# SUPERLETTER

Serving SuperBrain and CompuStar Users Around The World

Oct. / Nov. Vol. 3 No. 5

uperletter's offices will be closed from Novembers 7th through December 27th due to international travel.

No news yet from Intertec regarding their new product line. Early reports from good sources speak about an 8/16 machine with a detached keyboard that will run as a multiuser system with CP/M 80, CP/86 and MP/M. Still no confirmation the new products will be able to use IBM-compatible software.

Rumors lately about Intertec's financial health have not been encouraging. Yet we also get conflicting information from certain dealers and those close to the company that Intertec is here to stay and any news otherwise is false.

We're hearing grumblings about the video portions of the SuperBrain II's which are going down due to weak solder and contact points on the *underside* portion of thevideo boards. Actually, it's a comparatively simple problem to fix, but if you don't know what to look for, it can be a headache.

The life of Superletter is dependent on a fragile connection between thousands of Intertec users, the factory and us. In order to have a strong end-user network, we need your continued support. Subscriptions and ad dollars help dealers, vendors and us maintain a presence in the marketplace.

As other computer clubs fold or gently disappear, we move forward. As of now, we are one of the longest-lived, if not *the* oldest, professionally published microcomputer newsletter devoted to a single computer system.

Our aim is to always be here for your needs, regardless of what happens in the rapidly changing computer landscape.

Have a Happy Holiday Season!

Albert Abrams
EDITOR

### Technical Corner

Floppy Disk Motor Shutoff, Extended Storage Capacity, and Variable Seek Rates for the CMC Targa BIOS.

James N. Herron, Ph.D.
University of Utah
Dept. of Biochemistry
410 Chipeta Way
Salt Lake City, UT 84108

With the addition of a CMC Targa 10 MB disk drive to my SuperBrain, I was distressed to learn that the CMC PROM did not support the extended floppy disk features which had been provided by Information Engineering's SB/E PROM. Most of my diskettes were formatted at 40 tracks per side, and a 40 TPS operating system was needed in order to dump them onto the hard disk. Inspection of the CMC BIOS revealed provisions for several different disk formats. Unfortunately, CMC did not follow through and implement these formats at the PROM level. Additionally, with the hard disk in operation, the floppy disks sit idle most of the time, and it would be quite advantageous to turn them off between their infrequent operations.

The solution to these problems is to use a different PROM, one which knows about extended floppy disk functions. A "quick fix", which works with the SB/E PROM is provided though use of CMC's BOOTHD utility. BOOTHD is a program which runs in the CP/M transient program area (TPA) and loads the CMC operating system from the hard disk's system tracks. First, boot the SuperBrain with SB/E and optionally use ACTIVATE to establish the desired floopy disk parameters. This will modify two addresses in CPU2's RAM, which control disk motor shutoff, seek rate, and tracks

per side. Then run BOOTHD, which will overlay SB/E with the CMC operating system. Because the control bytes are in CPU2's address space, they aren't modified by the CMC BIOS. This procedure may also work with the SOS PROM, but I've never had an opportunity to try it.

A more elegant solution involves modification of the CMC BIOS so that it initializes the floppy disk control bytes in CPU2's RAM during a restart operation. The following patch for CMC BIOS was designed for the SB/E EPROM. It's build around two of the Intertec DOS 3.2 floppy disk subroutines, which swap CPU2's RAM in and out of CPU1's address space. Information Engineering placed the two disk control bytes starting at location 8800H in CPU2's RAM. Location 8800H controls disk motor shut off and seek rate, while location 8801H contains the number of tracks per side. The low order nibble of 8800H controls the seek rate, with values of 0, 1, 2, and 3 selecting respective seek rates of 30ms, 20ms, 12ms, and 6ms. The high order nibble controls disk motor shutoff, with a value of 8 activating disk motor shutoff, and a zero value deactivating it. For example, storing 83H at location 8800H selects disk motor shutoff and 6ms seek rate.

I located the patch in GOCPM, immediately after CALL TRANS, which initializes floppy disk control bytes during a cold start. It's also possible to put the patch at the beginning of WMRET, in which case the control bytes are initialized during a cold start and updated with every warm start. Values to be stored at locations 8800H and 8801H can be added to the end of the Intertec configuration area to allow software modification of floppy disk characteristics. When you add the patch to CMC BIOS, make sure that the HFIRST equate is set to TRUE, which selects the hard disk as disk A. Also set an equate for the desired floppy

Continued on Page 2

Superletter is published, bi-monthly, by Abrams Creative Services. Mailing address: P.O. Box 3121, Beverly Hills, CA 90212. Phone: (213) 277-2410. Copyright<sup>©</sup> 1983 by Abrams Creative Services. All Rights Reserved. No part of this newsletter can be copied by any means without prior written permission from the publisher. Editor: Albert Abrams. Advertising & Subscription: Linda Patchell. (SuperBrain® and CompuStar® are registered trademarks of Intertec Data Systems, Inc. CP/M® is a registered trademark of Digital Research, Inc.)

### **Technical Corner**

Continued from Page 1

disk format (35/40/70/80 tracks per side, single/double sided). Assemble the patched BIOS and incorporate it into CPMH10S. COM (SPMH10D. COM for a double sided system) using DDT. Run CPMH10S and write a copy of the new operating system onto both the hard disk and a floppy diskette. Unfortunately the SB/E PROM doesn't know that the hard disk is there, so you'll have to start the system by placing the floppy that you SYSGENed with CPMH10S, in drive A and pressing both reset buttons. This procedure boots the CMC operating system from the floppy disk, but all subsequent warm starts default to the hard disk. A better solution would be to program a new PROM which contained both the hard disk boot loader and the extended floppy disk driver routines (a hint to reader with an EPROM blaster).

> CMC BIOS Patch for Disk Motor Shutoff Variable Seek Rate and 40 Tracks per Side

Dr. James N. Herron

University of Utah, Biochemistry 410 Chipeta Way Salt Lake City, UT 84108

; Equates
;
; Equates
;
; Equates
;
; DMA1 EQU OEB99H ;Select CPU-2 RAM
DMA2 EQU OEBASH ;Select CPU-1 RAM
SETFD EQU 08800H ;Bottom of control byte area
;
; Configuration Area
;
; Note:add these bytes to the end of the
; Intertec Configuration Area (location OEF20H)

DB 83H ;Disk motor shutoff, 6ms seek rate
DB 28H ;40 tracks per side
;
; Code Segment
;

CALL DMA1 ;Select CFU-2 RAM
LHLD INTED;Load HL with control bytes
SHLD SETFD;Store them at 8800H
CALL DMA2 ;Restore CFU-1 RAM
;



### -New Products-

### Seek and Search

Signature Software 841 Sheldon El Segundo, CA 90245 (213) 640-8189 Contact: Bill Dotson

Seek and Search provides extensive diskette editing facilities, as well as capabilities for restoring erased files, renaming files, and numerous other facilities for both the novice and experienced CP/M user.

Seek and Search provides the user a convenient way to maintain CP/M diskettes. This includes an extensive set of diskette editing and patching capabilities, such as:

- Restoring erased files
- Repairing damaged files (limited to the remaining readable sectors)
- Appending files
- Locating file positions on a diskette
- Locating specific data within a file
- Moving data within a file (limited to one sector minimum)
- Searching forward or backward for specific information
- Directing data to a printer (CP/M list device)
- Moving data or files from drive to drive or from one location to another within a single drive
- Mapping the remaining free space on diskette
- Displaying the directories of all drives and user groups
- Providing an editing facility in either Hexadecimal or Ascii mode
- Providing a limited ability to backup otherwise uncopyable diskettes

A hard-disk version is in the works and their newest update features automated controls. Price: \$69.95

#### The Ultimate

Computer Creations 766 El Camino Real,#D San Carlos, CA 94070 (415) 595-4466 Contact: Bob Bosone

"THE ULTIMATE" is a 4 in 1 program; it provides you with a "WORD PROCESSOR"; a "MINI DICTIONARY"; a "DATA BASE MANAGER"; and a "MAIL MERGER" program.

"THE ULTIMATE'S" 4 sub-programs are tied together in such a way that you can move smoothly from one to the other in an efficient manner.

- 1. "Word Processor" types letters, memos, reports, schedules, forms of all types, apartment inventories, customer lists, recipe lists, telephone lists, address lists.
- 2. All may be checked for spelling errors by the "Mini Dictionary", which includes over 12,000 words. This subprogram allows you to add new words and/or delete old ones which then gives you a customized dictionary just for your needs.
- 3. The "Data Base Manager" subprogram is used to file away your completed forms, lists, inventories, etc. All of these forms and/or list formats are generated on the "Word Processor" then the specific one you wish to use is called up in the "Data Base Manager" program so that you may fill in the blanks of each form or list. Once filled in they become your neatly filed away records that can be called back at any time for use with any of the other sub-programs, or for you to get a print out of all or any specific records you need.
- 4. "Mail Merger" pulls out specific information from your records in the "Data Base Manager" area and automatically inserts it into another document which you just produced on the "Word Processor". An example of this would be a form letter to club members or clients, or Board members, etc. "Mail Merger" neatly types in the required information such as name and/or address to create what seems like a personal letter.

Available for both the CompuStar and SuperBrain, I and II.

#### **RS-232 Breakout Box**

Remark Datacom, Inc. 4 Sycamore Drive Woodbury, NY 11707 (516) 367-3806

Contact: Ms. Blossom Kramer

The "poorman's" breakout box has been made available by REMARK DATACOM INC. The Model 51 MINI-PATCH BOX provides all users of devices employing RS-232 interfaces with the ability to reconfigure the connections, or signal paths. This is usually necessary when interfacing new devices such as printers and CRT terminals.

