ±10.24V, 0 to 5.12V, 0 to 10.24V
Enclosures	Screw terminal and signal conditioning for thermocouples Remote cabinet and cable for daughterboard Multiplexer expansion boards to 256 channels

respectable. This allows the board to be connected directly to nearly any signal source, with no loading problems, at least at low frequencies. The input capacitance is reasonably low, about 100 pF for an "on" channel. This is fine in most cases. A possible problem could arise with the 100 KHz and 125 KHz options. At 125 KHz, 100 pF is an impedance of 13 kilohms, which might be low enough to load down a high-impedance source at high frequencies, or (perhaps) to introduce some roll-off. The latter is not a major problem, as it can usually be corrected in the data analysis program, but one should be aware of it when designing a system.

The input current is limited to a maximum of 20 mA, even if the MUX input fails. This provides a nice margin of safety for protecting one's experimental apparatus.

The accuracy and stability specifications are quite good (Table 2), and the board appears to meet the specifications. There is very little power-line voltage sensitivity, which is a necessary feature for long-term data taking. Tecmar recommends that the unit be recalibrated every six months or so, and with that we concur.

The maximum throughput depends very much on how the board is set up and how the programming is accomplished. With a 12-bit converter, Tecmar claims that 30,000 channels/second, single-ended should be possible. We did not check this, but, given the specifications of the A/D module, we feel it is reasonable. To obtain this rate, Tecmar uses a 5 microsecond delay (for MUX, input amplifier and sample/hold circuit settling) and 28 microseconds for the actually A/D conversion. To take advantage of this speed, Tecmar provides for "pipelining" the data, which improves throughput and data transfer rates to the point of approaching that of TMA. This is discussed further below.

Finally, the crosstalk and the differential amplifier CMRR are reasonably good for a board such as this and should suffice for all but the most demanding applications.

Addressing and data transfer. The AD-212 requires 16 address locations for the 16 internal registers associated with the A/D converter and timer sections. (It actually has eight 16-bit registers, but in many cases the upper and lower 8 bits contain the same information.) The addresses are contiguous and are determined by five DIP switches on the motherboard. These switches allow the board to be placed anywhere in memory or I/O space. The board can be configured for an 8-bit or 16-bit I/O space or for 16- or 24-bit memory space.

Tecmar addresses a number of possible problems that may arise in choosing between memory-mapped or I/O operation. In particular, older S-100 CPU boards may generate ill-timed status and bus timing signals. False selection of the AD-212 can occur if these older processors are used. Tecmar, recognizing the possible occurence of many of these problems, has provided several switch-selectable address and timing options to take care of them. Otherwise, the selection of memory-mapped or I/O operation, and the setting of the address switches, is straightforward.

Because SIXTN and sXTRQ are implemented, the AD-212 can recognize whether an 8- or 16-bit CPU is requesting data. If a 16-bit transfer is requested, the entire output word can be transferred in one data-transfer cycle. For an 8-bit CPU, two data-transfer cycles are needed, and Tecmar provides several software examples for doing this. A useful feature is that if an 8-bit and a 16-bit CPU are co-resident, they can both request data from the AD-212.

Multiplexer. There is only one A/D converter, and it can digitize only one signal at a time. The internal multiplexer (MUX) allows the user to connect more than one input to the A/D. Jumpers are used to configure the MUX inputs in three different ways: single-ended, differential, and pseudo-differential. The MUX actually consists of two 8-channel multiplexers, and the jumpers couple the two into the desired input configuration.

For single-ended operation, the two multiplexers are cascaded to form one 16-channel multiplexer. For true differential operation, the two work in parallel and switch pairs of inputs into the high and low sides of the input amplifier. In pseudo-differential operation, the multiplexers are cascaded for 16 inputs, as in single-ended operation, but the common line is allowed to float. This causes any common mode noise appearing on the signal return lines and on the analog common line to be rejected.

The MUX address lines are available at a ribbon connector on the daughter. A cable from this connector can be daisy-chained to (optional) multiplexer expansion modules and the channel capacity increased to a maximum of 256.

The channel selection techniques are important, as they determine the ultimate throughput. There are two methods used to select a channel. The first, and simplest, is to load the address of the channel to be read into the MUX address register before initiating the conversion. This is best for randomly sampling several channels; however, there is significant software overhead if several channels are to be read in sequence.

The second addressing method takes advantage of autosequencing capabilities built into the module. In this mode, after the sample and hold circuit locks onto the signal, the MUX is automatically stepped to the next channel. This allows the user to read repetitively a series of consecutive channels, and read them faster than can be done by software sequencing.

The A/D modules, as received from the factory, will step to and read all 16 channels, cycling through them repetitively. Tecmar has added some additional hardware to allow autosequencing between any two signal channels. The number of the last channel de-

Older S-100 CPU boards may generate ill-timed status and bus signals, and false selection of the AD-212 may then occur. Tecmar has provided switch-selectable address and timing options to take care of such problems.

GENESIS COMPUTER CORPORATION

introduces...

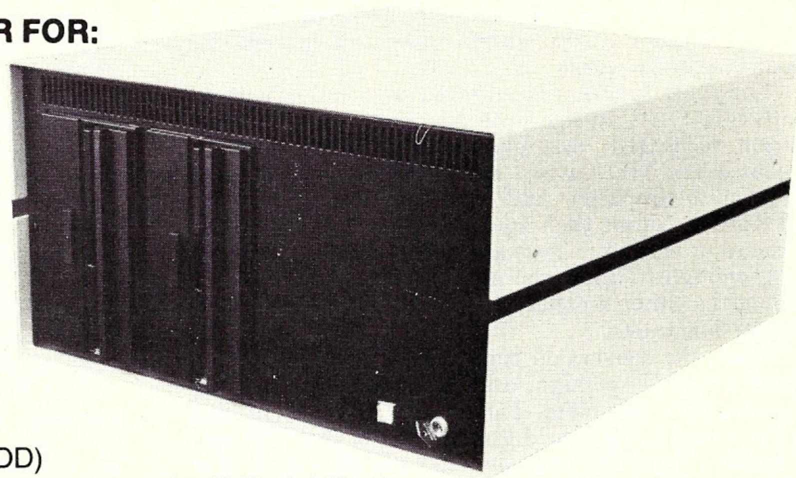
the Price & Performance Leader

the GENESIS 8D S-100 micro

A COMPLETE DUAL 8" DISK DRIVE S-100 MICROCOMPUTER FOR ONLY... **$2395.00**

THE IDEAL MICROCOMPUTER FOR:

★ Business
★ Education
★ Industry
★ Development
★ Networking



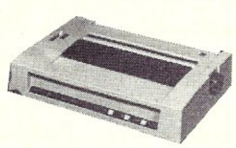
FEATURES:

★ Z80A CPU 4MHz
★ 64K Dynamic RAM
★ 2 1.024 MB 8" disk drives (DSDD)
★ Floppy Disk Controller
★ CPM 2.2© Operating System (MPM© and TURBODOS® Available)
★ 10 Slot IEEE 696 Mainframe
★ 2 Serial Ports
★ 2 Parallel Ports
★ DMA
★ Keylock ON/OFF Switch

COMPLETE THE SYSTEM WITH OUR INEXPENSIVE TERMINAL AND PRINTER COMBINATION FOR ONLY

$925

ADDS VIEWPOINT



C. ITOH PROWRITER

GENESIS COMPUTER CORPORATION

1444 LINDEN ST.
P.O. BOX 1143
BETHLEHEM, PA 18018
(215) 861-0850

SCHOOLS AND BUSINESSES ASK ABOUT OUR LEASE PURCHASE ARRANGEMENTS

© Registered Trademark Digital Research

® Registered Trademark Software 2000

CIRCLE 173 ON READER SERVICE CARD

sired is loaded into a register, and the number of the first channel is loaded into a second register, and into the MUX. As the MUX steps through the channels, the current channel number is checked against the stored last channel number. After the last channel is read, the starting channel number is reloaded into the MUX from the storage register, and the sequence repeats. This is all done in hardware without the intervention of the CPU.

It is worth noting that the easiest and quickest way to obtain the maximum conversion rate is to allow the A/D module to free-run. It can be set up to strobe itself at its maximum rate, the reciprocal of its conversion time plus delay time. Tecmar indicates in the manual (but not in the promotional literature) that two of the available A/D modules do not have this free-run feature. Again, they fail to say how one makes certain of getting (or not getting) one of those.

There are several clever features built into the system. In order to understand them, however, it is necessary to learn a few simple things about how the board actually works.

The A/D function is initiated by a pulse Tecmar calls STROBE. In the most straightforward configuration, STROBE first changes the MUX address. Because the MUX has a finite settling time, as does the input amplifier and the sample-hold circuit, STROBE cannot then start the conversion directly. Instead, it is used to start a timer whose period is just long enough to allow the input circuitry to stabilize. When the timer finishes, it triggers the A/D and the conversion starts.

When the conversion begins, the end of conversion (EOC) flag is set high, and it remains so until the conversion is complete. The change in the EOC flag is then used to load the digital output of the A/D into a buffer. The CPU can now read the data from that buffer while the A/D begins another conversion. This is the "pipelining" mentioned above.

In "pipelining" data it is necessary to ensure the integrity of the data. For this purpose there are two additional flags: DONE and OVERRUN. DONE, set by EOC, signals the CPU that data is available in the buffer. When the CPU reads the buffer containing the most significant data byte, DONE is cleared. OVERRUN is kept clear as long as DONE is cleared before new data is loaded into the buffers. Programs can run as fast as possible and, by checking the status of OVERRUN, one can be certain that the data is good and that none was lost. Note that for 8-bit data transfers, it is necessary to read the lower byte first or the safeguard will be defeated.

The DONE and OVERRUN flags are available at the interrupt header and are part of the status register. Tecmar remarks that if they are to be used, it is better to use them as interrupts, as it takes much too much time to check them after every conversion.

There are four ways to generate the STROBE signal; each has its special use. The first, and simplest, is to write anything to register four. This will produce a STROBE and initiate the conversion. Clearly, this is most useful when the A/D conversions are under direct software control.

The second method is to input a STROBE pulse through a gated input. This input, available on the timer/counter header connector, can be enabled and disabled under software control. It allows the user to trigger a conversion from an external source.

Thirdly, one can generate STROBE pulses at regular, timed intervals without involving the CPU. To do this, one takes advantage of one of the many capabilities of the counter-timer, and has it supply strobe pulses. The timer output pin can be jumpered to the external input pin on the counter/timer header. There is, however, an objection to placing a jumper on those pins, as discussed below. This configuration is the one expected by the software package.

The fourth and last method is to use an ungated strobe input, available on the multiplexer expansion header. Conversion is initiated by any negative transition on this line. This is a most useful method, as that line can be daisy-chained to several AD-212s and can be used to trigger them simultaneously.

The counter/timer. The counter/timer module is based on an Advanced Micro Devices Am9513 system timing controller (STC), an extremely versatile chip. It is available to the CPU as a general-purpose counter/timer besides being available for A/D conversions. The Am9513 has a large variety of operating modes that allow it to perform all of the functions listed in the introduction to this section. These modes can be reconfigured under software control.

The STC is based on five general-purpose, independent, high-speed, 16-bit up/down, BCD/binary counters. The accumulated count can be read without disturbing the count, and the counters can be cascaded to form up to an 80-bit counter. Each of the five counters has three external I/O lines associated with it. There is a gated input to control when counting may proceed; a source input, for the signal to be counted; and a three-state output (it can output a pulse, a square wave, or a more complex waveform). The output circuitry of the counter detects when the counter register is zero and sends the output signal chosen. For two of the counters there is an optional alarm, the output being sent whenever the state of the counter matches that of the alarm register.

The 15 I/O lines from the five counters are buffered and brought out via a 40-pin header on the motherboard. There are two other useful lines brought out as well. The first is FOUT, the output of a four-bit counter that can be programmed to divide its input by any integer from one to 16. The second is the external strobe, the line used to trigger the A/D from an external source.

The placement of this latter line is somewhat of a blunder. As mentioned, in order to use the TM-G2-

Some clever features are built into the system. . . but to understand them one must learn a few simple things about how the board works.

SSP software package, it is necessary to use the output of one of the counters to trigger the A/D. Therefore, one has to jumper that counter's output pin (OUT5) to the external strobe pin. This is done on the counter/timer header, and when this jumper is in place, a ribbon cable cannot be connected. This, of course, precludes using any of the "excess" timers of the Am9513 unless one provides a jumper at the end of the attached cable. It's a rather silly design flaw.

The Am9513 has inputs for external crystals or frequency sources. Tecmar has connected these to a 1 MHz signal derived from the bus 2 MHz clock. This provides the reference frequency for the time-of-day and elapsed-time functions.

The internal data bus of the Am9513 is 16 bits wide, as are the internal data registers. The Am9513 status, control, data, etc., registers are accessible as ports and can be used by user programs. Tecmar provides the manufacturer's literature describing the chip and its use, as well as several example programs written in Basic and assembler.

The timer chip is the slowest part of the board and (possibly) cannot operate at processor speed with very fast processors. Tecmar provides an optional wait state when the AD9513 is directly addressed; the wait state lasts one clock cycle.

We must add that the Am9513 chip is one of the most versatile counter/timers we have seen, and Tecmar makes its full power available to the user.

The interrupts. The last of the hardware we wish to discuss are the interrupts. Tecmar designed the board to use either vectored or polled interrupts. Any one of four flags can be used to interrupt the CPU: OVERRUN, DONE, and two timer interrupts. These interrupt sources can be enabled or disabled with software commands: one simply sets or clears certain bits in the command register.

For the vectored interrupts, any combination of the four sources and the eight interrupt vectors is permitted. These are set by soldering jumpers on the interrupt header.

If your system does not have the hardware for vectored interrupts, the interrupt flags must be connected to the INT line and polled interrupts employed. The CPU must, after it is interrupted, read the state of the four flags in the status register. This is considerably slower than using vectored interrupts.

The software. Once the board is set up, programming it is reasonably straightforward and is basically a matter of proper I/O to the registers. There are sixteen 8-bit registers, used in pairs, or eight 16-bit registers. When used for status, commands, etc., the lower and upper registers in each pair contain the same data and perform the same function. Two of the register pairs are the timer command and data ports. One of them is the command and status port for the A/D converter. Two are used to clear the timers used by the A/D section: One clears the OVERRUN flag, and one is the data output for the A/D.

