

# The Unofficial OSI Users Journal

P.O. Box 347 Owings Mills, Md. 21117 (301) 363-3268

# INSIDE

WORD PROCESSOR FOR OS1
THE 2716, PHASE 2 & MYTHS 6
RELOCATING WP6502, PART 2. 7
HOOKS INTO BASIC V1.8 8
TIME & TASK PLANNER REVIEW 12

# Column One

The deal hinted at here last month has been signed! As of November 4, ISOTRON, Inc. is in the driver's seat. Since it has been just a very few days, details are scarce —but here is what we know:

ISOTRON, the parent company of which is Investments A. B. Beijer (pronounced Bayer), one of Sweden's largest investment houses with corporate links to Volvo and others, has been involved with OSI for years through its subsidiary ISOTRONIC, the very successful Swedish OSI distributor. ISOTRON has bought the assets, trade marks, etc. of OSI that were held by the Bank of America. Robert (Bob) Lewis has been installed as President and William (Bill) Weisberg as Assistant to the President.

Old OSI employees are beginning to be called back: we hear that Jim Cross (sales) and Eric Davis (manufacturing) are back, that others will be coming in shortly, and that manufacturing will be resumed on November 14. The Bedford, MA operation will close and be moved back to Aurora, OH, thus concentrating everything there except corporate headquarters, which will remain in Trumbull, CT.

It has been reported that the net worth of Investments A. B. Beijer is about \$75 million, and we hear they are determined to keep OSI in business, continue to support existing operating systems, and press forward with new products. One spokesman said to PEEK (65), "Our base and strength lie in the machines and users currently in the field. We have to support them. No way are we going to give up OS-UI" Likewise, dealers who have stuck through the past hard times are said to be "our Number One concern."

Concerning new products, work has continued on a new product line (a work station) which should be released in the Spring. We all hope this product will be priced to be a reasonable upgrade for all "P" machine users, as well as a valuable adjunct to the larger machines. Other rumors include a 5 1/4" hard disk and IBM compatibility.

PEEK(65)

PEEK(65), as always, will do everything we can to keep information flowing both from ISOTRON to you and from you to ISOTRON.

On the subject of information movement: it is easy for those of us who have been working with our machines a while to forget how confusing it all was the first time we turned on the power switch. With the factory back in production, and with used machines changing hands, we must remember that there is a never-ending stream of new users, all just as confused as we were.

We would like to help, by publishing a series of "beginner's instruction" articles in PEEK(65), and by facilitating communication among OSI users' groups.

For the former effort, we need authors. Send us your manuscripts, not just on the latest esoteric modification you have made to your board/system, but also on your experiences starting with opening the box.

For the second effort, we need to: 1) identify OSI dealers and users and 2) help them to communicate with each other. We are continually surprised

by letters which tell us that the writer has been using his OSI computer for years, but never heard of PEEK(65), knows of no users' group in his area, etc.

To help remedy this situation, we will continue to publish (with appropriate credit) materials received from users' group newsletters, and would like to have a regular users' group activity column -- just keep us informed as to what your group will do.

We also need to know who OSI's users and dealers are. We have fliers available for distributors, dealers and users groups, which we will be glad to supply them free -- we just have to know who they are!

Finally, we hope soon to be on CompuServe's OSI SIG (see Rick Tretheway's article on TRM65U last month). We support the effort suggested in the letter from OSI Users/Boston (in this issue).

Computing isn't easy. Working with/against the various incarnations of OSI has not always been easy. It has been our pleasure and our duty to help, and our delight that so many of you have joined us. Now that things are started up again, we need each other at least as much as ever.

al

## COMPUTER SERVICE GUIDE

Courtesy of TOSIE Toronto Obio Scientific Idea Exchange

Procedures for On-site Software and Hardware Support.