The Model 51 is equipped with a male and female, DB-25, 25 pin connector. This allows the placement of the MINI-PATCH BOX in series with any RS-232 cable path. Of the 25 pins, pin 1 (Frame Ground) and pin 7 (Signal Ground) are carried thru from the male to the female connector. The remaining 23 signals from each connector terminate on female jack receptacles. Sup-

Continued on Page 7

### Technical Corner II

### FLOPPY DRIVE MOTOR SHUT-OFF for any SUPERBRAIN or COMPUSTAR

by JONATHAN PLATT Copyright 1983

I have been accumulating information on the subject of motor shut-off for some time. It is at a point now where I think I can present it to the readers of Superletter with some authority and detail. It has been my tradition to provide not only the how-to but the how-come, the background and sometimes theory as well. No sense in breaking tradition.

Several methods of implementing a motor shut-off feature on the SuperBrain or CompuStar have been employed thus far. IE was the first with their SB/E BIOS and EPROM. They use drive select D to signal whether the motors are on or off. Of course, a hardware modification and their special EPROM is required. SB/E allows only two drives on a system. Both drives run when the motor signal is activated. The motors are turned off after ten seconds of idle use and there is a delay of one second when reactivated to let them come up to speed.

Quite some time later, Intertec released Engineering Change Order (ECO) number E030038. Part to the method covered with this ECO is a bit crude because it asks you to cut a trace on each floppy drive board. In addition to isolating each drive from the motor-off signal, one jumper must be installed on the drive's shunt block socket for all but CDC drives. This jumper's function is to use the drive select signal to also turn on the selected drive's motor. In this manner only the drive being used has its motor running, dramatically increasing every drive's lifetime. It is also much easier on those frequently flaky power supplies. The motor is turned off after three seconds of idle use and there is (or should be) a delay of 0.25 seconds when reactivated to let it come up to speed.

I have supplied my customers with instructions on how to implement motor shut-off in a less destructive manner. I advised them to cut the one conductor which carries the motor-off signal from the main board to the floppy drives under the assumption that a ribbon cable makes for a cheaper cutting board than a floppy drive. Well, this worked with most drives but not Shugarts. The reason was that even though the conductor was cut, the Shugart drives did not keep the signal isolated from the back section of cable (on the other side of the cut.) Thus,

more than one drive would be selected at once. Other drives did not have this problem.

One of my customers offered what I think is the best solution which is completely reversible if you decide that motor shut-off is not what you need. This is a streamlined version of my old method and is still functionally identical to the way Intertec implements it. But before I tell you what to do, let me tell you what it does.

#### TECHNICAL BACKGROUND

The boot EPROM initiates a drive operation by sending a command to the Floppy Drive Controller (FDC). When the FDC receives the command it activates a signal called Head Load (HLD). The HLD signal is the input to what is called a "one-shot". A one-shot basically delays a signal for a certain time set in hardware with a resistor and a capacitor. Intertec set it for 121.5 millseconds delay. After the delay the one-shot outputs the HLD signal back to an FDC input called Head Load Timing (HLT). Only after HLT becomes active will the FDC continue with the disk operation specified in the EPROM'S command.

Under Intertec's drive logic, a drive will only be selected when both the EPROM has the drive selected and HLD is active. So after one of the drives is selected by the EPROM and the FDC receives a command, HLD is activated. This sends a select signal to the selected drive. The select signal activates the drive and with the modifications it also turns the motor on.

Meanwhile, the one-shot is delaying HLD back to the HLT signal, giving the drive motor time to come up to speed before the FDC continues with the disk operation. Drive manufacturers specify a maximum time of 250 milliseconds (one quarter second) before the drive motor will be at its proper speed.

Once a disk operation had been completed, the FDC will count 15 revolutions of the disk and then turn off its HLD signal. Since the drive motors turn at 300 RPM, it works out to three seconds. When the HLD signal goes inactive, the drive is deselected and the motor turns off.

When Intertec designed their one-shot delay, they made it for 121.5 milliseconds. They were only thinking of the time it would take for the read/write head to load against

Continued on Page 4



### PRICE, SERVICE, AND MORE

- SuperBrains™ and CompuStars™ at competitive prices
- East Coast Service Center
- Mass storage for SuperBrain on Corvus™ hard disk
- Data media conversion (9-Track or 8" disk to SuperBrain or vice versa)
- Victor 9000™ we can make Intertec™ software run on the Victor 9000™

### CALL TO ORDER: 1-800-257-5217 (in NJ 1-609-424-4700)

TriStar Data Systems
2 Keystone Avenue, Cherry Hill, NJ 08003

SuperBrain, CompuStar, Intertec are trademarks of Intertec Data Systems Corvus trademark of Corvus Systems Victor 9000 trademark of Victor Technologies, Inc.

#### Technical Corner II

Continued from Page 3

the diskette surface. Even this was a little long since 30 milliseconds is the standard for mini-floppies. This was long before they even heard about turning motors off. But even after Intertec finally learned how to accomplish motor shut-off, they still shipped their machines with the 121.5 millisecond delay. I believe they later released an ECO with the proper resistor value change to bring the delay to about 250 milliseconds.

The main effect of not waiting enough time for the drive to come up to speed would be CRC errors. This was not only a symptom of the wrong hardware delay but also some deficient programming in the EPROM. I received Intertec EPROM version 4.2 with my CompuStar 30, the equivalent of a QD SB II, last December. I studied their programming closely. The machine came with motor shut-off already installed but still had the wrong resistor value in the delay circuit.

If there has been no new EPROM released since 4.2, then the bugs are still there as well. First, the internal programmed delay for drive speed is about 138 milliseconds instead of 250. Second, when the same drive is accessed after the three second idle shut-off time, the EPROM does not delay at all. Again, there should be a delay of 250 milliseconds. Luckily, they did think of delaying when going from one drive to another even though the three second time out has not occurred. This must be done since each drive runs independently. Think of how PIP goes back and forth between drives.

Any SuperBrain or CompuStar ever made will support motor shut-off. But if you are using an Intertec boot EPROM then you may have to make one hardware modification. The time allowed for the drive motor to come up to speed is crucial. The hardware delay problem can be remedied by replacing a resistor. The problem with the Intertec EPROM can only be fixed by buying a better EPROM unless they have fixed theirs by now. At the risk of sounding unhumble, the SOS BIOS EPROM handles all timing properly even without the aid of the one-shot. So any hardware change would not be necessary with this EPROM.

If you are using the SB/E BIOS motor-off feature, either stick with what you have or undo the damage your hardware modifications did and use the SOS EPROM. Intertec's EPROM will not respond to disk features of the SB/E BIOS.

#### MODIFICATIONS

If you are implementing motor shut-off with the Interrtec EPROM, you will need to do the board modification. If you are using the SOS EPROM skip the board modification. It involves removing a resistor from the PC board and replacing it with a new one of higher value. Drive modifications should be followed under all circumstances. If you use any other EPROM, check with the manufacturer before continuing.

If you are not certain about any of the terms used in the following directions then you would be better off by having a technician do the modifications for you.

#### **BOARD MODIFICATION**

First take the cover off and locate the floppy drive cable connector on the upper right edge of the main board. To the left of the connector are two columns of chips. Counting from the top edge downward, the resistor is between the second and third chips of those two columns. Just below the resistor there should be a capacitor between the third chips of those two columns. The resistor is color coded with a red, violet, orange and gold band in that order. It is also known as a 27 K Ohm resistor.

Remove the resistor and put a 56K Ohm resistor in its place. This involves soldering, so if you do not know what you are doing let a technician do it for you. A 56K Ohm resistor is color coded with a green, blue, orange and gold band in that order. The gold band means that it has a five percent tolerance.

This hardware change will increase the delay before a disk operation to 252 milliseconds. Now the drives will have enough time to come up to speed before trying to attempt any disk access.

### DRIVE MODIFICATIONS

Take the cover off and remove the disk drives. Cut a piece of heavy adhesive tape about one inch long and one eigth inch wide for each drive. I usually use clear stranded packing tape. The tape will cover a conductive connector strip on each drive. On a table, orient the drives as they would normally stand in the computer. Find the eigth gold connector strip up, counting from the bottom strip. Cover this strip on both sides of the connector tab.

If you have CDC drives in your computer, the drive modifications are complete. If you have Tandon, Shugart or other similar drives, continue by locating the shunt block socket on your drives. They usually have the break-away tabs plugged into them. If the sockets are labeled they will have DS0, DS1, DS2, DS3, MX, HM and HS printed on the board beside the socket. Jumper both HS and HM. HS should already have

been jumpered anyway. The drive modifications are complete. Put the connectors back onto the drives making sure the tape stays in place. Put the drives back where they belong, replace the cover and test the system.

If the signals are not labeled, the best I can do is give you examples. On the Tandon drives which have a sixteen pin socket, the additional jumper goes from pin eight across to pin nine. On the Shugart drives which have a fourteen pin socket, the additional jumper goes from pin seven across to pin eight.

Have a Super day!



### Super Upgrade

### Upgrade your SUPERBRAIN QD or JR to SUPERDENSITY

- 1.5 M formated capability
- Increase QD capacity by more than 200%
- Increase JR capacity by more than 400%
- Compatability to existing software
- Includes Jon Platt's SOS & PROM
- Up to 4 drives of any SuperBrain format
- Hard disks available also

From \$850 installed

For Information Call or Write MICROHELP! Computers 7 W. Acacia Suite 5 Stockton, CA 95202 (209) 466-6187

★ 10 Free 96TPI DS Disks ★

### *A*BRAMS

### CREATIVE SERVICES CP/M SOFTWARE DISCOUNTS!

dBASE II \$489! WordStar \$299! SuperCalc \$249! Spellbinder \$345! CBASIC-2 \$96! SpellGuard \$199!