Tecmar lists several sample programs in the manual, in Basic and assembler. (The same programs are offered in each language.) These assume an 8-bit system, I/O ported, with no wait states or interrupts—i.e., they are designed to run on nearly any 8-bit computer. We found them very useful for testing the board and for troubleshooting. They perform simple functions and are more valuable as examples than as usable data-taking programs. Using them, one can set and read the time of day, use the alarms and count events, take data from a single channel or from several channels, and use the A/D converter with a timer-generated STROBE.

The manual. The AD-212 is very complex, with numerous jumpers and switches; it can perform many different functions. Installing it your system requires considerable direction. Unfortunately, the manual fails to provide such direction. It has a wealth of detail about the maximum throughput, accuracy, speed, etc., and great quantities of technical information are given; yet, for installation, it leaves you on your own.

The lack of simple installation instructions is not the only problem with the manual. Much information about the switches and their functions is given, but there is little information about when to use them. The board has a number of unique options but, again, it is never made clear when to use them. The jumper summaries are quite terse and important information is often left out. There are numerous diagrams and drawings, but they aren't especially useful. In particular, the connector layout diagrams are very poorly done and difficult to follow.

The writing is not coherent. Information appears at random, as it occurred to the writer(s), with no thought for continuity or for the end user. Pertinent information about a particular function, switch, or jumper is scattered about the text. Often, in the middle of complex technical explanations, comments and remarks appear about setting certain jumpers or switches. Perhaps this is useful; you find out the reason for setting a switch as you're informed you have to set it. However, rooting about in all that verbiage, trying to discover how to set all the switches, is extremely time-consuming. It is essential that setup data and technical explanations be separated. Infinitely compounding these problems is the lack of an index.

We found using this manual an extremely frustrating experience. The user needs an ordered, logical account of the options, jumpers, and switches and how to set them. He needs to know when to use them. A major fault of this manual is that it is of no help whatever in sifting through the options. A user also needs a section on first-time installations. It wouldn't be a bad idea in this case to follow Godbout's example and include a section entitled "How to get this board up and running in 5 minutes without reading the manual."

The Am9513 chip is one of the most versatile counter/timers we have seen, and Tecmar makes its full power available to the user.

The manual needs to be redone completely. We wonder if hardware manufacturers realize that a good manual can do as much for sales as good hardware? Certainly some of them realize this, e.g., Godbout; others should learn from them.

There are also a number of minor problems. Our version of the manual doesn't tell you which are the positive and which are the negative input channels. There is no information about adjusting the zero on the converter module; the manufacturer's literature for the modules, which might have that data, is not provided.

The software examples are in Basic and assembler, but Tecmar doesn't tell you whose Basic they are using; it is not Microsoft's. The driver routines in their own software package are all in Fortran, but there are no example programs in Fortran.

The manual sets up a convention that solid lines are factory-installed jumpers, whereas dotted lines are user-installed jumpers. The manual is printed so poorly, however, that we often could not tell the difference.

The analog-to-digital software package

Tecmar provides software for analog-to-digital and digital-to-analog operations, including graphics, in the TM-G2-SSP software package. It consists of two parts, ADDA11 and ADLIB17, written in Microsoft Fortran.

ADDA11 is a general-purpose A/D and D/A operating system. It can be thought of as a software interface and operating system for the AD-212, the DA-100, and the MicroAngelo graphics subsystem. In ADDA11, Tecmar attempted to create a very flexible software package, designed for ease of use and wide applicability. Both source and machine code were purchased.

ADLIB17 is a library of Fortran relocatable subroutines. It contains all the subroutines called by ADDA11, as well as several other useful programs. It is intended to be used as a library for user-designed software.

Our opinion of the package is easily summarized. We found ADDA11 easy to use, but, beyond the initial testing and learning stages, not very useful for our purposes. We found the individual subroutines in ADLIB17 much more useful, and feel that most experienced programmers will write their own software and use the library.

The package is available on an 8" standard disc. The object file costs $75, the source code, $195.

ADDA11. ADDA11.COM controls the A/D, the D/A, and the MicroAngelo boards and allows the user to specify a variety of analog operations and experimental conditions. It logs onto the system with a set of default parameters that the user can alter before beginning data acquisition.

ADDA11 is complicated; there is a large number of parameters to go with many analog, digital, video, and other functions. The parameters, which determine speed, channel number, gain, scaling, etc., are varied by entering a parameter code followed by the new value. Functions are called by entering a function code, without parameters.

The best way to describe the software is to list the possible operations and functions. Five different analog-to-digital (and vice versa) operations can be performed: one can collect and store analog data with or without simultaneous video display. One can take a set of previously collected data and either display it on a video monitor, or convert it to (repeating) analog signals for display on an oscilloscope. Finally, one can also set up and draw *x-y* plots on the MicroAngelo monitor.

We found these functions to work as intended. The first two are, of course, the most useful and necessary, as is being able to display previously obtained data. This latter is obviously important in high-speed data taking where you cannot simultaneously graph the data. We did not find imitating oscilloscopes to be very useful; perhaps without video graphics they would be.

These functions take full advantage of the multichannel capability of the hardware. The user can take data from, and send it to, several channels; he can specify the channel numbers, or the beginning and ending numbers of a sequence of channels. He can specify the number of data points to be taken and the delay time between the points. (The software will not take data faster than 40 KHz, limiting the time interval a minimum of 0.05 milliseconds.) The user can set the gain of the input amplifier (if that option is installed), and he can specify that data collection only start in response to an external trigger.

On the MicroAngelo graphics subsystem, axes, ticks, scales, and titles can drawn along with the data. Special video, scale, mode and offset parameters can be specified to give you a range of screen positions, lines, symbols, etc.

Several useful non-data-taking functions are supported. These include setting and reading the real-time clock, setting and waiting for a 24-hour alarm, creating and reading graph set-up parameters from disk files, and reading and writing data to and from disk files.

ADDA11 does a lot and is very flexible. However, after using it for some time and in a number of ways, we decided it was trying to do too much and lacked several important features (discussed below). It appears to be written for the nonprogrammer, someone who wants to load it in, enter a few parameters, and perform a variety of analog functions. This comes at a cost in memory space and program overhead. Ultimately, we found it more efficient, and in some sense easier, to call the subroutines ourselves from ADLIB17. In addition, we were then able to add what we felt was missing.

The main strength of the ADDA11 program is that it allows you to begin to take data immediately, concentrating on the connections to, and operation of, the boards.

7 Good reasons why you should subscribe to *Creative Computing.*

It's the Number One magazine of computer applications and software!

There's one place you can always be sure of learning more about microcomputer software and applications: *Creative Computing.*

Every month *Creative Computing* provides you with a continuing education on everything related to microcomputers and computer equipment. Useful articles, "how to" tutorials, exciting new applications, games and "no holds barred" reviews of the latest software and equipment make up a major part of *Creative Computing*'s editorial content.

We give you probing features on programming breakthroughs and important news. Plus in-depth articles on elementary, intermediate and advanced software and applications topics—to help you develop your knowledge and skills, save hundreds (perhaps thousands) of dollars in unneeded software, discover uses for your personal computer that you might never have considered. Articles that increase your overall "computer consciousness." Here's how:

1 *Creative Computing* gives you things to actually do with a computer.

Just owning a computer isn't enough. You've got to know what to do with it. That's why applications are our primary focus. Text editing, animation, graphics, business simulations, data base and file systems, music synthesis, control of household devices, communications, games—some of the applications and software you'll learn about in *Creative Computing.*

2 *Creative Computing* discusses business applications in simple, nontechnical language.

If you're a business person who needs to know about the latest developments in word processing and office applications, turn to *Creative Computing.* We clarify such business applications as investment analysis, futures evaluations, data base management, mailing list programs, text editing, word processing and simulations. And all the software available for business people.

3 *Creative Computing* helps you decide which computer equipment is best for you.

Our tough, no-nonsense equipment profiles arm you with the facts before you walk into a computer store. You'll know the right questions to ask and how to cut through the jargon and sales hype. We give you authoritative guidance in deciding what you need, what you don't need—and what's right for you and your pocketbook.

4 *Creative Computing* covers computer education in depth.

We started out as a computer education publication, and we're still committed to the educational community. We regularly carry articles on designing educational software, evaluating educational software, teaching concepts and terminology in computer education, text editing applications for literature and computer simulations in the classroom—plus a great deal more.

5 *Creative Computing* brings you hours of mind-expanding game entertainment.

We've got a soft spot for the computer game addict—and computer game software. We know you want to understand more about the new computer games flooding the market: which ones are easiest to learn? Require the most skill? Offer the most surprises? Give you the best graphics? Provide the most challenge? Contain a new twist? *Creative Computing* brings you the answers.

6 *Creative Computing* features the state of the art.

Columns on the most popular personal computers, a "software legal forum," letters to the editor. Reviews of books, games, organizations, dealers and events. Fascinating interviews with leading innovators, equipment designers, program developers and game inventors—men and women who'll give you a real glimpse of the future!

7 Our price is right.

By subscribing to *Creative Computing* now, you can save as much as 33% off the full subscription price. To learn elsewhere what you'll learn from *Creative Computing*, you might spend hundreds of dollars in course fees and books. Then you'd have to winnow out what you could use from all that you'd learned. But *Creative Computing* does that for you, so you'll have time to enjoy your own computing interests. And that saving of time makes this offer very inexpensive indeed.

Join over 150,000 *Creative Computing* readers by subscribing today! Just use the coupon at right.

SAVE UP TO 33%!

Creative Computing · P.O. Box 5214 · Boulder, Colorado 80322

YES! Send me *Creative Computing* for:

☐ One year (12 issues) for $19.97—I save 20%!
☐ Two years (24 issues) for $36.97—I save 26%!
☐ Three years (36 issues) for $49.97—I save 33%!

Savings based on full one-year subscription price of $24.97.

Check one: ☐ Payment enclosed. ☐ Bill me later. 8H435

Mr./Mrs./Ms.________________________
(please print full name)

Address____________________ Apt._____

City________________________

State____________ Zip____________

Offer valid in U.S. and possessions only. Please allow 30 to 60 days for delivery of first issue.

The program's main strength is that it allows you to begin to take data immediately, concentrating on the connections to, and the operation of, the boards. You can test the boards, take some data, etc., without worrying about programming details. This program, if it can be run as is, is very helpful in learning the hardware and software systems and the functions and parameters.

We were not able to run it as is. ADDA11, as compiled and linked by Tecmar, expects the AD-212 to be ported at 10H, the DA-100 at 20H and the Microangelo at its standard F0H. In our computer, an Intersystems DPS-1, the I/O card occupies port 10H, among others. In order to use ADDA11, we readdressed the boards, reassembled the I/O drivers with the new port equates, and relinked ADDA11.

There were several problems we encountered. First, Tecmar does not tell you which routines need changing; you must check the source code for each one. Second, we had problems with the linker. Apparently, our copy of L80 V3.44 would not link Fortran modules with M80 macros to produce a working machine-language file. We didn't know this, of course, and wasted many days in fruitless troubleshooting of Tecmar's software. Ultimately, after reassigning the Intersystems board and putting the AD-212 back to 10H, we were able to run ADDA11, as supplied by Tecmar. Although someone at Tecmar was aware that such a problem existed, apparently customer service wasn't informed and, over the course of several months, we never learned about it.

ADDA11 is easy to use. It signs on and asks for the time and date to set the real-time clock. After the clock is set, all of the parameters and their default values are displayed; this covers nearly all of an 80 x 24 screen. The first time you see this, it is quite intimidating, but it does not take long to become familiar with all the parameters and to start to take data. The manual has good descriptions of the commands; couple them with the material in the other manuals, and you will be able to operate the system easily. It is advisable to read all of the manuals cover-to-cover before starting.

As mentioned, we found ADDA11 useful in learning the capabilities of the data acquisition subsystem but felt it was not really suitable for our use as a general-purpose laboratory data acquisition program. Several drawbacks lead us to this conclusion.

First, there is no simple D/A routine. ADDA11 can take an array of data and convert the entire array into analog signals, but there is no provision for converting a single data point. This is needed, for example, for voltage-programmable instruments. One could, we suppose, create an array of one, but this is cumbersome and requires unnecessary overhead. Our solution was to write our own D/A driver and incorporate it into the ADLIB17 library.

Second, we found the graphics software minimal. It appears that you can only graph the data taken on one channel versus that taken on another, with only one data curve per graph. This is very limiting; in many experiments it is necessary to plot a number of variables versus time. Obviously, one can create a file of time values, but this shouldn't be necessary. Tecmar has informed us that plots versus time can be done, but it is not obvious how.

In addition, the types of graphs are limited. ADDA11 lets you plot the data in ascending cartesian coordinates. We often need descending coordinates, as well as polar and logarithmic ones. We quickly abandoned the Tecmar graphics routines, preferring instead to use those available in Graphpak; they have considerably more flexibility.

It ought to be noted that ADDA11 writes unformatted data files for efficiency in disk space and access time. Many users are not familiar with this; the details can be found in Fortran manuals. Tecmar has provided two useful Fortran programs: One reads the contents of an open data file, the other reads it and rewrites it as an ASCII file.

ADLIB17. ADLIB17 contains all of the routines linked into ADDA11 as well as several others. Included are the subroutines for D/A and A/D conversion, timing and alarms, video graphics, and the disk file handlers. These functions can be called by programs written in Microsoft's Fortran-80 or in any other compiled language that uses similar formats. We found these routines to be very useful, much more so that ADDA11. We have several comments to make and several hardware snags to discuss.

In general, we are pleased with the subroutines. Except for the lack of a simple D/A routine and the limited graphics, they are flexible and enable you to do what needs to be done. They are somewhat cumbersome to use, as there are many different variables to be initialized and large common blocks to employ. However, the manual is clear about what needs to be done for each subroutine, and no real problems arise. The software works.

The subroutines make extensive use of common blocks for parameter passing. Many of the common blocks are extremely large, which creates a number of problems. First, it is very cumbersome to write custom programs; the common blocks must be properly included and be of the appropriate length. Tecmar notes that for programming ease, a file called COMMON.BLK is provided. It is meant to be edited into user routines to save time. Unfortunately, this file was not provided on our disk.