- Approach the computer in a confident manner. This will undoubtedly intimidate the system and will impress anyone who might be watching. A side benefit; if the system suddenly starts working, you will be credited with solving the problem.
- 2. Wave the System Reference Manual at the machine. This will cause the monitor to assume that you are at least familiar with the sources of knowledge and are a force to be reckoned with.
- 3. If the customer asks a question (any question) begin reciting scheduling algorithms, swap-rates, function parameter tables, or anything else technical sounding. He will walk away shaking his head and you will be left alone to proceed to the next step, shaking your head.
- 4. Ask the user what his operating system parameters are. This will give you time to phone your favorite expert, since most users won't be sure what you want, never mind where to find it.
- 5. Always make sure that all cables (even power) are plugged in. If possible they should even be plugged into the right sockets. It is amazing the number of problems this has solved in the past. It is also very important that no one sees you, why let them in on your secret problem solving methods.

Cupyright #1983 by PEEK (65) Inc. All Rights Reserved. published monthly

Editor - Al Penbody
Technical Editor - Brian Hartson
C(reculation & Advertising Mgr. - Karin Q. Gieske
Production Dept. - A. Fusselbaugh, Ginny Mays

Production Dept. - A. Fusselbaugh, Ginny Mays

Subacription Rates

US (surface)

Canuda & Mexico (lst class)

So. & Cen. America (Air)

Europe (Air)

Other Foreign (Air)

533

540

All subscriptions are for 1 year and are payable in advance in US Dollars.

For back issues, subscriptions, change of address or other information, write to:

PEEK (65) P.O. Box 347 Owings Milh, MD 21117

Mention of products by trade name in editorial material or advertisements coolained herein in no way constitutes endorsements of the product or products by this magazine or the publisher.

- 6. Jar the computer alightly.
  Results of this operation
  vary, but every approach
  should be tried.
- 7. Ask the computer operator what he thinks. This will win him over. Operators are not used to being treated like thinking beings. Besides, he may know what the problem is.
- 8. Pray.



### WORD PROCESSOR FOR OSI

By: Stanley Harshfield 5758 Fox Bend Ave. Memphis, TN 38115

Like many OSI owners, I desired a word processor, but could not justify its cost based on the amount of writing that I do. As a compromise, I have been able to combine the best features of the AARDVARK JOURNAL Letter Writer (June '80, pg 12) with the OSI SMALL SYSTEMS JOURNAL Word Processor to make a word processor that is very effective for my use. The result is highly enhanced, and permits use with the least expensive printers (mine is an NCR thermal printer, with an 80 character width). Although the program is written for a CIP-MF, it should work with other OSI computers with sufficient memory (mine is 32K). As listed, the program takes up 6.7K, plus two disk buffers, but the stand-alone REMS may be removed. Even BASIC-in-ROM computers can be utilized, following the changes that I will outline.

This program makes extensive use of OSIs "GET" routine, located at 9547 (\$252B) in DOS, and at 64768 (\$FD00) in ROM. This routine is located early in the program in line 5, in order that the fastest possible response time is obtained. While expert typists may be able to outrun this routine, it will keep up with my typing. The entry pointers for this routine are located in line 1940. Later on, I'll list the changes needed for BASIC-in-Rom. In my computer, location 9504 holds the ASCII value of the key pressed. I understand that newer versions of 0565D3 may store at 9834, while the C4PMF stores at 9815. Change line 5 to agree with the model you are using.

The keyboard routine (lines 10-22) works in conjunction with the "GET" routine to convert the screen display to

upper/lower case (if desired). The keyboard works like a normal typewriter, using either right or left shifts, with the following exceptions. In the upper/lower case mode, the left shift will produce "N" or "O" when these letters are typed. If the right shift is used with these letters, the result will be an "f" or backspace, respectively. With this routine, a line may be entered at any time with a <return> (see line 11). On the other hand, a large amount of text may be entered with no <return> at all, due to the automatic parsing routine (line 20).

The program is initialized in lines 1900-1960, and the user specifies the format of output in lines 2000-2180. Answer "BOSS" or "boss" to the query "WHO ARE YOU?", and your name and address will automatically be entered for the letterhead (see lines 850-860). Otherwise, enter information as requested. Once the format is established, (upper/lower case, letter, etc.) the program is ready to accept the text. The number "1" will appear, and you may start to type. As the line fills, the number "2" will appear, and a new line will be started, without the need of carriage returns, although the <return> may be used, if desired.