Welcome to the world of super low software prices! The lowest prices anywhere on nationally advertised CP/M software. For any computer and any format, including hard-to-find SuperBrain, CompuStar, Osborne, CPM/86 and IBM-PC formats. You can even use your VISA or MasterCard to pay for it. The only thing we won't discount is quality. Every software package you buy is guaranteed to be factory-fresh, pretested and shipped to you quickly and carefully. We're adding to our large list of products each day, so if you don't see what you've been looking for, call us. Chances are we've already begun to sell it at a lower than discount price.

	Your		Your		Your
Manufacturer / Product	Price	Manufacturer / Product	Price	Manufacturer / Product	Price
AMERICAN TRAINING INT'L		Pascal BZ (Business version)	\$399	STAR COMPUTER SYSTEMS	
All ATI-Power Packages	\$69	Pascal Z and BZ Combo	\$599	Call for discount prices on all modules.	
Medical	\$820	T/Maker	\$299	STATCOM	
Dental	\$820	METASOFT	\$299	CRT Form	\$369
ASPEN SOFTWARE	•	The Benchmark	\$399	SUPERSOFT	
Grammatik	\$70	The Benchmark Mail List	\$219	Ada	\$263
Proofreader	\$50	MICRO BUSINESS APPLICATIONS	9219		
FRONTIER/ASYST DESIGN	•	Call for discounts on all modules.		Diagnostic II	\$85
SOFTWARE		MICROPRO PRODUCTS		Forth (Z80 or 8080)	\$149
Prof Time Accounting	\$525	WordStar	\$299	Fortran	\$267
Fixed Asset Management	\$325	Mail-Merge	\$199	Fortran w/Ratfor	\$337
General Subroutine	\$262	WordStar/Mail-Merge	\$475	Tiny Pascal	\$84
Application Utilities	\$429	DataStar	\$269	Disk Doctor	\$84
BYROM SOFTWARE	4425	WordMaster	\$130	Utilities I	\$54
BSTAM	\$145	SuperSort I	\$220	Utilities II	\$54
BSTMS	\$145	SpellStar	\$220	Nemesis	\$40
CAXTON	\$173	CalcStar	\$175	Dungeon Master	\$40
"Cardbox"	\$199	InfoStar	\$399	Analiza II	\$45
CHANG LABS	\$199	MICROSOFT PRODUCTS	\$399	Term II	\$165
Microplan	\$349	Microsoft Basic 80	\$285		\$249
DESIGNER SOFTWARE	<b>434</b> 3	Microsoft Basic Compiler	\$324	Scratch Pad	
Palantir Word Processor	\$375	Fortan 80	\$345	Stats Graph	\$174
DICTRONICS	93/3	Cobol 80	\$569	Optimizer	\$174
Random House Thesaurus	\$139	Macro 80	\$144	TARGET	
DIGITAL RESEARCH PRODUCTS	\$135	Mu Math/Mu Simp	\$219	PlannerCalc	\$99
PL/1-80	\$450	Mu Lisp/Mu Star	\$170	MasterPlanner	\$290
MAC	\$83	M-Sort		A POTPOURRI OF EXTRAS	
TEX		Edit 80	\$170	Plan 80(Financial Modeling)	\$447
	\$88	Multi-Plan	\$100	Access 80, Level I	\$241
DeSpool	\$48	MICROSTUF	\$229	Access 80, Level II	\$400
CB-80	\$175	Crosstalk	\$149	Statpak	\$429
	\$450	NEXUS	Ψιισ		\$199
Pascal MT +	\$419	Zip (M Basic or C Basic)	\$149	Lynx	\$319
	\$305	OASIS	ΨΙΤΟ	Quic-N-Easi	
Pascal SPP Speed Program	\$175	The Word Plus	\$140	NAMOR (Mail List Manager)	\$150
Access Manager	\$279	ORGANIC SOFTWARE	\$ 14U	Syntax CP/M Self-Tutor	\$120
	****	Textwriter III	\$109	Catalog	\$75
FMS-80	\$850	Datebook II	\$279	Citation	\$235
FMS-80-1	\$395	Milestone	*	CP+	\$145
			\$279	Move-It	\$119
Call for discounts on all modules. ECOSOFT		PEACHTREE SOFTWARE		The Protector	\$159
	***	Call for discounts on all modules.		Autocode	\$199
Microstat	\$219	PEARL SOFTWARE	***	Ten OPUS 5¼"SSDD Diskettes	\$24
EPIC COMPUTER	***	Personal Pearl	\$199	Ten 5¼" DSDD Quad Diskettes	\$34
Supervyz	\$99	PRO/TEM SOFTWARE		10 MB Hard Disk	\$2499
FBN		Notebook	\$150		\$3099
SmartKey	\$59	Synopsis	\$120	20 MB Hard Disk	
SmartPrint	\$33	Footnote	\$120	1200 Baud Auto Answer Modem	\$539
FOX AND CELLER		SELECT SOFTWARE		300 Baud Auto Answer Modem	\$269
d∪til	\$85	Select w/SuperSpell	\$379	Daisywriter 2000 Printer	\$1295
Quikcode	\$249	SORCIM		Sheet Feeder	\$850
INTERNATIONAL SOFTWARE		Pascal/M	\$349	48K Buffer for Daisywriter	\$129
		B1() 4 (000C)	\$425	I DI . W. COC T. b I DIOC	\$190
Mathe-Magic	\$95	Pascal/M (8086)	<b>744</b> 3	Jon Platt's SOS Enhanced BIOS	3130
	\$95	SuperWriter	\$345	Com-Stute SB   Graphics Package	\$165

Please add \$4.50 per product for shipping and handling. VISA, MasterCard, and money orders are accepted. Checks OK but require 10 working days to clear. International orders: please add \$10 and provide for additional postage. All checks must be drawn in U.S. dollars from American banks. California residents: add 6½% sales tax to total. Yes, we accept purchase-orders from corporations, educational institutions and government agencies.

To order, call or write:

**ABRAMS CREATIVE SERVICES** 

369 South Crescent Drive Beverly Hills, California 90212 (213) 277-2410





### -Programs-

These programs offer a good way to move files between CP/M machines, even with toatlly different disk systems. The only requirement is that each have a serial port. Transfer can be made with either a direct interface cable or a pair of modems. The programs are very tolerant of timing and contain the facility for retransmitting records containing transmission errors.

Listing 1
Program to transmit a CP/M file

;	Program	to transmit a C	P/M file through a serial port	·
;	Accesse Assumes	s port directly, an 8-bit word l	bypassing the RFOS ength and no parity	
1	Vorcion	of 9/2/83		
;	version	01 9/2/03		
; BCOT	EQU	6000H	:CP/M reboot address	
BDOS	EQU	0605H	;CP/M BDGS entry point	
TFCB SIOSTA	EQU EQU	005СН 07Н	;FCB for file to be transmitted ;SIC status port	
SIODAT	EQU	06H	:SIO data port	
IFLAG OFLAG	EQU EQU	62H 61H	;Input flag for serial port ;Cutput flag for serial port	
BUFREC	EQU	128	;Buffer size (CP/M records)	
;				
1 '	URC	166H		
; ;	lf requ	ired, place seri	al port initialization code here	
;				
TX:	LDA CPI	TFCB+1	:Check for filename	
4	JNZ	OPEN		
ABORT:	LX1 MV1	D, FNMER C, 9	;Print error message and reboot	
	CALL	BDOS		
1	MVI CALL	E,04H XMTBYT	;Send LOT character	
	JMP	BOOT	;Return to CP/M	
OPEN:	LXI MVI	D,TFCB C,15		
	CALL	BDCS	Open file	
1	INR JNZ	A FOUND	;File present	
Ē	LXI	D, FNFER	;Point at error message	
FOUND:	JMP XRA	ABGRT A	Print and reboot	
	STA	TFCB+32	:Set next record byte to zero	
REACY:	MOV PUSH	C.A	;Set record count in buffer to zero ;Save record count	
•	MVI	C, 11		
j	CALL POP	BDOS B	;Get console status	
l	URA	λ	No. hour annual of	
	JZ PUSH	REALY I B	;No key pressed	
1	MV1	C, I		
	CALL POP	BDOS B		
l	CP1	Ø3H READY I		
ENDXMT:	JNZ LXI	D, ECTMSG	;Print ECT message and abort	
DIADVI	JMP	ABORT		
READY1:		S10STA 1FLAG	;Read serial status port	i
	JZ JZ	READY	;Wait for character to be received	
	IN	SIODAT	Get received character; Mask off bit 7 for ASC11 codes	
	ANI CPI	7FH 04H		
	JZ	ENDXMT	;End transmission	
	CPI JZ	Ø1H XMTREC	Transmit next record	
	CPI	02H	;Ignore other characters	
	JNZ LHLE	READY DATPIR	rignore other characters	i
F .	LXI	D,-0080H	;Point at last record	
1	SHLD	DATPTR		
VMm DEG	INR	C ·	;Go back one record	
XMTREC:	STA	RPTFLG	;Save repeat flag	
XMTRC1:	CZ	COUNT	;Increase record message if new record ;Decrease record count	
AMTRULE	JP	XMTRC 2	:More in buffer	
	LDA ORA	EOFFLG A		
I	JZ	READFL	:More in file	
	LXI	L, EOFMSG	· ·	
1	MVI CALL	C,9 Bros	Print ECF message	
I	MV1	E,03H		
•	CALL JMP	XM1BYT BOOT	;Send LOF byte	
READFL:	: LX1	H, DATBUF	. Brink of Brainsin of St. 1966	
1	SHLD	DATPTR	;Point at beginning of data buffer	
READ:	INR PUSH	C B	;Initialize record count to zero	
KEMU:	PUSH	D.		
I	MVI CALL	C,26 BDOS	·Set DMA address	
I	LXI	D, TFCB	;Set DMA address	
	MVI CALL	C,20 BDOS	Read next record	1
1	POP	D	, next record	
	POP	B A		
	JNZ	NOMORE	;EOF detected	
1	DAD DAD	н <b>, 00</b> 8 <b>0</b> н С	;Point at next record in buffer	
I	XCHG			
1	INR MVI	C A,BUFREC	;Increase record count	
	CMP	C	Chan at and at aut:	
ľ	JNZ JMP	READ XMTRC1	;Stop at end of extent	
			cont	inued