Second, although the use of common blocks is machine efficient, we feel that they seriously compromise the independent use of the subroutines and cause extra overhead in writing user routines. We wonder if the gain in machine efficiency is worth it.

A major problem we encountered was the lack of documentation about the hardware configuration required by the software. Many man-hours were spent trying to sort things out. At a very late stage in the

The software is very good, although it is not as powerful as the hardware and does not use the hardware's full capabilities. A stronger software support system could and should be created.

$49.95

CP/M™ will never be the same again!

CLIP™, our Command Line Interpretive Program, makes CP/M more efficient than ever before. It replaces CP/M's command processor with a user friendly "shell" containing over 50 commands. CLIP runs as a normal CP/M program using only 5K of memory when running programs and 25K when interpreting commands.

Editing Keys

CLIP adds editing keys to all standard CP/M programs—just like a word processor! Single keystrokes allow you to move forward and backward to insert or delete text. The command interpreter remembers your last 10 commands, which can be retrieved, edited, and reentered effortlessly.

Make Your Own Commands

Sequences of commands may be placed in a file to create a "macro" command.

Macro commands can:

- Take arguments and switches
- Execute programs
- Ask questions
- Conditionally execute commands
- Loop
- READ and WRITE files
- Call upon other macros

Built-in Text Editor

CLIP contains a simple editor for creating macros, programs, or data files. English-like commands simplify any editing task.

Automatic File Searching

Keep commonly accessed files on one disk and user number. CLIP searches for files when not found on your current disk and user number—saves disk and directory space.

Built-in Calculator

The functions +, −, ×, ÷, Modulus, AND, OR, XOR, and NOT are available with ten memory registers. Results may be passed to programs and macro commands.

A Unix™-like Environment with I/O Redirection and Pipes

CLIP can direct screen output from a program or macro to a file for viewing or printing, or cause program input to be taken from a file instead of the keyboard. Pipes force the output from one program into the input of another automatically.

Sophisticated command language

- Conditional command execution
- Command file tracing
- Multiple commands per line
- File I/O from the command line
- Extensive string manipulation
- Descriptive error messages

CLIP is equivalent to a high level programming language, yet simple enough for non-programmers, too!

Software Tools Package Just $25.00 with CLIP!

We have many software tools that compliment and enhance CLIP.

BC	- Binary file comparison
BFE	- Binary file editor
DMP	- Hex file dumper with ASCII
HC	- Horizontal file concatenate
LCASE	- Convert to lower case
PAD	- Pad lines with spaces
SC	- Source file comparison
SR	- Search for a pattern in a file
SRT	- Simple in memory sorter
UCASE	- Convert to upper case
UNIQ	- Extract unique entries from a file
WC	- Count words, lines and bytes
XC	- Extract columns from a file
XW	- Extract words from a file

Data Encryption Program $25.00 with CLIP

A twelve character password encrypts a file such that no one may read it but you.

CLIP; a Help compiler; over 60K of help information; example macros; user's manual and binder—all for $49.95.

Software Tools $25.00.
File Encryptor $25.00.
User's manual $25.00 (credit upon purchase of CLIP).

Visa, Mastercard, check or money order accepted. Please add $3.50 for shipping. Arizona residents add 6% tax.

CLIP, the tools and the file encryption program have been designed for the Z80™, running CP/M 2.2. Distribution is on 8 inch SS/SD or Osborne 1, Kaypro, Apple, Heathkit, Northstar, or Attache . Call for availability of other formats.

Dealer inquiries invited.

Thoughtware, Inc.

800-821-6010/602-327-4305

P.O. Box 41436, Tucson, AZ 85717

process, we discovered, buried at the very end of the manual, an appendix that contains some of this information. It is, however, poorly labeled, incomplete, and there is no reference to it anywhere in the text.

This lack of information about the hardware configuration is important. There are several switches on the AD-212 that are crucial to the operation of the software. The first of these is the "wait for conversion" switch. We were not aware that we had to set this until we carefully read the listing of one of the A/D subroutines.

The second is the switch for fast CPUs. We use a 4 MHz Z80; apparently it is a fast CPU. We didn't think so, and our programs failed to run.

The third is the interrupt-enable switch. We suppose we could have anticipated the problems we had with that one, since we weren't using interrupts at the time. In any case, enabling it caused great problems.

Let us again mention one of the very necessary jumpers, the one connecting the trigger input of the A/D converter to the output line of timer five. This jumper is used when the timer is to initiate a conversion; the need for it is discussed in the appendix but not in the text. The software uses the timer to initiate conversions; we did not realize that and assumed that the conversion was initiated directly by the software. Our failure to use this jumper resulted in our returning the board for service unnecessarily. At least we learned that Tecmar's turnaround on repairs is fast (about two days).

In sum, we feel that the software is very good, although it is not as powerful as the hardware, and does not use the full capabilities of the hardware. A stronger software support system could and should be created, with more independent and therefore intrinsically more useful modules.

The manual. Compared to the manual for the AD-212, this one is good.

An overview of the software and a reading of Tecmar's advertising tells you that Tecmar's intent is to have the user employ ADDA11 on a regular basis for data collection. Thus, one would think that ADDA11 forms the focal point of the manual. Curiously, however, the bulk of the manual provides details about ADLIB17.

The discussion of ADDA11 is clear and readable. The various functions and parameters are listed and defined and their default values given. Familiarity with the AD-212 and the DA-100 is necessary of course, but given that, the manual will get you started.

Except in setting up the hardware. As stated, nowhere in the text are the switch and jumper settings necessary to use the software given. The board addresses are not explicitly stated either.

The discussion of ADLIB17 is excellent. The reader is first reminded of the various ways of transferring parameters to subroutines in Fortran, and of the use of the common blocks. The common blocks are then described; this is a large and very useful section, well written and easy to follow.

Tecmar continued . . .

The most important part of the manual is the descriptions of the various subroutines. These are extremely well done, very useful, and easy to use. Each subroutine is described individually, and the descriptions are well separated and easy to locate; each one begins on a new page, in a logical grouping. Furthermore, each description follows the same pattern; you always know where to look for a particular piece of information.

The reader is first given the necessary set-up information. He is told which common blocks are required, and which parameters must be passed in the calling statement. The variables which must be defined prior to calling the subroutine are listed, as are the variables modified by the subroutine's execution. Next, there is a sample program illustrating how a call to the subroutine is made. The sample routines are useful in themselves for many purposes. If there are any special features to be noted, or remarks to be made, they are listed after the sample routines.

In closing we must offer an observation. We had reason to call Tecmar many times with various problems. Invariably, they were of little help. The people who answer their telephones know little about the product and the customer service people weren't helpful either. Their usual conclusion was "We better have our software guy call you." Fifty percent of the time he did: two to four four days later. Tecmar produces excellent products but, unless they are willing to offer adequate technical support (and manuals!), the quality of their product is negated.

PMMI's MM-VT1

Interface a Touch-Tone transmitter/receiver and voice synthesizer to your CPU and telephone line

by Mark Zeiger

Have you ever wanted to control an appliance in your home when you would be away? Did you ever want to leave a message for someone who might call? In this computer age, you might think these questions trivial, since it isn't hard to establish a telecommunications system or set up a bulletin board. But wait—to receive your message, doesn't the calling party need a computer also? And can you expect any person who might be calling you to carry one around? With PMMI's latest product, the answer to both these questions can be "No."

The MM-VT1 is an IEEE-696/S-100 board from PMMI (Potomac Micro Magic, Inc.) that contains a Touch-Tone transmitter/receiver and a speech synthesizer. You will probably recognize the company's name because of the S-100 modem board they sell. The MM-VT1, their latest product, can do the following:

- Dial a number (Touch-Tone or pulse)
- Determine if the number is busy or if the phone has been answered
- Answer the house phone when it rings
- Decode touch tones entered from the caller's end
- Control external devices via a parallel port
- Speak

Suppose you are running a business where your salesmen must call in for instructions. With the MM-VT1, you can give each salesman a code to enter (on a Touch-Tone phone) after he has called your computer, and the MM-VT1 can issue (speak) an appropriate message depending on the code. There is probably no limit to how interactive this process can be (i.e., there may be replies and subsequent messages based on the replies).

One point may need clarifying: the MM-VT1 is not a modem. You cannot use it to communicate with a timesharing system unless that system will accept and decode touch tones.

Is programming such a system difficult? Yes and no. Getting the computer to respond to a series of touch tones is not very hard, but the programming the board to speak a message can be unbelievably tedious. Fortunately, PMMI has included some very sophisticated software to make programming of the Votrax SC-01A (a 22-pin integrated circuit that talks) a great deal easier.

The following components and accessories are included in the MM-VT1 package:

- S-100 board
- FCC-approved protective coupler and line cord
- Manual
- Phoneme dictionary
- 8" CP/M disk (optional at additional cost)

The manual is nicely printed and bound, and fairly well organized. It is short (only 24 pages) and somewhat technical, but it does explain all the features of the board. Software examples are given in Basic and 8080 assembly language, and the examples in the manual are also provided on disk along with other software not discussed in the manual. If you have not used hardware of this type before, you might find the examples difficult to understand. Since I am familiar with PMMI's S-100 modem board, I had no trouble understanding the operation of the dialer and speech synthesizer, but the uninitiated might have to reread the manual several times before it becomes clear.

The phoneme dictionary contains phoneme constructions of approximately 1,000 words. Phonemes are the 60 or so basic sounds of human speech, and the Votrax chip is capable of reproducing these sounds. To get the board to speak you must translate words into the constituent phonemes that make up the sound of the word. The dictionary is written by Votrax, Inc., of Troy, Michigan—and unless you're a speech therapist, the dictionary is essential if you want the board to reproduce human sounds. Indeed, one wonders if a 1,000-word dictionary is large enough. The software itself gives unlimited vocabulary; the dictionary is thus a convenient supplement to the software.

Physically the board is nicely laid out. It contains about 40 ICs that are soldered to the board (no sockets), one six-position DIP switch, a few potentiometers, and two header sockets. One of the sockets is for the protective coupler and the other for an auxiliary connector used to power up the computer when the phone rings (although you must add some circuitry yourself to do this). There is also a 26-pin header on the top of the board for the parallel ports. The potentiometers are located horizontally near the top of the board so that they may easily be adjusted without removing the board from the computer. The board also has those little ejectors that make removal from a tight S-100 slot easy. Every S-100 manufacturer should be required by law to either have them or provide a lifetime supply of bandaids!

On the back of the board (where there should be only traces) were three wires and a resistor that were soldered on. This board might have been a prototype; but if not, it's really no big deal.

There are three voltage regulators on board (+5, +12, and −12), so the board does not draw much power (400 ma at 5V). PMMI claims it meets the IEEE 696/S-100 specifications, and naturally it is FCC approved for connection to the phone network.

Mark Zeiger, 198-01B 67th Street, Apt. 3B, Flushing, NY 11365

Using the board

Setting the board up is the essence of simplicity. Select the base port address via the DIP switch (if you don't want to use the standard address 0C4H) and then put the board in a slot in the computer. Plug the ribbon cable from the coupler into the socket on the board (the correct socket is marked) and then attach the coupler to the phone line. The line cord supplied contains modular connectors at each end. One goes into the coupler and the other to the modular jack attached to your phone line. PMMI takes great pains to point out that you must notify the phone company if you connect this device to the telephone system.

How well does it work? It dialed flawlessly, answered the phone when it rang, and decoded the touch tones perfectly. The software for this is fairly straightforward, as you will see in the examples later. It may be programmed to wait for a dial tone before starting to dial, and with a little bit more code, it may detect when the phone on the other end is ringing after the call has been placed. It can also detect when the number dialed is busy, but this requires changing one of the potentiometers.

However, if this is done, you may no longer be able to detect the dial tone. In other words, the board can detect dial tone and ringing, or busy signal and ringing, but it can not detect both dial tones and busy signals unless you stand over the board with a screwdriver in the pot and constantly adjust it. It is also possible to change the frequency to which the board will respond by changing a resistor. This must be done by unsoldering the resistor on the board and replacing it with one of another value. The manual gives values of resistors that should be used.

Operating the Votrax chip is not difficult, but programming it is time consuming to the utmost. You must figure out the phonemes that make up each word (hence the phoneme dictionary), and then output those phonemes by sending 6 bits (64 different values) to the appropriate register. Four levels of inflection may also be programmed with each phoneme by using the remaining high-order two bits in the aforementioned register.

Picking the correct inflection and stress is probably the most difficult aspect of programming the Votrax speech synthesizer. However, PMMI has included some extensive software, at additional cost, which will take ASCII text and write code that may be translated into the phoneme code. This code may be placed into programs that you write, thus making Votrax programming somewhat simpler.

Unfortunately, as to the quality of speech, I cannot give PMMI high marks. The speech quality is a function of the Votrax chip used on the PMMI board. The speech is difficult to understand and takes a lot of getting used to. Nobody is every going to think that they're talking to a human being. Pitch can be adjusted by a potentiometer between what the manual says is a low male voice and a high female one, but the best I could get was something ranging from a medium pitch to what sounded like a fight between Donald Duck's nephews. I tried coding some words myself instead of using PMMI's software, but the results were the same. I had some friends call up and listen to what the unit was trying to say, and they also had a difficult time understanding the speech. The speech can be made to sound better by changing the inflection, but there is a great deal of trial and error involved in getting the correct stress and pitch.

In the following sections I will discuss each of the capabilities of the MM-VT1 and give some coding examples to show that some of the functions of the board are very easy to implement.

Programming the MM-VT1

The MM-VT1 is I/O mapped and uses four consecutive port addresses with a base port that must be a multiple of four. It is shipped from the factory set for a base port of C4 (Hex) and therefore uses port addresses C4, C5, C6, and C7. These ports are chosen by PMMI because they are directly above the ports used by the PMMI modem board (C0H-C3H). However the ports are very easily changed by a DIP switch (although this will also necessitate changing all the supplied software). It's not really a problem, since source code that can be assembled using the CP/M assembler is provided for all the programs.