If at the beginning of a line you type the single letter "H" or "h" <return>, the HELP menu will be displayed. The HELP command may be the most important command for beginners until the other commands are committed to memory. Other commands which are available include the first letter of the words List, Verify, Advance, Find, Save, Get, Edit, Print, and New.

Since the text is stored as numbered strings, it may be desired to view these strings with the List command. All or any part of them may be examined. Likewise, the Verify command may be used to examine any single given numbered string. Consecutive strings may then be viewed by using the Advance command. The user may locate a given string (or word) with the Find command. This is helpful in finding unique or mispelled words. The Save and Get commands allow the user to record to or input from disk or tape, any portion of the text that is desired. Separate texts may be combined this way. The Edit command will allow a number of options: Word, In-





sert, Delete, Replace, Global edits. The I, D, & R options refer to whole lines, while the W & G options refer to combinations of characters within lines. The Print option relates to the final edited output, and will be discussed later. Type N < return> to start over.

Also shown in the menu are the control characters that determine the format of the output. When one of these characters is used to begin a line, a special routine is invoked which will indent, skip a space, center a title, etc.. These same characters may be used within a line, with no special effect. The character "!" will indent the text following it (such as "Very truly yours,") by 40 spaces to end a letter. It is important to tell the computer when to stop printing, so use ")" as a final line, and it will signal an end to the job and then return to the menu.

The Print command invokes a very involved routine that is the final outcome of the process. You will first be asked how many characters per line (from 24 to 78) are desired. The computer will then calculate the tab setting line (from 24 to 78) are desired. The computer will then calculate the tab setting needed to center the lines on the page. As it prints, it will count out the number of words that will fit within the number of characters that you specified. If you specified right justification, it will also automatically insert extra spaces between words (inexpensive printers do not allow for proportional spacing) so that the right-hand margin is straight. This section will check for control characters at the beginning of each line, and act accordingly. If at any time, the program stops with an error message (e.g. out of memory), type "GOTO90 <return>". This will allow you to return to the program, and save all the text that you have laboriously entered.

For a disk system, prepare a floppy with a six track file for the word processor. The remaining tracks are used to set up 12 two-track files, called "FILE1", "FILE2",... "FILE12". Tracks 7-11 may be used. since they are not used. used, since they are not used by DOS by the ClP-MF. Before starting to type, run "CHANGE" and set up two buffers. They will be used to store 4K of text.



- 1 GOTO1900:\*\*\*WORD PROCESSOR\*\*\*
  5 X=USR(X):P=PEEK(9504):RETURN:\*\*\*GET SUBROUTINE\*\*\*
  10 A\$="":REM\*\*\*KEYBOARD SUBROUTINE\*\*\*
- 11 GOSUB5:Q=PEEK(57088):IFP=13THEN22 12 IFP=76THEN16

- 13 IFQ=252ANDP=94THEN19
  14 IFQ=250ORQ=252ORQ=255ORQ=218THEN18
  15 IFQ=220THENPRINTCHR\$(8);:A\$=MID\$(A\$,1,LEN(A\$)-1):GOTO11
- 16 IFT0=2ANDP>64ANDP<91THENP=P+32 17 GOTO19 18 IFF>80ANDP<107THENP=P-16

- 18 IFPSUANDECTOTHEND=P-16
  19 IFP=64THEND=80
  20 PRINTCHRS(P); IFLEN(AS)>55ANDP=32THEN22
  21 AS=AS+CHRS(P):GOTOLL
  22 PRINT:RETURN
  89 REM\*\*\*TEXT ENTRY SECTION\*\*\*

- 89 REM\*\*\*TEXT ENTRY SECTION\*\*\*
  90 PRINT:PRINTL;:GOSUB10
  100 IFLEN(A\$)=1THEN130
  110 A\$(L)=A\$:L=L+1:GOTO90
  130 A=A\$C(A\$):IFA=70ORA=102THEN950:FIND
  140 IFA=76ORA=108THEN280:LIST
  150 IFA=71ORA=103THEN670:GET TEXT
  160 IFA=80ORA=112THEN320:PRINT
  170 IFA=83ORA=115THEN500:SAVE TEXT
  180 IFA=72ORA=104THEN775:HELP
  190 IFA=78ORA=110THEN500:NEW DOCUMENT
  200 IFA=86ORA=118THEN710:VERIFY
- 200 IFA-86ORA-118THEN710:VERIFY 210 IFA-65ORA-97THEN730:ADVANCE LINE 220 IFA-69ORA-101THEN1600:EDIT
- 230 GOTOL10
- 279 REM\*\*\*T.TST\*\*\*