#### Listing 1, continued

List	ting 1, con	ntinued		
T	NOMORE:	MV1	À, 1	
1		STA	EOFFLG	;Set ECF flag
1		JMP	XMTRC1	
	XMTRC2:		RPTFLG	
ı		INR	A	
		MOV	E, A	
		CALL	XMTBYT	;Send response to request for record ;Point at next record to transmit
•		LHLD MV1	DATPTR B,80H	;Character count for record
1		MVI	D, 0	;Checksum for record
	XMTRC3:		A, M	;Get next byte to send
		INX	Н	;Update pointer
1		MOV	E, A	;Save in E
ı		ADD	D	
ł		MCV	D, A	;Update checksum
1		DCR	В	
ı		JNZ	XMTRC3	;More in record
ı		MOV	E,D	
1		PUSH	XMTBYT	;Send checksum
		MVI	B	
i i		MV1	C,4 E,00H	
,	XMTRC4:		XMTBYT	;Send 4 NULs to replace any missed bytes
		DCR	C	
		JNZ	XMTRC4	
ł		SHLD	DATPTR	;Save pointer to next record
1		LX1	D, RECMSG	
		MV1	C, 9	;Print record message
		CALL LDA	BDOS RPTFLG	FITTH Tecord message
1		ORA	A	
1		JZ	NEXTLN	;Skip 'again' if successful first time
		LXI	D, AGAIN	, okip again ii oaccessiai iiise cime
1		MVI	C, 9	
ı		CALL	BDOS	;Print 'again' if record repeated
,	NEXTLN:		D, CRLF	
1		MVI	C,9	
1		CALL POP	BDOS R	;Print CRLF
		JMP	READY	;Go wait for next prompt to send
	COUNT:	LXI	H, RECCNT+4	, to wait for hear prompt to only
•	COUNT1:	INR	м	
1		MVI	A, '9'	
ı		CMP RNC	М	;Over 9?
L		MVI	м.'0'	
1		DCX	н	
I		MOV	A, M	
1		CPI		
ı		JNZ	COUNT1	
		MVI	M, 'Ø'	But A to .
•	XMTBYT:	JMP IN	COUNT1 SIOSTA	;Put 0 in message
	ANIBIT:	ANI	OFLAG	:Transmit character through serial port
		JZ	XMTBYT	
		MOV	A, E	
		OUT	SIODAT	
		RET		
	NMER:	DB	'File name missi	ing' ODH OAH 'S'
	FNFER:	DB .	'File not found'	'. ODH. OAH. 'S'
	EOFMSG:		'Transfer comple	ete',0DH,0AH,07H,'\$'
	EOTMSG:		'Transfer termir	nated', 6DH, 0AH, 07H, '\$'
	RECMSG:		'Record #'	
1	RECCNT:		. 0.	:Record # transmitted
1 .	ACA IN.	DB DB	'transmitted\$'	*Beneated vocavd
	AGAIN: CRLF:	DB	'again\$' 0DH,0AH,'\$'	;Repeated record ;CRLF sequence
	DATPTR:		DATBUF	;Pointer to next data byte to send
	EOFFLG:		0	;Flag for EOF read
1	RPTFLG:	DB	ø	;Flag for repeated record
	DATBUF	EQU	\$	;Data buffer
	;	END		•

### Listing 2 Program to receive a CP/M file

;	Access	es port direct	a CP/M file through a serial port tly, bypassing the BDOS rd length and no parity	
;	Versio	n of 9/17/83		
; BOOT	EOU	øøøøн	:CP/M reboot address	
BDOS	EOU	0005H	;CP/M BDOS entry point	
TFCB	EOU	005CH	;FCB for file to be transmitted	
SIOSTA		Ø7H	;SIO status port	
SIODAT		Ø6H	;SIO data port	
IFLAG	EOU	Ø2H	;Input flag for serial port	
OFLAG	EOU	Ø1H	Output flag for serial port	
BUFREC		128	;Buffer size (CP/M records)	
RETRY		4	;Number of retries before quitting	
;	LQU	•	mamber of feeties before quiceling	
•	ORG	100H		
;				
;	If req	uired, place s	serial port initialization code here	
;				
RX:	LDA	TFCB+1		
	CPI	1, 1	;Check for filename	
	JN2	OPEN		
	LXI	D, FNMER	Print error message and reboot	
ABORT:		C,9		
	CALL	BDOS		
	MVI	E,04H	C A FOR -b	
	CALL	XMTBYT	;Send EOT character	
	JMP	BOOT	;Return to CP/M	1
OPEN:	LXI	D, TFCB		
	MV1	C,19	;Delete old file if present	
	CALL LXI	BDOS D.TFCB	;belete old lile it present	
	MVI	C. 22		
	CALL	BEOS	:Make new file	
	INR	A A	, make new lile	
	LXI	D. DDFER	;Point at error message	
	JZ	ABORT	:Print and reboot	
START:		C.30	:Send some NULs first	
JIANI.	MVI	E,00H	700110 001110 11000 111111	
NULS:		XMTBYT		
NOLD.	ECR	C		
	JNZ	NULS		
	IN	SIODAT	;Clear serial data port	
READY:	LDA	RPTCTR	, FT	
DI.	CPI	RETRY		
	MVI	E, 01H	cont	i

	JZ	REACY1	First time for this record
	INR	E	;Change request character to 02H
	ORA	Α	:Test repeat counter
	JNZ	READYl	;Try again i∳ not zero .
ENDXMT:		D, EOTMSG	Print EOT message and exit
	JMP	ABORT	
READY1:		XMTBYT	;Send prompting byte
READY2:	MVI CALL	C,11 BDOS	.0.1
	ORA	BLOS A	;Get console status
	JZ	READY3	;No key pressed
	MVI	C, 1	the help pressed
	CALL	BDOS	
	CPI	Ø3H	
	JZ	ENDXMT	;Terminate, if control-C pressed
READY3:		SIOSTA	;Read serial status port
	ANI	IFLAG	
	JZ	READY2	;Wait for character to be received
	IN ANI	SIODAT 7FH	Get received character
	CPI	03H	Mask off bit 7 for ASCII codes
	JZ	CLOSE	:Flush buffer and close file
	CPI	04H	,1100. Sollet did Close 1116
	JZ	ENDXMT	;End transmission
	CPI	01H	
	JZ	RCVREC	;Receive next record
	CPI	Ø2H	
	JNZ	REACY2	;Ignore other characters
RCVREC:		в,86н	;Byte count for record
	MVI	D,Ø DATPTR	;Initialize checksum
RCVRC1:		RCVBYT	Prepare to store data
NC VICI.	MOV	M,A	;Get a byte ;Store in data buffer
	INX	н	, Store in data buller
	ADD	D	
	MOV	D, A	;Update checksum
	DCR	В	•
	JNZ	RCVRC1	;Continue for 128 bytes
	CALL	RCVBYT	;Get checksum
	CMP	D	
	PUSH JNZ	PSW	;Save status
	SHLD	RCVRC2 DATPTR	;Bad read
RCVRC2:		D, RECMSG	;Save new record pointer
ne vice.	MVI	C, 9	
	CALL	BDOS	;Print received record message
	LDA	RPTCTR	
	CPI	RETRY	
	JZ	RCVRC 3	;First try
	LXI	D, AGAIN	
	MVI	C,9	
RCVRC3:	CALL	BDOS	;Print 'again'
ncvnc3:		D, CRLF	
	MVI CALL	C,9 BDOS	.Drint CRIP
	LXI	H, RPTCTR	;Print CRLF
	DCR	M M	;Decrement repeat counter
	POP	PSW	, seezament repeat counter
	JNZ	START	; successful read
	MVI	A, RETRY	, omenaccessiul read
	STA	RPTCTR	Donot country for
	LXI	H, RECCNT+4	Reset counter for next record
COUNT:	INR	M.	
	MVI	A, '9'	
	CMP	M	;Over 9?
	JNC	BUFCHK	• •
	MVI	м,'0'	•
	ECX MCV	H A, M	continue