The registers used are as follows:

OUTPUT

- *Register 0 (C4)*—Enables Touch-Tone decoder and Votrax, decodes touch-tones to be sent, and selects timer pulses for dialing and timing.
- *Register 1 (C5)*—Interrupt masks, phone line control, and power control
- *Register 2 (C6)*—Votrax speech control
- *Register 3 (C7)*—Parallel port output

INPUT

- *Register 0*—Touch-Tone decoding, Votrax status, timer pulses, dial tone and ring detect.
- *Register 1*—Phone line control and timing.
- *Register 2*—Not used
- *Register 3*—Parallel port input

Control of the board is accomplished mostly by polling the ports and checking appropriate bits in the registers. The board may also be set to use interrupts; hence jumper pins are provided to allow the board to use the S-100 interrupt line or the eight vectored interrupts. Interrupts are enabled through software.

Dialing using touch tones

The MM-VT1 is capable of sending 12 DTMF (Dual Tone Multi-Frequency) frequencies. These are the standard telephone frequencies 0-9, *, and # found on Touch-Tone phones, as well as the nonstandard frequencies A, B, C, and D. To send a tone, just output the numbers 0 to 16 (binary) to the lower four bits (bits 0-3) of output register 0. The tone will be sent when bit 4 of output register 0 is low, and turned off when bit 4 is high. Therefore, the duration of the tone can be controlled by software and, if you desire, the hardware timing pulses on the board may be used to make CPU-independent timing loops.

See Listing 1 for an example of how to dial a num-

The inflections of words and sentences can be edited to make the speech somewhat more human.

ber using DTMF frequencies. Naturally, a few things have been left out. For instance, you might want to put in an instruction to wait for a second dial tone when calling from an office that requires dialing a number for an outside line (usually a "9"). You might also want to code the routine to ignore all characters that are not valid DTMF characters.

Touch-Tone decoding

As mentioned earlier, the MM-VT1 is capable of answering the phone when it rings and determining whether tones are being entered on the caller's end—a necessary step when people call in and enter a code using DTMF. Depending on the code, the program can branch to a given routine. That routine might be to operate some equipment interfaced with the TTL output port on the MM-VT1, or to issue a spoken message using the Votrax chip.

Some thought must be given to the situation in which a caller hangs up prematurely. The PMMI board provides a means of detecting a new dial tone (the normal result of a caller hanging up), and also a means of detecting the "line current" produced by ESS (Electronic Switching Service) exchanges. Polling these bits in the status register will allow the board to disconnect after a remote hang-up.

The routines in Listing 2 may be used to answer the phone and decode DTMF frequencies. The equates r0-r3 are as in Listing 1. The caller will enter a series of tones terminating with the "#" tone. A number will be created from the tones, and the program will go to an appropriate routine if the code entered is valid. A "*" will cancel input and require the caller to enter his code again.

Speech synthesis

Getting the Votrax to speak requires sending a bit pattern to output register 2. The lower six bits determine the particular phoneme, and the two most significant bits determine the pitch or inflection. Hence there are four levels of inflection. If bits 6 and 7 are 00, the lowest pitch will be sent; if the bits are 11, the highest pitch will be sent. After sending the phoneme code to the output port, bit 5 of input register 0 must be polled. When it goes high, it indicates that the Votrax buffer is empty, and another phoneme may be sent. The speech synthesizer will keep issuing the same phoneme until another phoneme is sent to the port, or the command is given to stop. This may be done by making bit 5 of output register 0 low, or by issuing the phoneme code 3Fh (the "stop" code).

The manual lists 64 phoneme codes along with the sound issued by each. However, in programming the MM-VT1 for speech, you will find the dictionary provided by Votrax to be much more helpful. For example, if you want to have the speech synthesizer say "Good Morning," you would look in the dictionary to find the phoneme constructions for "good" and "morning." They are *G, OO1, OO1, D* and *M, O2, O2, R, N, I1, I3, NG* respectively. These symbols are not computer symbols. They are symbols established by the powers that be in the speech world. Now you would have to look in the PMMI manual to find the code for each of the above phonemes. The codes in hexadecimal are *1C, 16, 16, 1E* for "good" and *0C, 34, 34, 2B, 0D, 0B, 09, 14* for "morning." These codes do not take into account the use of pitch control by the two most significant bits.

Listing 3 contains a routine that will make the MM-VT1 speak a word or group of words. The HL register points to the phoneme codes.

Interrupts

All of the above software has been implemented by polling the appropriate registers. The MM-VT1 also has a powerful interrupt scheme similar to that found in the PMMI modem. Interrupts may be generated by any or all of the following conditions:

1. The DTMF decoder has detected a valid tone
2. The Votrax buffer is empty and ready to receive another phoneme
3. Timer pulse
4. Dial-tone or ring detect
5. Current on the phone line

Probaby the most useful of the interrupts is the fourth. Using the ring-detect interrupt, the computer does not have to be dedicated to waiting for the phone to ring. It can perform other useful work and still be able to answer the phone. Be aware, however, that interrupt-driven software can be very difficult to implement.

Software provided by PMMI

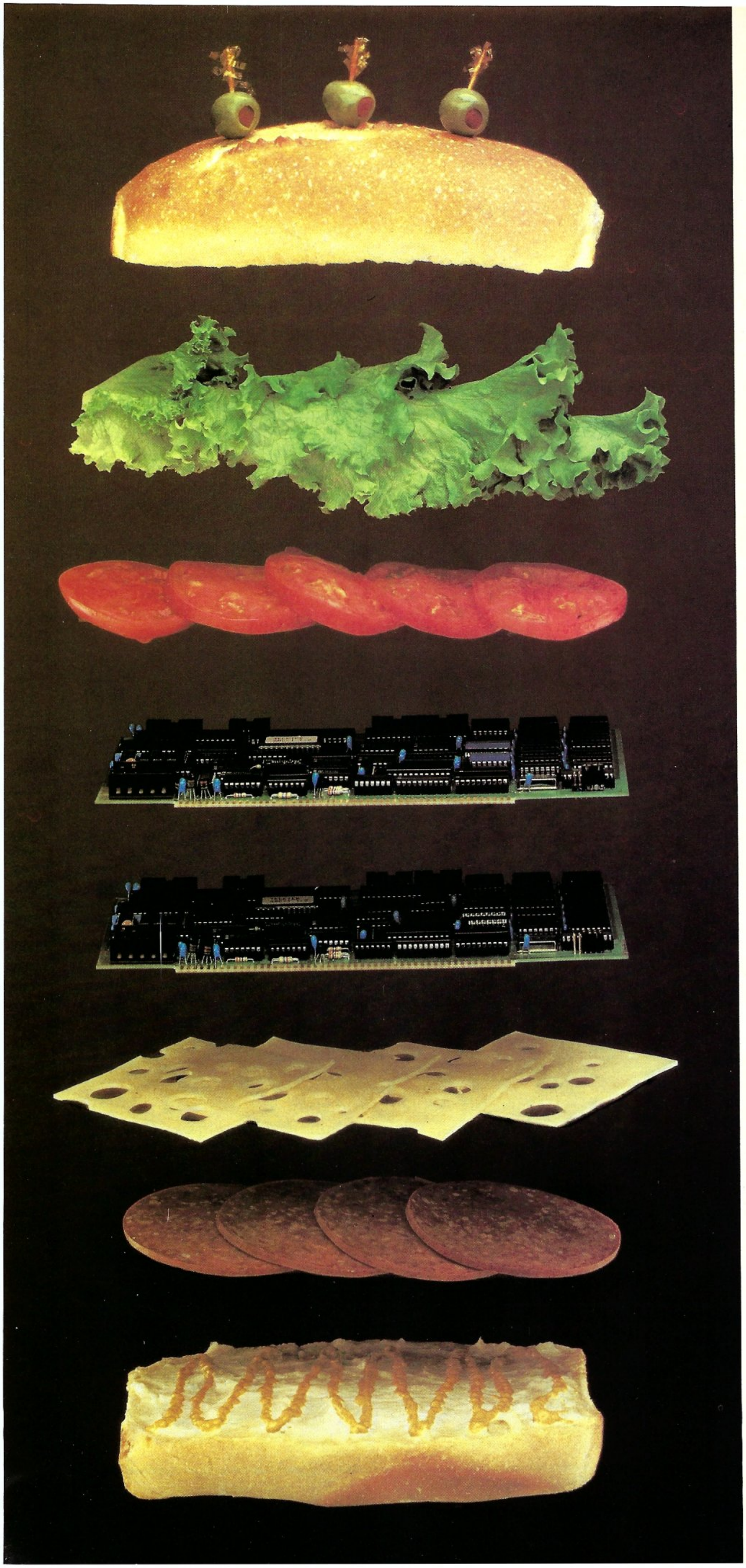
PMMI provides a number of short programs that are listed in their manual and supplied on an 8″ CP/M disk. The disk contains a few demonstration programs written in Microsoft Basic and 8080 assembly language. These routines are really provided as examples to get somebody started in writing code for this board.

Two additional programs are not mentioned in the manual These are complex programs that operate the speech synthesizer, create files containing phonemes, and control the phone line.

The main program is called VTTALK. It was written by PMMI and Peter B. Maggs and is a masterpiece. Provided both in source code and as an object file for people who don't know assembly language, it makes coding the Votrax speech synthesizer much easier. Interactively, you can perform the following steps:

1. Enter a sentence via the keyboard and have the Votrax chip speak the sentence.
2. Have the program speak the words in an ASCII file.
3. Have the program write the binary phoneme values of the words in an ASCII file or words entered from the keyboard.
4. Have the program write the Hex values of the words in an ASCII file or words entered from the keyboard. The Hex values may be sent to the console, or more important, be placed in a file.

The last routine is what really makes programming the Votrax much easier. With this program, you may



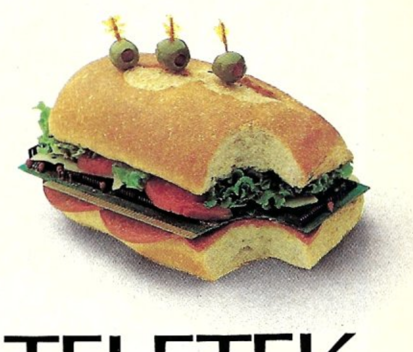
Teletek's New Combo Could Make You A Hero!

The SBC-II could be just the right ingredient for your latest concoction. The newest member of Teletek's family of multi-user, multi-processing S-100 products, the SBC-II essentially combines, or "sandwiches" two Teletek SBC-I's into one board. The SBC-II provides the capability to support two users from one standard size IEEE-696/S-100 slave board.

The SBC-II maintains full performance for each user with an independent CPU (Z80A or Z80B), 64K RAM, Serial I/O, and FIFO communications port to the system master. The system integrator benefits by getting complete support for two users for the price of one board.

TurboDOS and MDZ operating systems will support combinations of SBC-I's and SBC-II's offering system design efficiency and flexibility never before possible.

If you're hungry for value and efficiency, order an SBC-II from Teletek. You'll love every byte.



TELETEK

4600 Pell Drive
Sacramento, CA 95838
(916) 361-1777
Telex 4991834 TELETEK
Dealer inquiries invited.

CIRCLE 174 ON READER SERVICE CARD

create a file using any text editor and then have the words in that file translated to the Hex codes that will make the Votrax speak those words. The Hex values may then be incorporated into your software and used in a way similar to what I did in the above program. The only difference was that I had to do the phoneme translation myself.

It is also possible to edit the inflections of the words and sentences. This helps to make the speech somewhat more human. Naturally, it is impossible to have a program that translates any word into correct phoneme speech. The program will mess up occasionally, and this is where the editing features come in handy.

The second program, TRAN (written entirely by Peter B. Maggs), is actually a subset of the first program. This may be incorporated into programs you write so that some of the functions of VTTALK may be done without actually using the interactive parts of VTTALK.

Service, reliability and price

Since I used the board for only a short period of time, I cannot comment on its long-term reliability. However, I can comment on the reliability of PMMI. I've used their S-100 modem for three years and have been delighted with it. I had only one thing go wrong with it, and it was fixed within a week. I also have a friend whose PMMI modem was put out of commission twice by lightning. PMMI fixed it and replaced the coupler for a mere $30. Try finding prices like that from any other manufacturer! The turnaround time on the repairs was just one week. I consider this service extraordinary, and I highly recommend the company.

The board with the speech synthesizer is $385. It may also be purchased without the Votrax and parallel port for $260 if you just want the DTMF encoding and decoding.

The CP/M disk, which includes VTTALK, is an extra $30. The program is certainly worth the money, but I think that, with a board price of $385, it should be included free.

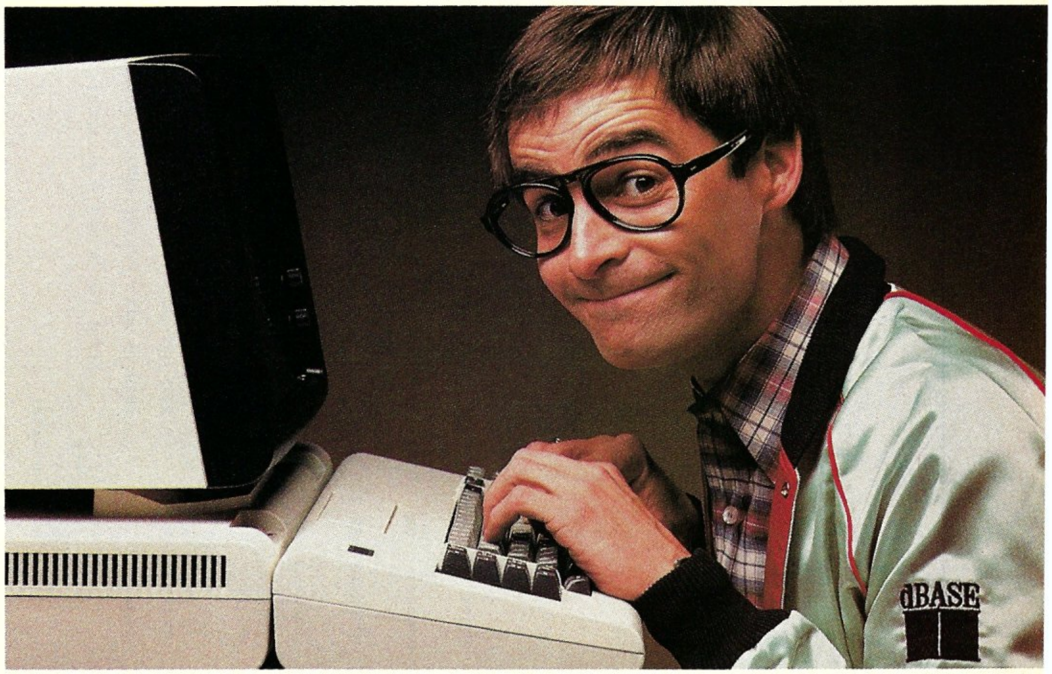
For more information, contact:

PMMI Communications
5201 Leesburg Pike, Suite 604
Falls Church, VA 22041
(703) 379-9660

Note: Touch-Tone is a registered trademark of The Bell Telephone System.