- 279 REM\*\*\*LIST\*\*\*
  280 GOSUB3000:INFUT"FRCM (type A for all)";A\$
  285 IFASC(A\$) =65THENX=1:Y=L-1:GOTO300
  290 X=VAL(A\$):INFUT"TO";Y:FRINT:IFY>LTHENY=L
  300 PRINT"HARD COPY?":PRINT:GOSUB5:IFP=89THENGCSUB3010
  310 FORI=XTOY:PRINTI;A\$(1):NEXT:GOSUB3020:GOTO90
  319 REM\*\*\*PRINT\*\*\*

- 320 GOSUB3000
- 320 GOSUB3000
  321 R=0:INFUT"HOW MANY CHARACTERS FER LINE (MAXIMUM 78)";WIDTH
  322 IFWI>=78THENWI=78:H=WI:T=0:G=T:GOTO335
  325 IFWI<24THENWI=24:H=WI
  330 WI=INT(WI):H=WI:T=INT((78-WI)/2):G=T
  335 PRINT:PRINT"RIGHT JUSTIFICATION?":GOSUB5:IFP=89THENR=1
  340 PRINT:PRINT"HARD COPY?":GOSUB5:IFP<89THENRFINT:GOTO380

- 350 GOSUB3010

- 380 PRINT: IFF>1THEN395
  390 PRINTTAB(39-(LEN(N\$)/2))N\$
  391 PRINTTAB(39-(LEN(C1\$)/2))C1\$
  392 PRINTTAB(39-(LEN(S1\$)/2))S1\$:PRINT
  395 IFF<>2THEN402

- 400 PRINTTAB(60-T)C1\$:PRINTTAB(60-T)S1\$
  402 IFF=4THEN420

- 402 IPF=4THEN420
  405 PRINTTAB(60-T)D\$: PRINT: PRINT: IFF=3THEN420
  410 PRINTTAB(T)B\$: PRINTTAB(T)C\$: PRINTTAB(T)\$: PRINT: PRINT
  420 T\$="":FORX=ITGL-1:T\$(X)=A\$(X): IFASC(T\$(X))<48THENGOTO1200
  425 IFX=1THENT\$=T\$(X)
  430 IFX>1THENT\$=T\$+" "+T\$(X)
  435 IFLEN(T\$) <WITHENEXTX
  440 IFLEN(T\$) <WITHENEXTX
  440 IFLEN(T\$) <WITHENFR\$=T\$:T\$="":GOTO485
  450 Q\$=LEFT\$(T\$,WI):FORY=WITOLSTEP-1:IFMID\$(Q\$,Y,1)<>" "THENNEXTY
  460 PR\$=LEFT\$(Q\$,Y):T\$=RIGHT\$(T\$LEN(T\$)-LEN(PR\$))
  465 IFRIHENGOSUB1050
  485 PRINTTAB(T)PR\$:IFFITHEN491
  486 IFLEN(T\$)=>WITHENA45

- 486 IFLEN(T\$)=>WITHEN445
- 490 NEXTX 491 FL=0:IFF=30RF=4THEN495
- 492 PRINT: PRINT: PRINT: PRINTTAB (40) NS
- 495 GOSUB3020:GOTO90 499 REM\*\*\*SAVE TEXT\*\*\*

- 759 RAWL-SAVE TEXT-FROM LINE # (type A for all)";X\$
  510 IFASC(X\$)=65THENX=1:Y=L-1:GOTO530
  520 X=VAL(X\$):INPUT"TO";Y:FRINT:IFY>L-1THENY=L-1
  530 GOSUB3050
- 540 FORI=XTOY:PRINT#6,A\$(I):NEXT:PRINT#6,"&&&&"
  550 GOSUB3070:GOTO90