CPT				
JNZ NN1 N, No JNN1 N, No JNN1 COUNT NN1 N, No JNN1 COUNT BLECH LATE COUNT LIKE LATE CATBUF BUFFREC*128) CC FLUSH (LATBUF BUFFREC*128) CC FLUSH (LATBUF BUFFREC*128) CC FLUSH (LATBUF BUFFREC*128) CC FLUSH (LATBUF BUFFREC*128) CNO A, F (CMP H COUNT BUFFREC*128) CNO A, FLUSH (CMP H COUNT BUFFREC*128) CNO CNO A, FLUSH (CMP H COUNT BUFFREC*128) CNO CNO BUFFREC COUNT BUFFREC*128 CNO CNO BUFFREC COUNT BUFFREC COUNT BUFFREC COUNT BUFFRECOUNT BUFFREC COUNT BUFFREC COUNT BUFFREC COUNT BUFFREC COUNT BUFFREC COUNT BUFFRE COUNT BUFFR		CP1		
UNP COUNT PUT 0 in message  LLTI LATE LATE LATE LATE LATE LATE LATE LATE	•	JNZ	COUNT	•
HUFCHA: LHIL		MV1	м,'0'	
LXI C. (LATBUP+BUPREC*128)  FAC L CC FLUSH ;Flush buffer if full JMP REACY ;Go look for next record FLUSH: LAI F.,DATBUP ;Start at beginning of buffer FLUSH: LAI F.,DATBUP ;Start at beginning of buffer FLUSH: LAI FLUTPR ;Compare to see if empty MCV A.F. CMP H JNZ FLUSH2 ;More to go MCV A.E. CMP L JNZ FLUSH2 LXI H.,DATBUP ;ShiLL DATPPR ;Reset data pointer RET FLUSH: PUSH C MVI C. 26 CALL BTOS ;Set FMA address LXI E.,TFCB MVI C.,21 CALL BTOS ;Write record POP D CRA A JZ FLUSH3 ;Good write LXI E.,DIFFER ;Flisk error FLUSH3: LXI H.,00860H TAI FLUSH3 ;Go write it if present FLUSH3: LXI H.,00860H TAI CHOS ;Close file LXI E.,DUFBS ;Good write LXI E.,DUFBS ;Flush ;Flush buffer of data LXI E.,DUFBS ;Close file LXI E.,DUF			COUNT	;Put 0 in message
TAC	BUFCHK:			
CC FLUSH ;Flush buffer if full JMP REALY ;Go look for next record FLUSH: LAL LAL F, LATBUP ;Start at beginning of buffer FLUSH: LAL LATPIK MCV A,I ;Compare to see if empty MCV A,E CMP H JNZ FLUSH2 ;More to go MCV A,E CMP L JNZ FLUSH2 LX1 H, LATBUP ;Start at beginning of buffer FLUSH: PUSH C ;Compare to see if empty MCV A,E CMP L JNZ FLUSH2 LX1 H, LATBUP ;More to go MCV A,E CMP L JNZ FLUSH2 LX1 H, LATPIR ;Reset data pointer RET FLUSH: PUSH C ;Compare to see if empty MCV A,E CMP L JNZ FLUSH2 LX1 H, LATPIR ;Reset data pointer RET FLUSH: PUSH C ;Compare to see if empty MCV A,E CMP L MCV A,E CMP L MCV C, 26 CALL BEOS ;Set FMA address LX1 E, TFCB MV1 C, 21 CALL BEOS ;Write record POP D C CALL BEOS ;Write record POP D C CALL BEOS ;Write record POP D C CALL LAT L,				EC*128)
JMP				
HILSH: LAIL LATPIK MCV A,I CMP H JNZ FLUSH2 ;More to go MCV A, k CMP L JNZ FLUSH2 LXI H, LATBUF SHILL BCS SHILL BCS CALL BCS CALL BCS CALL BCS LXI L, EFFER LXI L				
FLUSH): LHLL LATPTK  MCL A, I CMP H JNZ FLUSH2 LX1 H, LATBUF SHLL LATPTR R KET  FLUSH2: PUSH C CALL BIOS SET FMA address LX1 E, TFCB MV1 C, 26 CALL BIOS SET FMA address LX1 E, TFCB MV1 C, 21 CALL BIOS SET FMA ADDRESS LX1 E, TFCB MV1 C, 21 CALL BIOS SET FMA ADDRESS LX1 E, TFCB MV1 C, 21 CALL BIOS SET FMA ADDRESS LX1 E, TFCB MV1 C, 21 CALL BIOS SET FMA ADDRESS LX1 E, DEFFER SET SET FMA ADDRESS LX1 E, DEFFER SET				
MCV A, CMP H JNZ FLUSH2 ;More to go MCV A, E CMP L JNZ FLUSH2 LX1 H, LATBUF SHLL LATPTR ;Reset data pointer  KET FLUSH2: PUSH LJ L, TFCB MV1 C, 26 CALL BEOS ;Set FMA address LX1 L, TFCB MV1 C, 21 CALL BEOS ;Write record POP D CRA A JZ FLUSH3 ;Good write LX1 L, LFFER ;Fisk error JMP ABCRT FLUSH3: LX1 H, 00860H TAL LX1 L, LFFER ;Fisk error JMP FLUSH3 (Go write it if present FLUSH3: LX1 H, 00860H TAL LX1 L, LFFER ;Flush buffer of data LX1 L, LFFER ;Flush duffer of data LX1 L, LFFER ;Flush duffer of data				;Start at beginning of buffer
CMP	FLUSHI:			
JNZ	1			Compare to see if empty
MCV A, L CMP L JNZ FLUSH2 LX1 H, LATBUF SHLL LATTR R KHT  FLUSH2: PUSH T  CALL BIOS SET FMA address LX1 E, TFCB MV1 C, 26 CALL BIOS SWRITE record POP D CRA A JZ FLUSH3 SGOOD WRITE LX1 L, DLFFER STISK error JMP ABCRT  FLUSH3: LX1 H, ØB60H FAL D TAL D TA				Mana A
CMP				;more to go
JNZ FLUSH2 LX1 H, LATBUF SHLL LATTR ktT  FLUSH2: PUSH				•
LX1	•			
SHLL DATPTR REST data pointer RET  FLUSH2: PUSH				
Harmonia				·Reset data nointer
FLUSH2: PUSH			DATATA	, neset data pointer
MV1   C, 26	FLUSH2:		1E	
CALL LN1 C, TFCB MV1 C, 21 CALL BFOS ;Write record POP D CRA A JZ FLSH3 ;Good write LN1 L, DLFFR ;Fisk error JMP ABCRT FLUSH3; LX1 H, ØB6ØH FLUSH3; LX1 H, ØB6ØH TAL D T				
LX1				;Set DMA address
MV1 C,21 CALL BFOS ;Write record POP D CRA A JZ FLSH3 ;Good write LX1 L,DLFFR ;Fisk error JMP ABCRT FLUSH3 LX1 H,00500H FLUSH3 LX1 H,00500H TAL D TAL	•	LX1 .		
POP C CRA A JZ FLUSH3 ;Good write LX1 L, DLFFER ;Fisk error BCRT FLUSH3 LX1 H, ØB6ØH FLUSH3 LX1 H, ØB6ØH FLUSH3 LX1 H, ØB6ØH CLOSE: CALL FLUSH ;Flush buffer of data LX1 E, TFCB MV1 C, 16 CALL BLOS ;Close file LX1 L, LOFFG MV1 C, 9 CALL BLOS ;Print LOF message JMP HOOT ;Reboot RCVBYT IN SIOSTA ;Receive byte from serial port ANI IFLAG JZ KCVBYT IN SIODAT RET XMTBYT: IN SIOSTA ;Transmit byte to serial port OPLAG JZ XMTBYT XMTBYT: IN SIOSTA ;Transmit byte to serial port OPLAG JZ XMTBYT FNMER: DB 'File name missing', ØDH, ØAH, 'S' FOFMSG DB 'Fransfer complete', OPH, ØAH, OPH, 'S' ECOTMSG DB 'Transfer terminated', ØDH, ØAH, ØTH, 'S' ECOTMSG DB 'Transfer terminated', ØDH, ØAH, ØTH, 'S' RECCMS DB 'Record f' received' ' ' ' ' ' ' ' ' ' ' ' ' ' ' ' ' ' ' '		MV1		
CRA   A   J2	1		Bros	;Write record .
JZ FLUSH3 ;Good write LN1 LDIFFER ;Fisk error  ABCRT  FLUSH3: LN1 H,0060H  TAI			-	
LNI L, LLFER JMP ABCHT FLUSH3: LNI H,0060H INT IN				
JMP ABCHT FLUSH: LXI H,0860H  TAI L  XCHG JMP FLUSH: ;Go write it if present CLOSE: CALL FLUSH: ;Flush buffer of data LXI L,17CB MVI C,16 CALL BLUSS ;Close file LXI L,EURNSG MVI C,9 CALL BLUSS ;Print EOF message JMP MOOT ;Reboot RCVBYT: IN SIOSTA ;Receive byte from serial port ANI FLAG JZ RCVBYT IN SIODAT RET  XMTBYT: IN SIOSTA ;Transmit byte to serial port ANI OPLAG JZ WATHEYT MOV A,E OUT SIODAT RET  ;FMMER: DB 'File name missing',0DH,0AH,'\$' DDFER: DB 'Disk or directory full',0DH,0AH,'\$' RCVBYSG DB 'Transfer terminated',0DH,0AH,07H,'\$' RECMSG DB 'Transfer complete', ODH,0AH,07H,'\$' RECCOTS DB 'Transfer terminated',0DH,0AH,07H,'\$' RECCOTS DB 'Record f' received' RECCOT: DB 'Preceived' AGAIN: DB 'Again\$' ;Repeated record CRLE: DB DDH,0AH, '\$' ;CRLF sequence DATBUF DB CETTY ;COUNTER TO repeated record CRLE: DB DDH,0AH, '\$' ;CRLF sequence DATBUF ;Pointer to next storage location in buffer ;Databuffer	•			
FLUSH3: LX1 H,0000H  TAL C  XCHG  JMP FLUSH1 :Go write it if present  LX1 L,TPCB :Flush buffer of data  LX1 L,TPCB :Flush buffer of data  LX1 L, LbURNG :Close file  LX1 L, LbURNG :Close file  LX1 L, LbURNG :Print buffer of message  JMP HOOT :Reboot  RCVBYT: IN SIOSTA :Receive byte from serial port  ANI IPLAG JZ KCYBYT  IN SIODAT :RET  XMTBYT: IN SIOSTA :Transmit byte to serial port  ANI OFLAG JZ XMTBYT  MOV A,E OUT SIODAT :RET  FNMER: DB 'File name missing',0DH,0AH,'S'  EDTHSG: DB 'Transfer complete', OPH,0AH, 07H,'S'  ECOTHSG: DB 'Transfer terminated',0DH,0AH,07H,'S'  RECCNS: DB 'Record f' received  BECCNS: DB 'Record f' received  CRLF: DB ODH,0AH,'S' :Repeated record  DATBUF EDU S ;Data buffer :  Pointer to next storage location in buffer reporter to repeated record  CRLF: DB ODH,0AH,'S' :CRLF sequence  DATBUF EDU S ;Data buffer :  Pointer to next storage location in buffer reporter to repeated record  CRLF: DB ODH,0AH,'S' :CRLF sequence  DATBUF EDU S ;Data buffer :  Pointer to next storage location in buffer repeated record  COUNTER TO SET				;l'isk error