Mark Zeiger teaches mathematics and computer sciences in a Long Island high school. He has been a computer hobbyist for several years, and has done a great deal of work on assembly language in the CP/M operating system. Several of his programs may be found in the CP/M user's group library. He is also active in the Long Island Computer Association, having served as its president for a year.

All you dBASE II™ hotshots are about to get what you deserve.



You've written all those slick dBASE II programs.

Business and personal programs. Scientific and educational applications. Packages for just about every conceivable information handling need.

And everybody who sees them loves them because they're so powerful, friendly and easy to use.

But that's just not good enough.

Uh-uh.

Because now you can get the gold and the glory that you really deserve.

Here's how.

We've just released our dBASE II RunTime™ application development module.

And it can turn you into an instant software publisher.

The RunTime module condenses and encodes your source files, protecting your special insights and techniques, so you can sell your code without giving the show away.

RunTime also protects your margins and improves your price position in the marketplace. If your client has dBASE II, all he needs is your encoded application. If not, all you need to install your application is the much less expensive RunTime module.

We'll tell the world.

With your license for the dBASE II RunTime module, we provide labels that identify your program as a dBASE II application, and you get the benefit of all the dBASE II marketing efforts.

We'll also provide additional "how to" information to get you off and running as a software publisher sooner.

And we'll make your products part of our Marketing Referral Service. Besides putting you on our referral hotline, we'll publish your program descriptions and contact information in *dBASE II Applied,* a directory now in computer stores world-wide.

Go for it.

But we can't do any of this until we hear from you.

For details, write RunTime Applications Development, Ashton-Tate, 10150 West Jefferson Boulevard, Culver City, CA 90230.

Or better yet, just call (213) 204-5570. And get what you deserve today.

dBASE™ II

ASHTON·TATE

Listing 1

```
base      equ   Øc4h
rØ        equ   base
r1        equ   base+1
r2        equ   base+2
r3        equ   base+3

start:    call  dtwait          ;wait for a dial tone
          lxi   h,number        ;address in memory of string to dial
diallp:   mov   a,m             ;get digit to be dialed (in ASCII)
          ora   a               ;if binary Ø then end of number
          jz    done
          call  translt         ;translate number into output code
          out   rØ              ;send tone
          mvi   b,7             ;length of duration (7Ø msec)
          call  duration        ;wait a specified amount of time
          mvi   a,1Øh
          out   rØ              ;bit 4 high turns off tone
          inx   h               ;get next digit
          jmp   diallp          ;repeat until end of number

done:      .                    ;come here when done
           .
           .
number:   db    '2127256856',Ø      ;MICROSYSTEMS' number


;SUBROUTINES FOR DIALING
;
;
;tone:    Ø  1  2  3  4  5  6  7  8  9   A    B    C    D    *   #
;
;output   7  Ø  4  8  1  5  9  2  6  ØAH  ØCH  ØDH  ØEH  ØFH  3   ØBH
;to rØ
;
;For example, if the digit 4 is to be dialed, binary 1 must be output
;to register Ø. Note that for any value, bit 5 will be low which
;means that the tone will be enabled.

translt:              ;returns with proper output code in A-reg

          cpi   '#'        ;test for # and * as special cases
          jz    ret#
          cpi   '*'
          jz    ret$star

          cpi   'A'        ;see if A, B, C, or D tones
          jc    anumb      ;if carry, then it's a number
          adi   9          ;"A" -> 1Ø, "B" -> 11, etc. in low nibble
anumb:    ani   Øfh        ;strip ASCII.
          mov   e,a        ;put offset in DE-reg
          mvi   d,Ø
          push  h
          lxi   h,decodetbl  ;beginning of translate table
          dad   d          ;HL points to translated number
          mov   a,m        ;return with code in A-reg
          pop   h
          ret

ret#:     mvi   a,Øbh
          ret
ret$star: mvi   a,3
          ret

decodetbl:     db   7,Ø,4,8,1,5,9,2,6,Øah,Øch,Ødh,Øeh,Øfh
```

```
duration: in   rl              ;check the 100 Hz oscillator..
          ani  80h             ;..which is bit 7 of port 1
          jz   duration        ;wait until it goes high
dl:       in   rl              ;and now wait until it..
          ani  80h             ;..goes low so that 10 milli-..
          jnz  dl              ;..seconds have passed
          dcr  b               ;do another 10 msec until
          jnz  duration        ;..B-reg is 0
          ret


dtwait:   out  rl,40h          ;bit 6 set high takes phone off hook
          mvi  b,2
          call duration        ;a slight pause to let line settle
dtlp:     in   rl
          ani  04h             ;when bit 4 is low, dial..
          jnz  dtlp            ;..tone is present
          ret

;The translation for decoding DTMF tones received is not the
;same as that for encoding DTMF tones for dialing
;
;The translation code is:
;
;DTMF:    0   1  2  3  4  5  6  7  8  9  A    B    C    D  *    #
;
;Input:   0ah 1  2  3  4  5  6  7  8  9  0dh  0eh  0fh  0  0bh  0ch
;at r0
;
```

Listing 2

```
start:    xra  a               ;zero A-reg
          out  rl              ;turn off all interrupts - phone on hook
rngwt:    in   r0                                       (hung-up)
          ani  80h             ;bit 7 low means phone is ringing
          jnz  rngwt           ;keep looping until phone rings
rngwt2:   in   r0              ;now wait for first ring to stop
          ani  80h             ;bit 7 high means ring stopped
          jz   rngwt2

          mvi  a,40h
          out  rl              ;take phone off hook (answer it)

billdly:  in   rl              ;the phone co. requires a 2 sec billing..
          ani  20h             ;..delay before enabling DTMF. This bit..
          jz   billdly         ;.. high means delay is over

          mvi  a,80h           ;turn on touch tone decoder
          out  r0

restart:  lxi  h,0             ;zero number caller will enter..
          shld code            ;..by touch tones

toneinwt: in   r0              ;see if a DTMF tone has been..
          ani  10h             ;..received. High if tone received
          jz   toneinwt        ;wait for a tone
          ani  0fh             ;the lower 4 bits of r0 contain..
                               ;..the code received
```

PMMI continued . . .

Listing 2 *(Continued)*

```
          cpi   Øch         ;this is code for "#"
          jz    rcvcode     ;caller has completed his entry
          cpi   Øbh         ;this is code for "*"
          jz    restart     ;begin building code again
          jnc   toneinwt    ;ignore "A", "B", or "C" and get
                             next
          cpi   Ø           ;see if "D" entered
          jz    toneinwt    ;ignore and get next tone
          cpi   Øah         ;see if "Ø" entered
          jnz   gdnum       ;if not, then 1 - 9, so continue
          xra   a           ;make zero code a binary zero
gdnum:    call  buildcd     ;build a code
          jmp   toneinwt    ;get next tone

rcvcode:        .           ;we have received the code
                .           ;now go to appropriate routine
                .           ;based on code received

;SUBROUTINE

buildcd:                ;build a code out of numbers. It can't
                         be..
                        ;..over Øffffh
     ;mult previous value by ten and then add new value

          lhld  code
          dad   h           ;times 2
          mov   e,L         ;save in DE
          mov   d,h
          dad   h           ;times 4
          dad   h           ;times 8
          dad   d           ;add times 2 to get times 1Ø
          mvi   d,Ø         ;new value in DE
          mov   e,a
          dad   d           ;add to old value times 1Ø
          shld  code        ;save result
          ret
```

Listing 3

```
talk:     mvi   a,3Øh         ;bit 5 high enables the
                               Votrax chip
          out   rØ            ;keep bit 4 high to disable
                               DTMF
                              ;encoder

          lxi   h,phrase      ;beginning of phonemes
          mov   a,m
          cpi   128           ;indicates end of phonemes
          jz    done          ;can't use Ø since it's..
                              ;..a phoneme code
          out   r2            ;output phoneme code

phover:   in    rØ            ;now wait until Votrax..
          ani   2Øh           ;..buffer empty. Indicated..
          jz    phover        ;..by bit 5 going high

          inx   h             ;get next phoneme
          jmp   talklp        ;continue

done:     mvi   a,1Øh         :disable Votrax (bit 5 low)
                               and..
          out   rØ            ;..DTMF encoder (bit 4 high)
          .
          .
          .

phrase:   db    1Ch, 16h, 16h, 1Eh       ;"good"
          db    3Fh                      ;no sound  -
                                          produces a 47..
                                         ;..millisecond
                                          delay
          db    ØCh, 34h, 34h, 2Bh, ØDh  ;"morning"
          db    128                      ;end of string
```

VEDIT

CUSTOMIZABLE
FULL SCREEN EDITOR

The Professional's Editor for Program Development Word Processing Source Code Translations

Widely acclaimed as an editor, VEDIT has evolved to be much more. Only VEDIT offers the combination of a versatile full screen editor integrated with a powerful command language. For the first time you'll be able to perform complex, yet useful, text manipulations that are virtually impossible with other editors or word processors. Plus, its customizability and hardware support ensure that VEDIT will be perfectly matched to your individual needs and to any microcomputer you are ever likely to own.

With two modes of operation, VEDIT never compromises its speed or ease of use for its power and sophistication. As one reviewer (Bradford Thompson, BYTE) wrote: 'If this review gives you an appetite for simplicity while editing, then VEDIT is well worth considering.' Its command language, based on TECO, is virtually a text oriented programming language, allowing command macros to be created, loaded and saved on disk. Yet its simplicity allows even a novice to perform tasks beyond the capabilities of any word processor.

VEDIT cuts programming time in half - with multiple file handling, macro capability and special features for Pascal, PL/1, 'C', Cobol, Assembler and other languages. And it can help with source code translations (example ZILOG to/from INTEL translator macros are included). A complete line of translators will be available by the year's end.

Word processing is a snap with word wrap, paragraph and print functions. Command macros free you from tedious search/replace operations. Hundreds of search/replace on dozens of files can be performed by VEDIT without waiting or intervening.

VEDIT easily configures to your favorite keyboard layout. Use any function or cursor keys you wish. It optimally supports nearly every 8080, Z80 and 8086 computer.

Go ahead and expect a lot from VEDIT. Its performance and our support will satisfy your most exacting needs.

To order, please specify your 8080, Z80 or 8086 microcomputer, operating system and disk format.

COMPARE VEDIT'S FEATURES

- True Full Screen Editing
- Horizontal scrolling
- Edit files one disk in length
- Automatic Disk Buffering
- Compact (only 16K) and Fast
- Display of line and column #
- Set/Goto text markers
- 'Undo' key to restore line
- Automatic Indent/Undent
- Adjustable Tab positions
- Repeat function key
- Text Move and Copy
- 10 Scratchpad Buffers
- Load/Save buffers on disk
- Powerful command macros
- Directory display
- Edit additional (small) files simultaneously
- Insert another disk file
- Unlimited file handling
- Recovery from 'Full Disk'
- Change disks while editing
- Word wrap, format paragraph
- Simple Printing
- 150 page indexed manual
- Startup command file
- Menu driven installation
- Program CRT function keys
- Support newest CRT terminals
- Support smart CRT functions
- Flexible Memory Mapped support
- Customizable keyboard layout

IBM PC, Displaywriter ° Zenith Z100 and Z89 ° NEC APC ° DEC Rainbow and VT180 ° Televideo 802
TRS-80 I, II and 16 ° Xerox 820 ° Apple II Softcard ° SuperBrain ° NorthStar
MP/M ° CP/M-86 ° MP/M-86 ° Concurrent CP/M-86 ° Cromix ° Turbo DOS ° MSDOS ° PCDOS

VEDIT - Disk and Manual
8080, Z80 or IBM PC $150
CP/M-86 or MSDOS $195
Manual only $18

CompuView
PRODUCTS, INC.

1955 Pauline Blvd., Suite 200 ° Ann Arbor, Michigan 48103 ° (313) 996-1299

CIRCLE 35 ON READER SERVICE CARD

Software Directory

Program name: HDIS
Hardware system: CP/M 2.x
Minimum memory size: 60K
Language: Object code
Description: HDIS produces labeled, correctly disassembled 8080 source code automatically. Besides generating and verifying the labels themselves, HDIS automatically shifts between instruction and data disassembly during creation of the source code and finds unlabeled character strings.

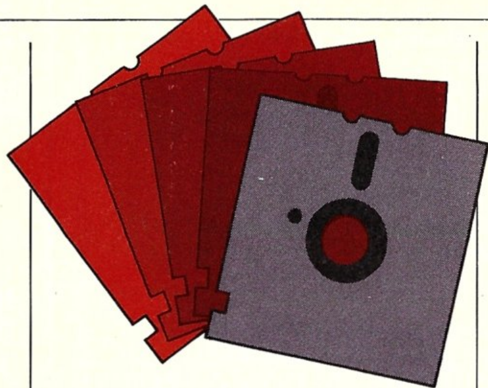
HDIS disassembles an 8080 machine code program directly from disk, in either a continuous or an interactive mode. In continuous disassembly mode, HDIS outputs source code continuously until the entire machine code program has been disassembled. Disassembly in the interactive mode pauses so the operator can control its state and progress. Other aids include going to other parts of the program and saving the label tables on a disk file.