- 559 REM\*\*\*REPLACE LINE\*\*\*
  560 INFUT"REPLACE LINE #";1:1F1>L-1THEN90
  570 PRINT:PRINT"REPLACE:":PRINT" "A\$(1):PRINT:PRINT"WITH:":PRINT
- 580 GOSUB10
- 590 A\$(I)=A\$:GOTO90

Continue



Continued

```
629 REM***INSERT LINE***
630 INPUT"INSERT AFTER LINE &";X:PRINT:IFX>LIMEN90
640 PRINT"?";:GOSUB10
  650 L=L+1:FORY=LUCX+2STEP-1:A$(Y)=A$(Y-1):NEXT:A$(X+1)=A$:GOTO90
669 REM***GET TEXT***
  670 GOSUB3000:GOSUB3030
670 GCSUB3000:GCSUB3030
680 INPUT#6,A$:IFA$="&&&&"THEN90
690 A$(L)=A$:L=L+1:PRINT:PRINTA$:GONO680
709 REM***VERTFY LINE***
710 GCSUB3000:INPUT"WHICH LINE DO YOU WANT TO VERIFY";I:PRINT
715 IFI=OTHENPRINT:PRINT"TOO LOW, TRY AGAIN":GONO710
720 PRINT:PRINTI;A$(1):GONO90
729 REM***AUVANCE LINE***
730 GCSUB3000:InTL1:GONO200
729 REM***AIVANCE LINE***
730 GCSUB3000:I=I+1:GOTO720
749 REM***NEW DOCUMENT***
750 FORX=0TCI+1:A$(X)="":NEXT:PRINT:PRINT:I=1
760 GOSUB3000:PRINT"NEW DOCUMENT":PRINT:GOTO2000
774 REM***HELP--LIST COMMANDS***
775 GOSUB3000:PRINT" Help
777 PRINT" Find word
780 PRINT" Edit text
782 PRINT" Verify line
785 FRINT" Advance line
788 FRINT" Save text
790 PRINT" Get text
792 PRINT" List
795 FRINT" Print
  795 PRINT" Print
797 PRINT" New document
800 PRINT: PRINT
800 FRINT: PRINT
805 FRINT" &:CENTER LINE ON PAGE
810 PRINT" /:NEW LINE
815 FRINT" *:SKIP LINE
820 FRINT" *:SKIP LINE
820 FRINT" *:INDENT LINE
821 FRINT" *:INDENT LINE
822 FRINT" *:INDENT SUB-SECTION
833 PRINT" ::INDENT CLOSING
840 FRINT" ::INDENT CLOSING
840 FRINT" ::END OF TEXT":PRINT:PRINT:GOTO90
849 REM***ENTER LETTERHEAD**
850 N$="Stanley Harshfield"
860 Cl$="5758 Fox Bend Ave.":SL$="Memphis, TN 38115":GOTO2140
869 REM***WORD EDIT***
870 INFUT"EDIT WHAT LINE":I:PRINT:PRINTI:AS(I)
809 REMARKURI EUITAKA

870 INFUT"EDIT WHAT LINE";I:PRINT:PRINTI;A$(I)

875 PRINT:PRINT"CLD STRING?":GOSUBLO:Q$-A$

878 PRINT:PRINT"NEW STRING?":GOSUBLO:X$-A$

880 PRINT:PRINT"REPLACE: "Q$:PRINT:PRINT"WITH: "X$

885 PRINT:PRINT"CORRECT?":FRINT

890 GOSUB5:IFP<897HEN90
  900 Y=LEN(Q$):FORX=1TOLEN(A$(I)):IFMID$(A$(I),X,Y)=Q$THEN930
920 NEXTX:GOTO720
930 REXTX:GOMO/20

930 REXI:HENAS(I)=XS+MIDS(AS(I),X+Y):GOTO/20

940 AS(I)=LEFTS(AS(I),X-1)+XS+MIDS(AS(I),X+Y):GOTO920

949 REM***FIND STRING***

950 GOSUB3000:FRINT*WHAT STRING ARE YOU LOOKING FOR?":GOSUB10
950 GGSDBJOUD: FHCINI WHAT SERING ARE YOU LOOKING FOR.
960 PRINT: INPUT ON WHICH OCCURRENCE"; A: IFA=0GOTO960
970 X=0:B=LEN(A$): FORI=1TOL: FORY=1TOLEN(A$(I))
980 IFMID$(A$(I),Y,B)=A$THEN1000
990 NEXTY,I: PRINT: PRINTA$" NOT FOUND": GOTO90
1000 X=X+1: IFX<ATHEN990
1010 GOTO720

1049 REM***RIGHT JUSTIFICATION***

1050 A-WI-LEM(PR$):IFA-OHENRETURN

1060 Z-1:Y=1:PORV=1TOLEM(PR$)

1070 IFMID$(PR$,V,1)<>""HEENNEXTV

1080 F$(Y)=MID$(FR$,Z,V-Z+1):IFV=LEM(PR$)+1THEM1110

1100 Z-Z+LEM(F$(Y)):Y=Y+1:NEXTV

1110 FORV=1TDY-1:F$(V)=F$(V)+"":IFV=AHEM1150

1140 NEXTV:A-A-V+1:GOTOL110

1150 PR$="":FORV=1TOY-1:PR$=PR$+F$(V):NEXTV:IFM=OHEMM=1:REFURN

1180 M=0:REFURN
  1010 GOTO720
  1180 M=0:RETURN
1180 M=0:REIURN
1199 REM***SEPECTAL COMMAND ROUTINE***
1200 IFASC(A$(X))=42THEND=1:GOTO1300:42=*
1210 IFASC(A$(X))=47THEND=2:GOTO1300:47=/
1220 IFASC(A$(X))=33THEND=3:GOTO1300:33=1
1230 IFASC(A$(X))=38THEND=4:GOTO1300:38=6
1240 IFASC(A$(X))=37THEND=5:GOTO1300:37=%
1245 IFASC(A$(X))=43THEND=6:GOTO1300:43=+
1250 IFASC(A$(X))=35THEND=7:GOTO1300:35=%
1260 IFASC(A$(X))=41THEN1550:41=)
1270 GOTO425
 1270 GONC425
1300 T$(X)=RIGHT$(T$(X),LEN(T$(X))-1)
1310 CNDGCRC1360,1360,1400,1450,1500,1350,1350
                                                                                                                                                                                                                            Continued on page 6.
```