TAL E Point to next record  XCHG  JMP FLUSH ;Go write it if present  LXI L,TPCB  MV1 C,16  LXI L,TPCB  MV1 C,16  LXI L,EUMSG;Close file  LXI L,EUMSG  MV1 C,9  CALL BLUS ;Print EOF message  JMP HOOT ;Reboot  RCVBYT: IN SIOSTA ;Receive byte from serial port  ANI FLAG  JZ KCVBYT  IN SIODAT  RET  XMTBYT: IN SIOSTA ;Transmit byte to serial port  ANI OFLAG  JZ WATEVT  MOV A,E  OUT SIODAT  RET  ;FMMER: DB 'File name missing',0DH,0AH,'\$'  POPPER: DB 'Disk or directory full',0DH,0AH,'\$'  EOFMSG: DB 'Transfer complete', ODH, OAH, 07H,'\$'  RECCMSG: DB 'Transfer terminated', ODH, OAH, 07H, '\$'  RECCMSG: DB 'Transfer terminated', ODH, OAH, 07H	vileus.			,
CLOSE: CALL	r Losno.			Point to next record
CLOSE: CALL LEUSH LIXI LXI LXI LXI LXI LXI LXI LXI LXI LX	<b>4</b> .		L	, torne to next record
CLUSE: CALL ELUSH ; Flush buffer of data L.X1 L, TFCB WM1 C, 16 CALL BLOS ; Close file L.X1 L, EUFMSG WM1 C, 9 CALL BLOS ; Print EOF message JMP HOOT ; Reboot RCVHYT: IN SLOSTA ; Receive byte from serial port AANI IFLAG J2 KCVHYT IN SIODAT RET XMTBYT: IN SIODAT ANI OFLAG J2 XMTBYT WOV A, E OUT SIODAT RET ; FNMER: DB 'File name missing', 0DH, 0AH, 'S' COPTSG DB 'Tansfer complete', ODH, 0AH, 0'AH, 'S' EOFMSG DB 'Transfer complete', ODH, 0AH, 0'AH, 'S' RECOTS DB 'Transfer complete', ODH, 0AH, 0'AH, 'S' RECOTS DB 'Transfer terminated', 0DH, 0AH, 0'AH, 'S' RECOTS DB 'Teceived' AGAIN: DB 'Agains' ; Repeated record CRLF: DB DDH, 0AH, 'S' ; Repeated record DATBUF ; Pointer to next storage location in buffer RPTCTR: DB ETHY ; Counter for repeated record DATBUF EOU S ; Data buffer			FI.USH1	:Go write it if present
LX1 E,TFCB MV1 C,16 CALL. BLOS ;Close file LX1 L, LOFMSG MV1 C,9 CALL. BLOS ;Print LOF message JMP HOOT ;Reboot  RCVBYT: IN SIOSTA ;Receive byte from serial port AN1 IFLAG JZ KCVBYT IN SIODAT RET XMTBYT: IN SIOSTA ;Transmit byte to serial port OPLAG JZ XMTBYT MOV A,E OUT SIODAT RET ; FNMER: DB 'File name missing', 0DH, 0AH, 'S' RET ; FNMER: DB 'Disk or directory full', 0DH, 0AH, 'S' EOTMSG: DB 'Transfer complete', 0DH, 0AH, 07H, 'S' RECOMSG: DB 'Transfer complete', 0DH, 0AH, 07H, 'S' RECOMSG: DB 'Transfer complete', 0DH, 0AH, 07H, 'S' RECOMSG: DB 'Record f' received ' ' ' ' ; Record f received  AGAIN: DB 'Again\$' ; CRLF sequence DATBUF ED DATBUF ; Pointer to next storage location in buffer RETCY; Data buffer  RETRY ; Counter for repeated record  EXTRA CRLF: DB DATBUF ; Pointer to next storage location in buffer RETCYR: DB EATBUF ; Counter for repeated record ; ;	CLOSE:			
CALL BLOS ; Close file LX1 L, LorMSG MV1 C,9 CALL BLOS ; Print tOF message JMP HOOT ; Reboot RCVBYT: 1h SIOSTA ; Receive byte from serial port ANI IPLAG JZ KCVBYT IN SIODAT RET XMTBYT: IN SIOSTA ; Transmit byte to serial port OUT SIODAT ANI OFLAG JZ WATHEYT MOV A,E OUT SIODAT RET; FRMER: DB 'File name missing', DDH, 0AH, 'S' DDFER: DB 'Disk or directory full', ODH, 0AH, 'S' EOTMSG: DB 'Transfer terminated', DDH, 0AH, 07H, 'S' RECMSG: DB 'Transfer terminated', DDH, 0AH, 07H, 'S' RECMSG: DB 'Record f' received' 'RECONTS DB 'Record f' received' 'RECONTS DB 'Again\$' ; Repeated record CRLE: DB DDH, 0AH, 'S' ; CRLF sequence DATBUF DATBUF ; Pointer to next storage location in buffer RPTCTR: DB EATBUF ; Douter for repeated record CRLE: DB DDH, 0AH, S' ; CRLF sequence DATBUF EQU ; ; Data buffer	L	LX1	D, TFCB	
LXI L, LOPMSG MV1 C,9 CALL BLOS :Print LOF message JMP HOOT :Reboot RCVBYT: IN SLOSTA :Receive byte from serial port ANI IFLAG J2 KCVBYT IN SIODAT RET XMTBYT: IN SICSTA :Transmit byte to serial port ANI OFLAG J2 XMTBYT MOV A,E OUT SIODAT RET ; FNMER: DB 'File name missing',0DH,0AH,'S' EOFMSG DB 'Transfer complete', OPH,0AH,07H,'S' ECOTHSG: DB 'Transfer complete', OPH,0AH,07H,'S' RECCMSG: DB 'Transfer terminated',0DH,0AH,07H,'S' RECCMSG: DB 'Record f' RECCNT: DB '1' :Record f' RECCNT: DB '1' :Record f' CRLF: DB ODH,0AH,'S' :CRLF sequence DATBUF :DOTHOR DETTY DATBUF :Pointer to next storage location in buffer RETCTS: DB CETTY CRLF: DB ODH,0AH,'S' :CRLF sequence DATBUF EOU S ;Data buffer	1			
MVI C,9 CALL BUOS ;Print tOF message JMP HOOT ;Reboot  RCVBYT: IN SIOSTA ;Receive byte from serial port ANI IFLAG J2 KCVBYT IN SIODAT RET XMTBYT: IN SIOSTA ;Transmit byte to serial port ANI OFLAG J2 WMTBYT MOV A,E OUT SIODAT RET;  FNMER: DB 'File name missing',00H,0AH,'S' DOFFRE: DB 'Disk or directory full',0DH,0AH,'S' EOFMSG: DB 'Transfer terminated',0DH,0AH,07H,'S' RECMSG: DB 'Transfer terminated',0DH,0AH,07H,'S' RECMSG: DB 'Record # 'received' 'RECOTTSG: DB 'AGAIN: DB 'again\$' ;Repeated record CRLF: DB 'DH,0AH, S' ;CRLF sequence DATBUF :DB DATBUF ;Pointer to next storage location in buffer RPTCTR: DB ETTYY ;Counter for repeated record DATBUF EDU ; ;Data buffer	1			;Close file
CALL BLOS ; Print bOr message JMP MOOT Reboot RCVBYT: IN SIOSTA ; Receive byte from serial port ANI IPLAG JZ KCVBYT IN SIODAT RET  XMTBYT: IN SICSTA ; Transmit byte to serial port ANI OFLAG JZ XMTBYT MOV A,E OUT SIODAT RET ; FNMER: DB 'File name missing',0DH,0AH,'S' EDTHSG: DB 'Transfer complete', ODH, OAH, 07H,'S' ECOTHSG: DB 'Transfer complete', ODH, OAH,07H,'S' RECCMSC: DB 'Record f' RECCNT: DB '1' ; Record f' RECCNT: DB '1' ; Repeated record DB 'AGAIN: DB 'AgainS' ; Repeated record CRLE: DB ODH,0AH,'S' ; CRLF sequence DATBUF ; Pointer to next storage location in buffer RPTCTR: DB ETRY ; Counter for repeated record DATBUF ; Data buffer				
JMP HOOT Reboot  RCVBYT: IN SIUSTA Receive byte from serial port  ANI IFLAG JZ KCVBYT IN SIODAT RET  XMTBYT: IN SIOSTA Transmit byte to serial port  ANI OFLAG JZ WMTBYT MOV A,E OUT SIODAT RET;  FNMER: DB 'File name missing',00H,0AH,'S' DDFFR: DB 'Disk or directory full',0DH,0AH,'S' EOFMSG: DB 'Transfer complete',0DH,0AH,07H,'S' EOFMSG: DB 'Transfer terminated',0DH,0AH,07H,'S' RECCNT: DB 'Transfer terminated',0DH,0AH,0TH,'S' RECCNT: DB 'Transfer terminated',0DH,0AH,0TH,'S' RECCNT: DB 'Transfer terminated',0DH,0AH,0TH,'S' RECONTE DB 'Transfer terminated',0DH,0AH,0TH,'S' RECONTE DB 'Transfer terminated',0DH,0AH,0TH,'S' RECONTE DB 'Transfer terminated',0DH,0AH,0TH,'S' RECONTE DB 'TRANSFER TERMINATED TERMINA				
RCUBYT: IN SIOSTA RECeive byte from serial port IPLAG JZ RCVBYT IN SIODAT RET				
ANI IPLAG JZ KCVHYT IN SIODAT RET  XMTBYT: IN SICSTA ;Transmit byte to serial port ANI OFLAG JZ XMTBYT MOV A,E OUT SIODAT RET;  FNMER: DB 'File name missing',0DH,0AH,'S' DDFER: DB 'Disk or directory full',0DH,0AH,'S' EOFMSG: DB 'Transfer complete',0DH,0AH,07H,'S' EOFMSG: DB 'Transfer terminated',0DH,0AH,07H,'S' RECOTS : Record ; Recor	Decay Death			
JZ KCVHYT IN SIODAT RET  XMTBYT: IN SICSTA ;Transmit byte to serial port AN1 OFLAG JZ XMTBYT MOV A,E OUT SIODAT RET ; FNMER: DB 'File name missing',0DH,0AH,'\$' DDFER: DB 'Disk or directory full',0DH,0AH,'\$' EOTHNSG: DB 'Transfer complete',0DH,0AH,07H,'\$' RECMSG: DB 'Transfer terminated',0DH,0AH,07H,'\$' RECKOTS: DB 'Record' ; Record freceived 'RECCNT: DB '1' ; Record freceived 'RECCNT: DB '1' ; Record freceived 'CRLE: DB 0DH,0AH,'\$' ; CRLF sequence DATBUF : Pointer to next storage location in buffer RPTCTR: DB ETRY ; Counter for repeated record DATBUF EOU \$ ; Data buffer	RCVBII:			; Receive byte from serial port
IN SIODAT RET  XMTBYT: IN SICSTA ;Transmit byte to serial port ANI OFLAG J2 XMTBYT MOV A,E OUT SIODAT RET;  FNMER: DB 'File name missing',0DH,0AH,'S' DDFFR: DB 'Disk or directory full',0DH,0AH,'S' EOTHSG: DB 'Transfer complete',0DH,0AH,07H,'S' EOTHSG: DB 'Transfer terminated',0DH,0AH,07H,'S' RECOTS: DB 'Record # 'Record # received' 'RECOTT: DB ' againS' ;Repeated record CRLF: DB DH,0AH,S';CRLF sequence DATBUF :Pointer to next storage location in buffer RPTCTR: DB ENTRY ;Counter for repeated record DATBUF :Pointer to next storage location in buffer RPTCTR: DB ENTRY ;Counter for repeated record CTLF: DB ENTRY ;Counter for repeated record DATBUF :Pointer to next storage location in buffer RPTCTR: DB ENTRY ;Counter for repeated record DATBUF :Pointer to next storage location in buffer RETTY ;Data buffer				