The interactive mode also allows modifications of disassembly conditions, such as selection of console or printer output, console screen height and width, etc. Output may be directed to the CRT, printer, written as an .ASM file, or any combination of the three.
When released: March 1983
Price: $85; manual, $10 (applies to purchase)
Included with price: disk and manual
Where to purchase it:
Elliam Associates
24000 Bessemer St.
Woodland Hills, CA 91367
(213) 348-4278

CIRCLE #311 ON READER SERVICE CARD

Program name: CERES I
Hardware system: Z80 CP/M system
Minimum memory: 48K
Language: Z80 assembler
Description: CERES I is an interactive screen design and process utility that interfaces with many popular Cobols. With CERES I's screen design utility, screens are designed directly on the terminal. Each field is interactively positioned on the screen. Fields can be inserted, deleted, moved, or copied on the screen for fast and easy screen design and maintenance. CERES I's runtime system is called by the Cobol program and accepts, formats, displays, and extensively validates data. CERES I supports backup by field for data entry correction and help messages for data entry clarification. It also supports date editing, default values, and minimum/maximum value checking as well as many other edit validations.
When released: March 1983
Price: $300. Manual alone, $25.
Included with price: Programs with tutorial and reference manual.
Where to purchase it:
Ceres Software, Inc.
P.O. Box 1629
Portland, OR 97207
(503) 245-9011

CIRCLE 312 ON READER SERVICE CARD



Program name: NAMOR, Mailing List Program.
Hardware system: 8080/8085/Z80 CP/M (version 2.2) or MP/M system.
Minimum memory size: 56K CP/M system; program requires 45K.
Language: Machine code.
Description: NAMOR is a program for handling all name and address list management tasks. Five menu-driven screens give the user complete control of the list processing without the need to remember special instructions or codes.

Advanced features include a special new-name enter, review and correct section; a multilevel sort on any or all of the 10 elements; merge capability for combining list while maintaining sort order; full easy-to-use update capability; extensive search/select capability for printing or separating sub-list; match-search to locate duplicates; tag feature for marking entries; special code line handling features; and extensive formatting of the label or directory printout. A separate formatting program allows the list files to be used with the popular word processors.
Release: November 1982
Price: $150. Manual and training/demo disk only, $25.
Included with price: NAMOR disk, training/demo disk and manual.
Where to purchase:
SHAPE, Inc.
122 Spanish Village,
Suite 615
Dallas, TX 75248
(214) 644-6599

CIRCLE #313 ON READER SERVICE CARD

Program name: HASTE®
Hardware system: CP/M-80 with interrupts and synchronous modem
Minimum memory size: 64K
Language: Object code
Description: HASTE allows a CP/M-based microcomputer to communicate using HASP bisynchronous protocol. Using straightforward menus, you can direct incoming data to a disk file, console, printer/plotter, or any combination of the three. Data on a disk file can be sent in several ways, depending on whether HASTE is acting as a host or as a remote. A screen-oriented text editor allows jobs of 17 lines or fewer to be entered and sent without leaving HASTE.
When released: February 1983
Price: $500; demo disk, $30;

Software Directory
continued . . .

manual only, $25
Included with price: disk and manual
Where to purchase it:
Florida State University Computing Center
Tallahassee, FL 32306
(904) 644-4836
CIRCLE #314 ON READER SERVICE CARD

Program name: ASCOM
Hardware system: CP/M, CP/M-86, MS-DOS, PC-DOS
Language: written in assembly; menu-driven
Description: ASCOM is a versatile asynchronous communication package for microcomputers. It features interactive, menu-driven and batch operations; supports auto-answer and auto-dial modems; includes most popular protocols; provides network simulation; and many other options.

ASCOM enables the user to log on to a timesharing system to receive or send data; communicate any data files and programs, including binary (.COM) files; log on to a remote computer running ASCOM and control the remote computer from one's local computer; use auto-answer and auto-dial modems to perform communication functions automatically; instruct the computer to dial another computer system repeatedly until it answers.

ASCOM includes menus that permit the user to set up the communication port (baud rate, parity, etc.); to set the operating mode (batch, capture, command, remote, and local); select communications commands (receive, send, conversational mode, dial, answer, etc.); use operating system commands without leaving ASCOM (delete, directory, rename, reset, run, type, etc.); and call up HELP for descriptions of all commands.
When released: 1981; changed to menu driven in September 1982
Price: software plus documentation, $175; documentation alone, $25.

Linked at Last!

MSPRO™ unites MS-DOS® and CompuPro®.

Now, lovers of **CompuPro**® hardware can be users of most IBM software. The **MSPRO**™ subsystem turns a clash of titans into a league of nations, by making the world of **MS-DOS**® based software available to **CompuPro**® users for the first time.

MSPRO™ supports your existing 8-inch drives and up to four 5-1/4 inch drives for transferring files. Includes one 5-1/4 inch disk drive in enclosure with power supply ($325 single sided 48 tpi, $395 double sided 48 tpi), disk controller board ($350), **MSPRO**™ operating system diskette ($395, with cables and manual). Dealer inquiries invited.

MS-DOS is a registered trademark of Digital Research Inc.
CompuPro is a registered trademark of Godbout Electronics Inc.
MSPRO is a trademark of Computer House Inc.

See how the other half computes. Order **MSPRO**™ today, from **Computer House** (a Full Service CompuPro System Center), 722 Vie St., San Rafael, CA 95901. Call **(415) 453-0865** for further details.

CIRCLE 30 ON READER SERVICE CARD

Workman & Associates
112 Marion Avenue, Suite 2B
Pasadena, CA 91106
(213) 796-4401

The File Transporter

If you own more than one machine, you know how difficult it is to move files between them. One copy of The File Transporter will move any file between CP/M machines. It'll even send to CP/M-86 machines! Requires matching ports (serial or some parallel) or modems. A very detailed manual is included. The File Transporter is $69.50.

"A Primer on Pascal for CP/M Systems" by Alexander and Jerry Pournelle

Full of examples and suggestions to make learning Pascal easier. Contains both a disk and a detailed manual with a glossary and an error-correcting guide.

NEW PRICE! Discounted from $79.50 to $64.50!!!

Minimum Database

Don't want to spend $400 on dBase-II to organize your mailing lists? Would you like to have the source to your recipe keeper? The Minimum Database is what you need. It keeps track of almost any database and comes with the CBASIC-II sources.

~~[illegible]~~ **Now only $49.50!**

Disk formats include: 8", Apple CP/M, Osborne, Xerox, KayPro, Monroe, and Otrona. Please request our catalog.

CIRCLE 278 ON READER SERVICE CARD

THE V.I.P.* IS HERE!

*Versatile Instrumentation Peripheral

Perfect for Use in Process Control and Data Acquisition Systems or in Standalone Instrumentation

High Performance and Resolution

Extremely Cost Effective

- ★ Single Card S100 A/D, D/A, Digital I/O
- ★ Instrumentation Amplifier Input Channel
- ★ D/A Voltage and Current Source Output
- ★ Full 12 Bit Resolution
- ★ Software Configurable Analog Circuitry
- ★ 25 μs A/D Conversion Time
- ★ 48 High Voltage, High Current Digital Drivers, 100mA per Output
- ★ Drives LED Display and Decimal Points with Software Selectable ROM Decoding
- ★ Two Double Pole Utility Relays
- ★ 8 Bit TTL-Compatible Input
- ★ Self-Test Capability

$595

Fully assembled and tested. Documentation and applications software provided. Check, MO, MC, Visa accepted. MA residents add 5% sales tax, $4 shipping and handling charge.

AUTOMATED CONTROL SYSTEMS
1105 Broadway
Somerville, MA 02144
(617) 628-5373

CIRCLE 200 ON READER SERVICE CARD

PROGRAMMERS

"C" LANGUAGE programmers were surveyed. The products featured below are what they preferred. We carry all MSDOS & CPM software & addons for programmers.

If we don't stock it, we'll find it-at a discount.

Products for "C" Programmers	List Price	Our Price
Compilers: All have library source, no royalties		
8080 - BDS C - compile in memory, good library	$150	**$125**
Z80 - EcoSoft C - Full K&R, fast code., .REL files	350	**315**
APPLE - Aztec C - Float, doubles, editor included	199	**179**
8086 - C86 - PCDOS/CPM-86 - Full K&R, OPTIMIZER with ASM out	395	**Call**
8086 - Lattice - PCDOS - Full K&R	500	**459**
Editors: 8080 & 8086 - C Screen Editor w/source	NA	**60**
VEDIT/PC and CPM-80	150	**125**
VEDIT/86	200	**175**
Utilities: 8080/8086 - POWER - 55 programs	150	**129**
MICROSHELL - add UNIX features to CPM-80	150	**125**
CACHE/Q - CPM-80 & PC - automatic RAMdisk, spool, makes all run faster	225	**185**
UT-86 - Patch, List, etc. 8080 emulator	180	**159**
Communications: ASCOM - 8080/86 - flexible, powerful, main frame & micro	175	**145**
HAYES Smartmodem 300	289	**227**
Smartmodem 300/1200	699	**530**
Diskettes: We carry all brands. Convenient service		
Double Sided, Double Density - 8 inch	70	**39**
- 5.25 inch 48 TPI (PC)	65	**32**
Single Side, Single Density - 8 inch	45	**26**
- 5.25 inch (APPLE)	40	**24**
Printer: 100 cps - all features of MX80 FT, ProWriter	495	**395**

Call for the Programmer's Referral List, a catalog, literature, advice or prices. We're here to help. Add $2.50 for shipping.

THE PROGRAMMER'S SHOP™
135 Main Street, 3M
Maynard, MA 01754, 617-897-4750

VISA **800-421-8006** **MASTER CARD**

CIRCLE 5 ON READER SERVICE CARD

Included with price: disk and manual
Where to purchase it: major dealers and distributors; also:
DMA
545 Fifth Avenue
New York, NY 10017
(212) 687-7115

CIRCLE #315 ON READER SERVICE CARD

Program name: KEYDISK®
Hardware system: CP/M-80
Language: Object code
Description: A data entry driver program that transforms a CP/M system into a unit record entry station. This powerful column-oriented screen editor is easy to use, especially for operators used to unit record equipment (such as a keypunch). KEY2DISK makes data entry simple to set up and simple to do. Standard CP/M text files are produced. A verification feature is included.
When released: February 1983
Price: $250; demo disk and manual, $30; manual only, $25
Included with price: Disk and manual
Where to purchase it:
Computing Center
Tallahassee, FL 32306
(904) 644-4836

CIRCLE #316 ON READER SERVICE CARD

Program name: SUPERFILE
Hardware requirements: CP/M, Z80 CPU, 2 disk drives
Minimum memory: 48K
Description: SUPERFILE is an information retrieval system that works like an "electronic filing cabinet." SUPERFILE uses a set of keywords that allow data to appear to be in many places at the same time. This is the ideal filing system. With SUPERFILE you can find the letter, magazine article, recipe, report, file or data you want on the screen of your CRT. Just type in something that occurs on the document you are looking for, like the Title, Date, Name, Subject, or Ingredient. All documents containing those keywords are identified and listed on the screen. Select the one you need and the entire document will be displayed or printed.

Use SUPERFILE for magazine articles, correspondence index, medical histories, recipe file, property management, customer profiles, sales leads, personnel records, contact reports, mailing lists, card files, vendor information, real estate listings, and legal contracts.

The User's Manual is a 3-ring binder with a Table of Contents, Index, and over 100 double-sided pages divided into 12 sections. Every possible subject is covered, from "Set-Up" to "Learning Superfile," "Creating a Data Base," "Utilities," and "Practical Uses."
When released: April 1982
Price: $195
Included with price: DEMO data base, the Manual, and a 30-day money back guarantee.
Where to purchase it:
Digital Color Corp.
2252 Main St., Suite 15
Otay, CA 92011

CIRCLE #317 ON READER SERVICE CARD

Programming name: Z-COM
Hardware system: Morrow Micro Decision & other Z80 CP/M 2.x systems
Language: Object code
Description: Z-COM translates a BAZIC program into an assembly language program and thence into a fully operational machine language program. The resulting programs run faster than their BAZIC equivalents and as machine code fully protect the original source program. The only major restriction imposed on the program to be complied is that variable dimensions must be decimal constants.

Z-COM is personalized for the terminal supplied with the Morrow Micro Decision (ADM-31 emulation) but can be easily adapted to other terminals. The Z-COM compiler system includes the Z-COM compiler, MAKRO relocating macro assembler, CLINK linking loader, compiler support library, and a suite of applica-

tion support routines that perform string sort, string search, matrix operations, and extended precision arithmetic. Programs developed with Z-COM for commercial distribution must bear notice to that effect, but otherwise require no royalties. No runtime package is required for compiled programs. Complete documentation is included, and full user support is provided by mail or phone.
When released: March 1983
Price: $200
Included with price: Documentation, disk, and user support
Where to purchase it:

Allen Ashley
395 Sierra Madre Villa
Pasadena, CA 91107
(213) 793-5748

CIRCLE #318 ON READER SERVICE CARD

Program name: SDAM/II
Hardware system: CP/M 2.2
Minimum memory size: 48K
Description: Data management system with interactive modules to create a data base scheme, add data entries, retrieve data entries, create screen and printer report formats, and generate corresponding reports. Keys are maintained dynamically without operator intervention. A compression module removes deleted entries and repacks the data base. A subroutine module allows other applications access to the database.
When released: February 1983
Price: $149
Included with price: On-line modules, source-code subroutine, manual
Where to purchase it:

Computer Development Specialists of Long Island, Inc.
P.O. Box 999
Centereach, NY 11720
(516) 732-2407

CIRCLE #319 ON READER SERVICE CARD

Program name: SUPERFILE System 3
Hardware system: Z80 CPU and CP/M 2.x, 2 disk drives; IBM-PC, PC-DOS 1.1, 2 disk drives
Description: SUPERFILE System 3 offers easy updating and automatic reindexing of up to 65,000 records per data base. Each free-format variable-length record can have up to 500 of the user's own keywords, with a maximum of 32,000 keywords per data base.

SUPERFILE System 3 includes Post Haste®. Post Haste works with SUPERFILE System 3 or Micro-Pro's Mail Merge to provide formatting and printing of mailing labels, etc. Post Haste sorts alphabetically or by Zip code.
When released: January 1983
Price: $295
Included with price: Disk and manual
Where to purchase it:

FYI, Inc.
P.O. Box 26481
Austin, TX 78755
(512) 346-0133
(800) 531-5033

CIRCLE #320 ON READER SERVICE CARD

Program name: Relocation utility package
Hardware system: 8080/Z80 CP/M
Minimum memory: Variable (48K-64K), since all buffers are allocated dynamically
Language: Supplied as COM files
Description: A set of four programs allows the user to examine the contents of a REL file or library module. The output may be to the console, printer, or a disk file. The documentation includes information on REL file encoding procedures.

REL/MOD gives entry points, segment sizes and externals of a module.

REL/BIT displays the bit stream data in binary form, partitioned by field and with short comments for field interpretation.

REL/VUE produces a pseudo-assembly language output which has a nearly 100% correspondence with the input. This is very useful for testing the performance of a compiler or assembler, since the data is viewed with no interpretation.

REL/MAC produces an 8080 assembly language source file with reinsertion of public and external symbols. If the original REL file was created by M80, the output of this program will almost always reassemble to the same REL file. The source produced by REL/MAC can be edited with any text editor.
Price: $99.95 (documentation only, $10, and will apply toward purchase price).
Included with price: Manual and 8″ SSSD disk
Where to purchase it:

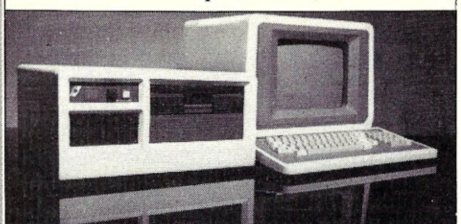
Microsmith Computer Technology
P.O. Box 1473
Elkhart, IN 46515

CIRCLE #321 ON READER SERVICE CARD

New Products

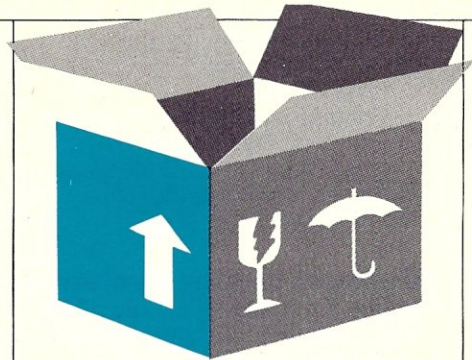
S-100 desktop computer

Sierra Data Sciences has introduced a highly compatible, low-cost multiprocessor system with unusual expansion and communication capabilities. Its new SDS-100 system is available in a wide range of configurations, all based on its SBC-100 S-100/IEEE-696 bus, single-board computer in stand-alone systems, or used with up to 16 SBC-100S slave boards for expansion in slave/satellite networks. On each card are the Z80A 4MHz processor, 64K of



RAM, and 4K to 16K of EPROM, two serial RS-232 channels, two parallel channels, four counter/timers, floppy disk controller and hard disk interface. Therefore, no bus is required for stand-alone operation, and it can act in multiprocessor configurations as a master CPU, co-master CPU, or intelligent peripheral controller.

Any of the desktop units can be variously configured to have up to four DSDD Shugart-compatible floppy disk drives (8″ and 5¼″), Micropolis (8″) Winchester or ST506-compatible (5¼″) hard disk drives for on-line storage and back-up capabilities ranging up to 180MB. This is a construction design of the SDS mainframe enclosures, which feature filtered power supply, filtered



forced-air circulation, LED display, momentary reset switch, key-lock power switch, and a 12-slot IEEE motherboard. The systems operate under Sierra Data Sciences supported CP/M or TurboDOS.

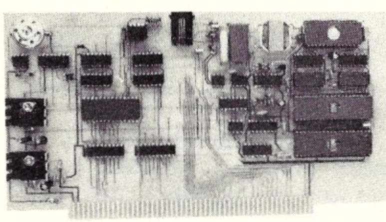
Prices vary according to configuration, quantity and peripherals supplied. SDS also offers all boards and enclosures separately.

Sierra Data Sciences, Product Support Group, 21162 Lorain Ave., Fairview Park, OH 44126; (216) 331-8500.

CIRCLE #301 ON READER SERVICE CARD

300/1200 baud auto-dial modem

U.S. Robotics has introduced an auto-dial/auto-answer modem created specifically for use with an S-100 based computer. The USR model 212 S-100



modem board snaps into any standard S-100 rack.

The S-100 modem automatically dial/answers and transmits at 300 or 1200 baud. At high speed, the unit can process about three or four typewritten passages per minute. It operates at full or half duplex (local echo) and contains an audible phone line signal monitoring system.

Its auto-dial feature permits the user to program his computer for automatically generating any individual or several sequential phone numbers, and to perform many other dialing functions unattended. USR's custom software package, Telpac, is designed for use with this and other USR auto-dial modems.

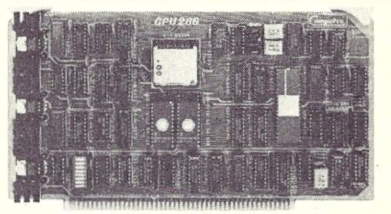
Price: $599

U.S. Robotics, Inc., 1123 W. Washington Boulevard, Chicago, IL 60607; (312) 733-0497.

CIRCLE #302 ON READER SERVICE CARD

iAPX 286 S-100 CPU board

The CompuPro division of Godbout Electronics has introduced an S-100/IEEE-696 CPU board based on Intel's iAPX 286/10 microprocessor. Designated CPU 286, the new board's features include sockets for an 80287 math coprocessor and up to 16K of EPROM. It is completely software-compatible with code written for Intel's



8086 and 8088 processors. Clock rate for the CPU 286 is 8MHz for the standard version and 10MHz for the high-reliability version. A clock-switching circuit permits 8-bit or 16-bit slave processors to run on the same bus at various clock rates without timing conflicts, enabling users to execute alternate software libraries.

With a 24-bit address and 16-bit data bus, the CPU 286 can access as much as 16 MB of on-line system memory without any segmentation. The unit's on-board logic can also read or write two bytes serially to simulate 16-bit operation in the presence of 8-bit memory of I/O. It also incorporates a memory management unit (MMU) that permits high-speed multiuser multi-tasking operation as well as memory protection for each workstation; its supervisory function allows

New Products
continued . . .

the operating system to protect users from destroying data on other workstations.

Price: $1,595.

CompuPro Division, Godbout Electronics, Oakland Airport, CA 946141 (415) 562-0636.
CIRCLE #303 ON READER SERVICE CARD

Dual-processor system

Magic Computer Co., Inc., will begin delivery of a dual-processor system, available with either one, two, three, or four floppy disk drives. The single-disk 320K model is designated the PBC-88/1. The main processor is a Z80A operating at 4MHz with a 64K user memory. The system has an auxiliary 3K display memory driven by a

separate 6502 processor. RS-232C and Centronics parallel printer ports are also included in the basic system. CP/M Plus and CBasic are supplied with the PBC-88/1.

The PBC-88/2 adds a second 320K drive to the system, as well as Perfect Writer, Perfect Speller, Perfect Calc, and Perfect Filer (registered trademarks of Perfect Software, Inc.).

Both systems include a high-resolution 12″ monitor that displays 24 lines of 80 upper- and lower-case characters, formed by a 7 X 9 dot matrix. The monitor is also sold separately for use with other systems.

Prices: PBC-88/1, $2,245; PBC-88/2, $2,995; monitor (separately, $299.

Magic Computer Co., Two Executive Drive, Fort Lee, NJ

New Products
continued . . .

07024; (201) 944-6700.
CIRCLE #304 ON READER SERVICE CARD

Programmable current loop converter

A completely programmable RS-232 to current loop converter has been announced by Remark Datacom Inc. The programming versatility of the Model 64 allows any computer to interface with current loop peripherals or utilize their internal current loop interface circuitry to drive remotely located RS-232 based peripherals such as printers, CRTs, ticket readers, etc.

Current loops provide noise immunity for long-distance (up to 1 mile) cable runs between the host CPU and the peripheral. Current loop interfaces can

be one of four types: half duplex-passive loop, full duplex-passive loop, half duplex-active loop, and full duplex-active loop. In addition the RS-232 port may be required to support a DTE or DCE interface.

The Model 64 contains switches to allow the user to program the type of loop to either active or passive and the communications protocol to either half or full duplex. The RS-232 port is switch configurable to DTE or DCE.

In the active configuration the Model 64 supplies the loop current, whereas in the passive mode the external device must supply the current. In the passive mode the 64 provides total electrical isolation between subsystems via opto-couplers. The Model 64 provides for 20 mA or 60 mA current loops and operates from DC to 9600 bits per second.

Connection to the converter is via a male DB25 type con-

New Products
continued . . .

nector which allows the unit to be attached directly to most peripherals without the need of a separate interface cable. Current loop connections are accommodated at a five-position terminal strip which gives the user the flexibility for accepting the various loop structures. The Model 64 derives its operating power from the host device via pins 9 and 10 of the RS-232 connector. The power inputs are protected against inadvertent application of reverse polarity voltages.

Price: $79.95

Remark Datacom Inc., 4 Sycamore Drive, Woodbury, NY 11797; (516) 367-3806.
CIRCLE #305 ON READER SERVICE CARD

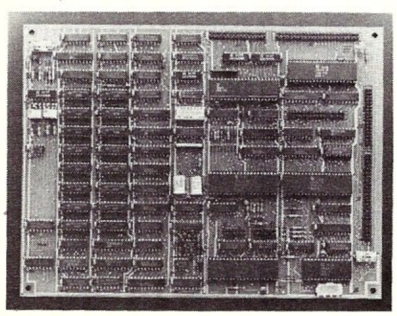
CP/M Plus based SBC

A new CP/M Plus based single board computer by Insight Enterprises Corporation, called the Equalizer EQ4, has on-board video with wide line and thin graphics capability, a Sasi interface, composite and separated video, and eight attributes per character.

The EQ-4 provides a Zilog Z80-A CPU, 128K of RAM, the entire Z80-A family of peripheral controllers (DMA, CTC, DART, and PIO), 4 RS-232 serial ports, 2 parallel ports, floppy disk controller for 5¼ and 8" drives, single and double density, extended track buffer to ensure fast disk access, and 16K of printer buffer for a more efficient system operation.

Price: $750 for a limited time only. OEM and dealer prices also available.

Insight Enterprises Corp., 373 N. Western, Suite 12, Los Angeles, CA 90004; (213) 461-3262.

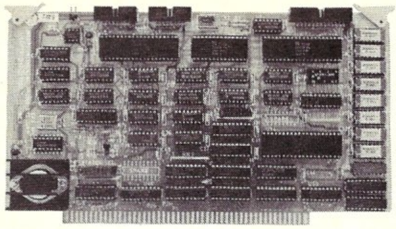


New Products continued . . .

3262.
CIRCLE #306 ON READER SERVICE CARD

S-100 slave SBC

Memory transfer rates of 517 KB/sec, direct memory access, memory mapping, and host-to-slave requests via interrupt control make the new CPS-MX from Intercontinental Micro Systems very fast. Fully compatible with TurboDOS, the CPS-MX slaves are available in four versions, synchronous or asynchronous serial ports, 4MHz or 6MHz. The CPS-MS



also allows the bus master to use slave memory at the user's discretion. The slave then acts as a 65K RAM card.

The CPS-MX is also easy to integrate with Intercontinental's full line of S-100 products: CPZ-48000SBC with 64K on-board RAM, 4 I/O channels, memory management, on-board floppy controller, DMA and vectored priority interrupts, and 256K bank-selectable or contiguous memory. A complete line of personality boards allows easy interface to a wide body of peripherals and doesn't take up S-100 bus space.

Prices start at $475.

Intercontinental Micro Systems, 1733 S. Douglass Rd., Suite E, Anaheim, CA 92806; (714) 978-9758.

CIRCLE #309 ON READER SERVICE CARD

THE UNIX™ GUIDE

An inexpensive way to discover the benefits of UNIX System III

- Orients computer users to UNIX software
- Special coverage of the UniSoft Edition with Berkeley enhancements
- Application oriented, clearly written and concise
- Helps readers determine that they are buying full UNIX System III
- Complements standard UNIX reference documentation

Order the guide by sending a check for $25.00 payable to:

PACIFIC MICRO TECH
5819 Poinsett Ave.
El Cerrito, CA 94530
(415) 233-3596

*UNIX is a trademark of Bell Laboratories

S-100 TIMEKEEPER

Plug-In Module

Time, day & date are accessible to your programs without the need to re-enter them each time your micro-computer is turned on. Listings of **SETTIME. & READTIME. BAS** are included with complete documentation. $124.95 plus $2.00 for shipping.

- also available for -

NORTH STAR ADVANTAGE

$139.95 Plus $2.00 for shipping.

PRINTER SELECTOR SWITCH
assembly (RS232) including cables from $69.00

ACTIVE SYSTEMS
Rt 2 Box 343 Afton, Va. 22920
(804) 361-1180

PROGRAM EPROMS FROM CP/M† OR MP/M† USING EPM

EPM IS A SOFTWARE MONITOR THAT INTERFACES HOMEBREW OR COMMERCIAL EPROM PROGRAMMERS TO CP/M OR MP/M. BINARY OR HEX FORMAT, MEMORY OR I/O MAPPED EPROM PROGRAMMING HARDWARE INTERFACED THROUGH SIMPLE I/O ROUTINES. SKELETAL I/O ROUTINES ARE PROVIDED IN SOURCE FORM FOR BINARY AND HEX FORMAT PROGRAMMERS.

FEATURES OF EPM:

- EPROM programming from CP/M or MP/M disk files
- EPROMS can be read to a disk file for archiving
- Programming without address or EPROM length calculations
- Menu operation and extensive operator messages for ease of use
- Blank EPROM verification before programming
- Verification of EPROM after programming
- User written I/O routines can be installed for varying hardware
- Virtually any EPROM type can be programmed using EPM
- EPROM editor option permits byte level modifications
- HEXROM utility can convert a hex file ORG'ed anywhere in memory
- Runs under CP/M version 2.x or MP/M II with at least 24 K of RAM

EPM ON 8 INCH S.D.DISK WITH COMPLETE DOCUMENTATION -$75
EPROM EDITOR OPTION -$45
SHIPPING-$2.50
C.O.D.-$2.50

For More Information Contact:

DANTEK
SOFTWARE,INC
4550 SCHOOLHOUSE ROAD
BATAVIA, OHIO 45103
(513) 752-1921

† CP/M and MP/M are trademarks of Digital Research

ISIS ⟷ CP/M®

Full bi-directional file transfer capabilities are provided in the ISIS-CP/M utilities package. Written in machine language and running under CP/M, these utilities permit the CP/M user to read or write files direct to/from an ISIS Diskette. They will run under any version of CP/M without regard to diskette density. The complete package is $250.00 including user's manual. Write for free brochure on other CP/M software.