RET  XMTBYT: IN SICSTA ;Transmit byte to serial port  ANI OPLAG J2 WATBYT MOV A,E OUT SIODAT RET  ; FMMER: DB 'File name missing',0DH,0AH,'\$' FDDFER: DB 'Disk or directory full',0DH,0AH,'\$' EOFMSG: DB 'Transfer complete',0DH,0AH,07H,'\$' ECOTMSG: DB 'Transfer terminated',0DH,0AH,07H,'\$' RECKNSG: DB 'Record # received 'RECKNT: DB '' ; Record # received  AGAIN: DB 'again\$' ; Repeated record CRLF: DB 0DH,0AH,'\$' ; CRLF sequence DATBUF : Pointer to next storage location in buffer RPTCTR: DB ETRY ; Counter for repeated record DATBUF EOU \$ ; Data buffer	ł			
XMTBYT: IN SICSTA ;Transmit byte to serial port  ANI OPLAG J2 XMTBYT MOV A,E OUT SIGNAT RET ; FNMER: DB 'File name missing',0DH,0AH,'\$' DDFFR: DB 'Disk or directory full',0DH,0AH,'\$' ECOPHSG: DB 'Transfer complete',0DH,0AH,07H,'\$' ECOTHSG: DB 'Transfer terminated',0DH,0AH,07H,'\$' RECONT: DB 'Record #' Record #' RECONT: DB 'Record #' Record #' RCONT: DB 'Record #' Record #' RCONT: DB 'SECONDER CONTROLL RECORD REC			STODAT	
ANI OFLAG J2 XMTBYT MOV A,E OUT SIODAT RET  FMMER: DB 'File name missing',0DH,0AH,'\$'  FOMER: DB 'Disk or directory full',0DH,0AH,'\$'  EOTHNSG: DB 'Transfer complete',0DH,0AH,07H,'\$'  EOTHNSG: DB 'Transfer terminated',0DH,0AH,07H,'\$'  RECKNSG: DB 'Transfer term	XMTBYT.		SICSTA	·Transmit bute to corial port
JZ XMTBYT MOV A,E OUT SIGLAT RET ; FNMER: DB 'File name missing',0DH,0AH,'S' DDFER: DB 'Disk or directory full',0DH,0AH,'S' EOTHSG: DB 'Transfer complete',0PH,0AH,07H,'S' EOTHSG: DB 'Transfer terminated',0DH,0AH,07H,'S' RECCMS: DB 'Record'; RECCNT: DB '1' ; Record f received  AGAIN: DB 'agains' ; Repeated record CRLE: DB 0DH,0AH,'S' ; CRLF sequence DATPTR: LW DATBUF 'Pointer to next storage location in buffer RPTCTR: DB ETTRY 'Counter for repeated record DATBUF LOU S 'Data buffer'	I			, manamit byte to serial port
MOV A,E OUT SIGNAT  RET  FRMER: DB 'File name missing',0DH,0AH,'\$' DDFER: DB 'Disk or directory full',0DH,0AH,'\$' DDFER: DB 'Transfer complete',0DH,0AH,07H,'\$' EOTMSG: DB 'Transfer terminated',0DH,0AH,07H,'\$' RECMSG: DB 'Record #' RECCNT: DB 'I 'Record #' RECCNT: DB 'I 'Record #' CRLF: DB 'DH,0AH, \$' CRLF sequence CRLF: DB 'DH,0AH, \$' CRLF sequence DATEUR: DB 'DATBUF' 'Pointer to next storage location in buffer RPTCTR: DB 'EETRY 'Counter for repeated record DATBUF EQU \$ ',Data buffer 'Pata Buffer'				
OUT SIGLAT RET ; FNMER: DB 'File name missing',0DH,0AH,'S' DDFER: DB 'Disk or directory full',0DH,0AH,'S' EOFMSG: DB 'Transfer complete',0PH,0AH,07H,'S' EOTMSG: DB 'Record'; ECOTMSG: DB 'Transfer terminated',0DH,0AH,07H,'S' RECMSG: DB 'Record'; RECCNT: DB '1' ; Record f received  AGAIN: DB 'againS' ; Repeated record CRLF: DB 0DH,0AH,'S' ; CRLF sequence DATPTR: LW DATBUF ; Pointer to next storage location in buffer RETCTS: DB ETTRY ; Counter for repeated record DATBUF EOU S ; Data buffer				
RET  ; PNMER: DB 'File name missing',00H,0AH,'S' DDFER: DB 'Disk or directory full',0DH,0AH,'S' EOFMSG: DB 'Transfer complete',0DH,0AH,07H,'S' EOTMSG: DB 'Transfer terminated',0DH,0AH,07H,'S' RECMSG: DB 'Record # 'received'  RECNT: DB ' 'received'  AGAIN: DB 'again\$' :Repeated record CRLF: DB 0H,0AH, S' :CRLF sequence DATEUR: DB OBH,0AH, S' :CRLF sequence DATEUR: DB ENTRY :Counter for repeated record DATBUF :Pointer to next storage location in buffer RETRY :Counter for repeated record  DATBUF : Data buffer  ; Data buffer	I			
FMMER: DB 'File name missing', 20H, 6AH, 'S' DDFER: DB 'Disk or directory full', 0DH, 0AH, 'S' EOFMSG: DB 'Transfer complete', 20H, 6AH, 67H, 'S' EOTMSG: DB 'Transfer terminated', 20H, 6AH, 67H, 'S' RECMSG: DB 'RECONT: DB '; Record # received  B' received'  AGAIN: DB 'again\$'; Repeated record CRLF: DB 'DH, 6AH, 'S'; CRLF sequence DATEUR: DB 'DATBUF'; Pointer to next storage location in buffer RPTCTR: DB 'ETRY'; COunter for repeated record DATBUF 'S 'Data buffer';  DATBUF 'S 'Data buffer'  File name missing', 20H, 6AH, 'S' 'RECORD THE NAME, S' 'RECORD THE NAME, S' 'S 'RECEIVED'  FOUNTE TO RECEIVED THE NAME, S' 'S '		RET		
DDFER: DB 'Disk or directory full',0DH,0AH,'S' EOFMSG: DB 'Transfer complete',0DH,0AH,07H,'S' ECOTMSG: DB 'Record #' RECCMSG: DB 'Record #'  BB '1' ; Record # received  DB 'received'  AGAIN: DB 'againS'; Repeated record  CRLF: DB ODH,0AH,'S'; CRLF sequence DATPFR: DW DATBUF ; Pointer to next storage location in buffer RPTCTR: DB RETRY ; Counter for repeated record  CRLF: DB CRLF: DB CRLF: Pointer to revealed record  CRLF: DB CRLF: CW DATBUF ; COUNTER for repeated record  CRLF: DB CRLF: Pointer to next storage location in buffer  CRLF: CRLF: CRLF: Pointer to next storage location in buffer  CRLF: CRL			4.1	
DIFER: DB 'Disk or directory full',0DH,0AH,'\$' EOFHNSG DB 'Transfer complete',0DH,0AH,07H,'\$' ECOTHSG: DB 'RECOTS' DB 'Transfer terminated',0DH,0AH,07H,'\$' RECCNT: DB '1' ;Record # received  DB 'received\$' AGAIN: DB 'again\$' ;Repeated record CRLF: DB 'DH,0AH,'\$' ;CRLF sequence DATPTR: DB 'DATBUF ;Pointer to next storage location in buffer RPTCTR: DB 'ETRY ;Counter for repeated record DATBUF EOU \$ ;Data buffer			File name miss	ing',0DH,0AH,'\$'
EOTMSG: DB 'Transfer terminated', 0DH, 0AH, 07H, '\$' RECMSG: DB 'Record #'  RECNT: DB 'I' : Record # received  DB 'received\$'  AGAIN: DB 'again\$' : Repeated record  CRLF: DB ODH, 0AH, '\$' : CRLF sequence  DATPER: CW DATBUF : Pointer to next storage location in buffer  RPTCTR: DB RETRY : Counter for repeated record  DATBUF EQU \$ :Data buffer  The counter for repeated record  CRLF: DB RETRY : Counter for repeated record  The counter for repeated record			Disk or direct	ory full',0DH,0AH,'\$'
RECMSG: DB 'Record # 'Record # 'Record # received DB 'received' Record # received S' Record # received S' Repeated record CRLF: DB 'agains' CRLF sequence DATEUR: LW DATBUF POINTER DB RETRY COunter for repeated record DATBUF EQU S 'Data buffer S' Counter for sex storage location in buffer CRLF DB RETRY COUNTER FOR TERMINISTER DB RETRY COUNTER DB RETRY			Transfer compl	ete', DDH, DAH, 07H, '\$'
RECCNT: DB '1' : Record # received  DB 'received\$'  AGA1N: DB 'again\$' ; Repeated record  CRLF: DB ØDH, ØAH, '\$' ; CRLF sequence  DATPER: CW DATBUF : Pointer to next storage location in buffer  RPTCTR: DB RETRY ; Counter for repeated record  DATBUF EQU \$ ; Data buffer  ;			'Record #'	nated , wuh, WAH, W7H, '\$'
DB 'received\$' AGAIN: DB 'again\$' ;Repeated record CRLF: DB ØDH, ØAH, '\$' ;CRLF sequence DATPFR: DW DATBUF ;Pointer to next storage location in buffer RPTCTR: DB RETRY ;Counter for repeated record DATBUF EQU \$ ;Data buffer			vecora *	:Pecord # remained
ACAIN: DB 'again\$ ;Repeated record ;CRLF: DB ØDH, ØAH, \$' ;CRLF sequence DATPER: EW DATBUF ;Pointer to next storage location in buffer ;Potabuff ;Counter for repeated record ;Data buffer ;				'wecord & Lecained
CRLF: DB ØDH, ØAH, '\$' ;CRLF sequence DATPER: DW DATBUF ;Pointer to next storage location in buffer RPTCTR: DB RETRY ;Counter for repeated record ; ;Data buffer	AGAIN:			:Repeated record
DATPTR: DW DATBUF Pointer to next storage location in buffer RPTCTR: DB RETRY Counter for repeated record  DATBUF EQU \$ Data buffer  The provided HTML RETRY COUNTER FOR THE P				:CRLF sequence
DATBUF EQU \$ ;Counter for repeated record;  DATBUF EQU \$ ;Data buffer;			DATBUF	
DATBUR EQU \$ ;Data buffer ;				;Counter for repeated record
	•	EQU	\$	;Data buffer
END	,	E:NIP		
		END		