CP/M is a registered trademark of Digital Research
ISIS is a trademark of Intel Corporation

SOUTHERN COMPUTER SYSTEMS, INC.
P.O. Box 3373A
Birmingham, AL 35255
(205) 933-1659

I WILL BEAT ANY COMPETITOR'S PRICE PROVIDED IT IS NOT BELOW MY COST. TRY TO BEAT THESE IC PRICES:

DYNAMIC RAM		
64K	200 ns	$4.79
64K	150 ns	4.99
64K	120 ns	5.90
16K	200 ns	1.25
EPROM		
2764	250 ns	$7.50
2732	450 ns	4.15
2716	450 ns	3.25
2532	450 ns	4.70
STATIC RAM		
6116P-3	150 ns	$4.24
2016	100 ns	4.00
2114	200 ns	1.60
Z80A FAMILY		
CPU, CTC, or PIO		$3.39
DART		8.25
DMA or SIO/0		12.50

MasterCard/VISA or UPS CASH COD
Factory New, Prime Parts

MICROPROCESSORS UNLIMITED
24,000 South Peoria Ave.
BEGGS, OK. 74421
(918) 267-4961

Prices 3/1/83

Prices subject to change. Call for volume prices. Subject to available quantities. Shipping & Insurance extra. Cash discount prices shown.

NORTHSTAR

INTEGRATED BOOKKEEPING

- General Ledger
- Accounts Payable
- Accounts Receivable
- Payroll

EASY TO USE!

Auto Merge into General Ledger
Auto Start
Written in Basic
Horizon or Advantage
CP/M or DOS, HD or Floppy

Detailed description available
$500. — DOS version.

Power's Computers
12491 San Pablo Avenue
Richmond, Ca. 94805

(415) 234-5412

Electronic Circuit Analysis

- DC and AC analysis
- Very fast, machine language
- Infinite circuits on multiple passes
- Worst case, sensitivity analysis
- Dynamic modification
- 64 Nodes, 127 branches
- Compare circuits
- Log or linear sweep
- Full file handling
- Frequency response, magnitude and phase
- Complete manual with examples
- CP/M $150.00

Tatum Labs
P.O. Box 722
Hawleyville, CT
06440
(203) 426-2184

FOR

CompuPro

IT'S

COMPUTER HOUSE

501 "B" Street
San Rafael, Ca. 94901

(415) 453-0865

We specialize in CompuPro products at excellent prices.
Call or write for Price List.

Computer House is an authorized CompuPro Systems Center.

SOURCE SOFTWARE

A professional-quality, CP/M compatible Z-80 assembler in a 200-page manual containing complete source listing.

Advanced techniques fully explained in accompanying tutorial include Radix 40, binary search of symbol table, expression processing by recursive descent, etc.

Modular construction for flexibility and easy revision as a cross assembler.

Manual alone - $25, ppd. in U.S.
(foreign orders add $2 surface, $10 air)

Source also available on standard format 8" SSSD diskette for $10 plus $2 shipping

KING SOFTWARE
PO Box 208
Red Bank, N.J. 07701
(201) 530-7245

NJ residents please add 6% sales tax
CP/M is a trademark of Digital Research

classified ads

CLASSIFIED RATES: Per Word, 15 Word Minimum. REGULAR: $1.00. EXPAND-AD: $1.50. GENERAL INFORMATION: Prepayment discounts available. Payment must accompany order except credit card — Am. Ex., Diners, MC, VISA (include exp. date) — or accredited ad agency insertions. Copy subject to publisher's approval; must be typewritten or printed. First word set in caps. Advertisers using P.O. Boxes MUST supply permanent address and telephone number. Orders not acknowledged. They will appear in next available issue after receipt. Send order & remittance to: Classified Advertising, MICROSYSTEMS Magazine, 1 Park Avenue, New York, NY 10016. Direct inquiries to: Kenneth L. Novotny (212) 725-3927.

COMPUTER NETWORKS

THE CRITICAL CONNECTION! Now you can buy an inexpensive ATARI-400, and have it use the disk drives, printer, and keyboard on your CP/M computer. Complete package $175.00 DEALER INQUIRIES WELCOME. USS ENTERPRISES, 6708 Landerwood Lane, San Jose, CA 95120, (408) 997-0264.

COMPUTER EQUIPMENT/SUPPLIES

COMPUTER PAPER—Save $$$: Lowest single carton prices. Free samples. Bob's charts, 405 East Third, Long Beach, CA 90802.

BUSINESS OPPORTUNITIES

COMPUTER CASH—101 exciting, new, sparetime home businesses. Report. $4.50, Associated Research, Box 22485MI, Houston, TX 77227.

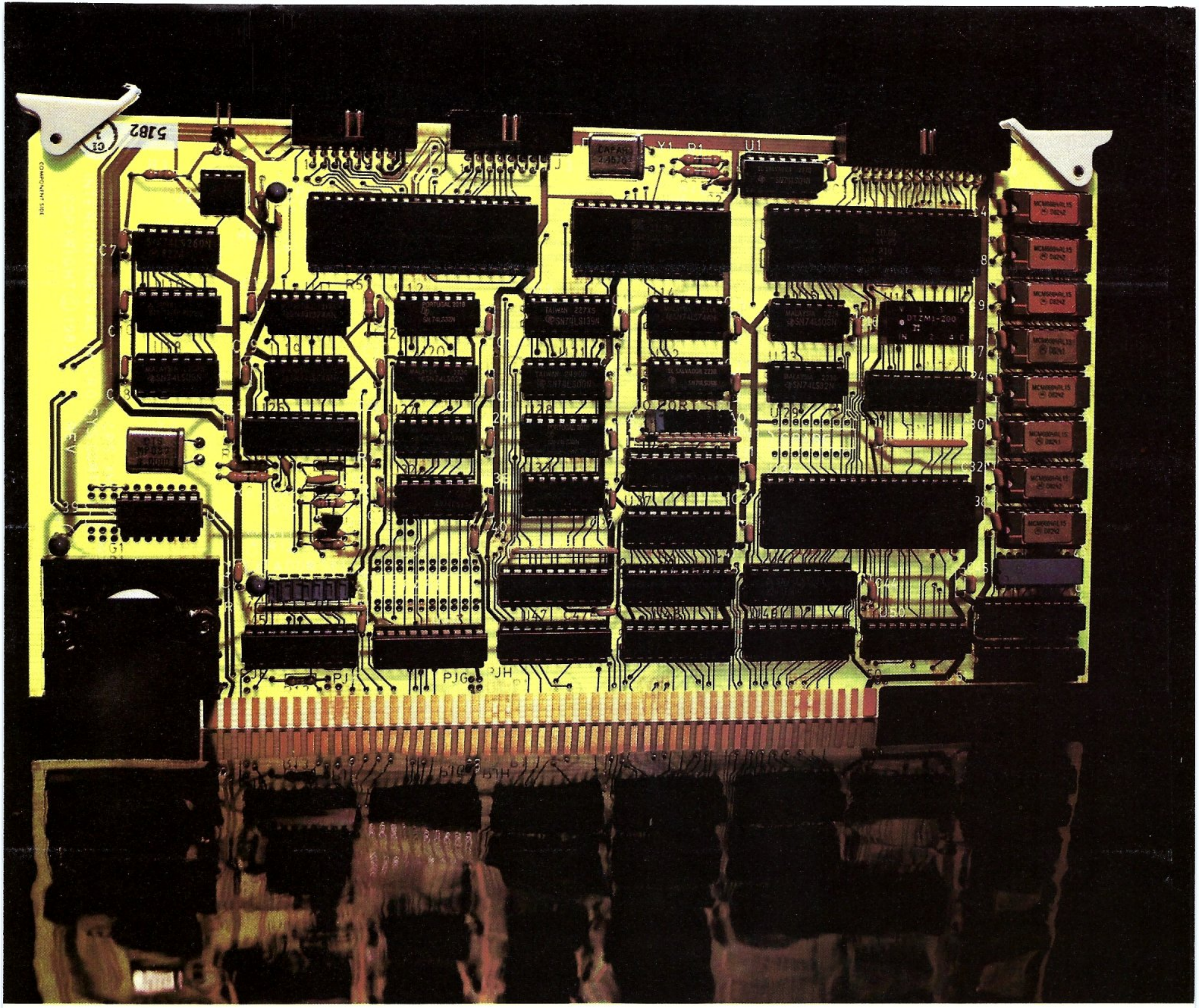
MISCELLANEOUS

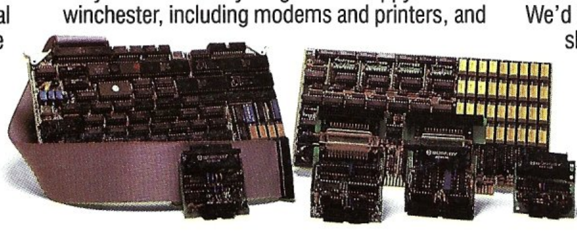
BUYING back issues of MICROSYSTEMS; Jan/Feb 1980 (Vol. 1, No. 1) to Mar/April 1981 (Vol. 2, No. 2). Issues must be in very fine condition. Will pay up to $3.00 per issue, including postage. Contact Samuel Vincent, Box 82, Hyde Park, PA 15641; or Call (412) 845-8613 after 7 P.M.

INDEXING AND MICROFORM AVAILABLE

This magazine is indexed in THE MAGAZINE INDEX available from Information Access Corporation, 404 Sixth Avenue, Menlo Park, California, and in microform from Xerox University Microfilms, Ann Arbor, Michigan and Bell and Howell, Micro Photo Division, Wooster, Ohio.

ADVERTISERS INDEX





CompuPro's System 68K...

Because Computers Shouldn't Blink.

Nothing is more aggravating to high-level computer users than a system that takes a little shut-eye every time they get on the bus. System 68K utilizes **CompuPro's** swift, sure CPU 68K, the jet-speed choice for scientists, industrial integrators and software developers. Consider the memory: 256K of 16-bit RAM, 2.4 megabytes of disk storage and **CompuPro's** M-Drive/H™ for 1.5 megabytes of ultra-fast RAM-based disk memory. Consider the ports: Four serial ports, one Centronics printer port and one parallel port. Consider the software: CPM-68K, mapForth and C languages. And now consider the price: **$8,995.**

RAM 22...

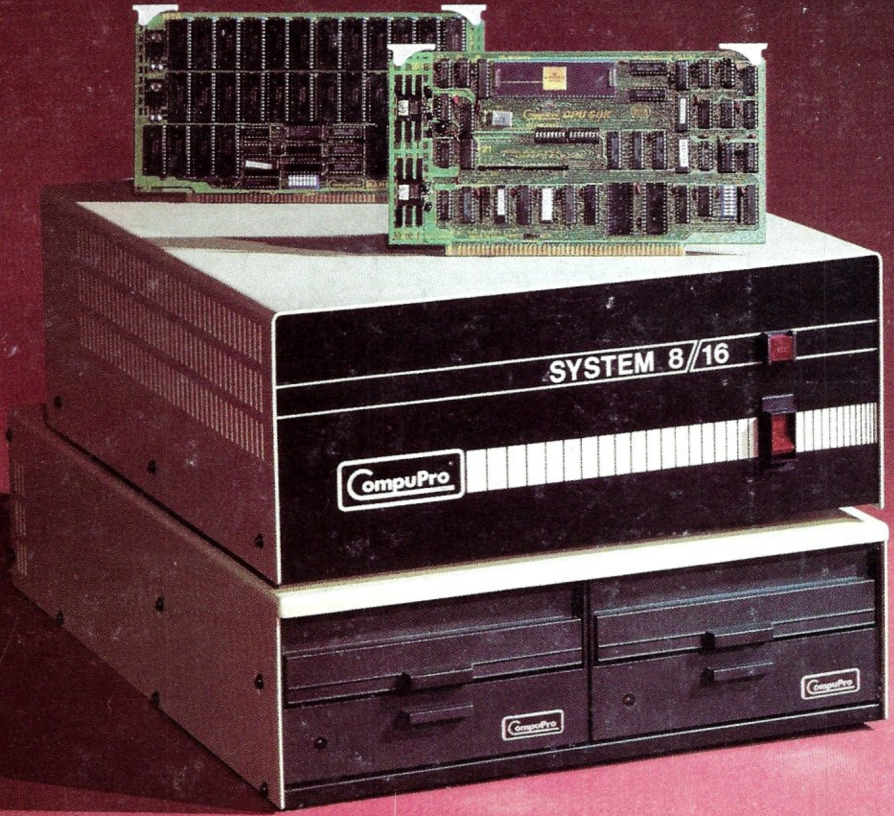
Because Memories Shouldn't Bloat.

If your random access memory is acting sluggish and overfed, **CompuPro's** RAM 22 provides a warehouse of extra space. You get an extra 256K x 8 (or 128K x 16) of memory, on a state-of-the-art board that operates up to 12 MHz. RAM 22 draws half the power of dynamic equivalents, but offers the speed and reliability of fully static operation. **$2,495** (ASM), **$2,695** (CSC).

CPU 68K...

Because Processors Shouldn't Blunder.

Ever have the feeling your commands are in the hands of a blundering gnome? **CompuPro's** indomitable CPU 68K brings state-of-the-art efficiency to your programming tasks. Operates up to 10 MHz, and works with both 8- and 16-bit memory. **$695** (ASM), **$850** (CSC).



CompuPro products are compatible with all S-100/IEEE 696 hardware. For performance, quality and reliability, contact your nearest **Full Service CompuPro System Center** today; call (415) 562-0636 for location.

CompuPro®

CompuPro, a Godbout company, Box 2355, Oakland Airport, CA 94614.

CIRCLE 38 ON READER SERVICE CARD