### **New Products**

Continued from Page 2

plied with 25 jumper plugs, the interconnect circuitry is patched by the user; for those applications where many signals must be patched together the MINI-PATCH BOX provides two bussing areas where multiple signals maybe connected together.

Price: \$37.95

### PDIR ELLIAM ASSOCIATES

24000 Bessemer Street Woodland Hills, CA 91367 Contact: Mr. Bill Roch

PDIR is a Paging Directory Display Program for CP/M 2.2.PDIR lists the directory file names in alphabetical order, as do most other directory display programs. PDIR stops when the screen is full before going to the next screen, like most other Directory Display Programs. Thats where the similarity ends.

With PDIR you can back up to the first screen of file names then go to the second screen-toggle between screens-without restarting the program.

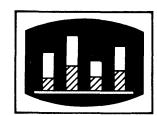
Not only does PDIR toggle between pages it

also can toggle between disk drives. If you look at the file names on drive 'A' then want to see what is on drive 'B' PDIR will switch and display the file names in drive 'B' without restarting the program. You can then go on to 'C' or back to 'A' or to whatever USER number you want on whatever drive.

PDIR also displays how many file names are being used, the amount of disk space used and remaining disk space for the current USER.

PDIR comes with an install program. Cursor positioning is required, and instructions are on the diskette. The price is only \$15.00 plus \$1.50 for shipping and handling.

Specify the disk format when ordering. Master Card and Visa are accepted.



## INCREASE YOUR SYSTEM'S PERFORMANCE & VERSATILITY WITH

SuperBrain Operating System (SOS) BIOS

For all SuperBrain or CompuStar I and II Stand-Alone Systems. \$190.00 INCLUDES:

### OS BIOS \* SOS BOOT PROM

- \* SOS BIOS \* SOS \* System Manual
- \* System Configuration Software
- \* Formatter
- \* Drive Tachometer Program
- \* WordStar © Patch File

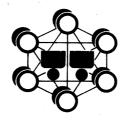
Character Set EPROMs Available SOS BOOT PROMs also Sold Separately

For more information, call (noon to midnight EST) or write:

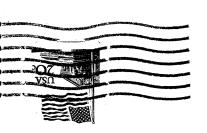
JPI SYSTEMS 2804 Fairway Drive Melbourne, FL 32901 (305) 724-6148

Attn.: Jonathan Platt

WordStar is a registered trademark of Micropro International Corporation. SuperBrain and CompuStar are registered trademarks of Intertee Data Systems.



### FIRST CLASS





# SUPERLETTER P.O. Box 3121 Beverly Hills, CA 90212

### **COMPUSTAR CABLES!**

Cables for INTERTEC COMPUSTAR Systems. Immediate delivery from stock of standard length Cables or quick delivery on cables made to specified lengths.

Cables meet same specifications as that supplied by INTERTEC. Cables are equipped with removable connectors for ease in installation in all situations including conduit.

Standard lengths - 10, 15, 25, 50, 75, 100 feet. Cables made to any length or cabling without connectors may be ordered - all at competitive prices. Dealer Quantity discounts, also.



EQUIPMENT RESOURCES, INC.

P. O. Box 723368 Atlanta, Georgia 30339

800/241-3574

404/955-0313 (in Georgia)

CDC 96 MB DISK DRIVES FOR COM-PUSTAR SYSTEMS AVAILABLE FROM STOCK. CALL US.

### **SUBSCRIPTION RATES**

U.S. 1 yr. (bi-monthly) \$25.00 Canada & Mexico \$25.00 All other Foreign \$35.00 Airmail Only Back Issues \$3.50 – 16 available Feb/Mar '81 through Aug./Sept. '83 Write To:

SUPERLETTER, Circulation Dept. P. O. Box 3121 Beverly Hills, California 90212

### SuperClassified rates are:

\$5.00 per line

(3 line minimum — 5 words per line. Please note: Publication of all Super-Classifieds are dependent on dates of receipt of copy and payment and space availability.)

Mail to: SuperClassifieds

P. O. Box 3121 Beverly Hills, CA 90212

### **Display Advertising**

Send inquiries regarding rates and specifications for display advertising to:

### **SUPERLETTER**

P. O. Box 3121 Beverly Hills, CA 90212

or call: (213) 277-2410

### SuperClassifieds

### DEALER CLOSEOUT!!!

INTERTEC computers at dealer cost! Brand new -- Full manufacturer's warranty. Limited time offer -- Limited to stock at hand! (800)845-5555 (In California call (714)832-3455)

#### FOR SALE:

1 SuperBrain SSDD, 1½ yrs. old, excellent condition. \$1395. 1 Mannesman -Tally T-1805, 200 cps printer, excellent condition, \$1195. Both for \$2500. Call Jim Johnson (805) 584-1111, (805) 583-4044!

### CDC DISK DRIVE

We can make them run better than your Tandons

- ★ \$35.00 Rate
- ★ 2 hour max. charge

if you're looking for a professional repair organization - Call today

Northshore Computer 617-827-4308