COMPUTER REPAIRS CLP - C2P - C4P

Bave your personal computer serviced by a qualified technicien familiar with OSI hardware. We will evaluate your computer and notify you of what should be done and how much it will cost. Any repairs will be made only by your approval. Please include a description of the problem if it is intermittent. Minimum charge is \$20 tent. Minimum charge is \$20 whether repairs are made or not.

CLP SERIES I OWNERS: CIP SERIES I CMNERS: A 24/32 column video mod. is available! Go back to a 24 column display at any time with just a flick of a switch. Uses the prototype area without piggybacking any IC's. Only \$89.95 installed.

Please include \$4.50 postage.

The Computer Shelter 8533 Facific Hwy SE Olympia, WA 98503

For tape users, following changes: make the

- X=USR(X): P=PEEK(531):RETURN
- IFQ=252ANDP=94THEN20 IFQ=252ANDP=95THEN17
- 14 15 IFQ=250 ORQ=252THEN18 IFT0=2ANDP>64ANDP<91THEN
- P=P+32
- GOTO19 16 17
  - PRINTCHR\$(95);:A\$=MID\$
    (A\$,1,LEN(A\$)-1):GOTO11
    PORI=XTOY:PRINTA\$(I):
    NEXT:PRINT"&&&&
- 540
- 680
- INPUTA\$: IFA\$="&&&&"THEN POKE515,0:GOTO90 A\$(L)=A\$:L=L+1:PRINT: GOTO680
- 690
- 1940 POKE11,0:POKE12,253 3000 FORX=1T032:PRINT:NEXT:

- RETURN
  3010 POKE517,1:RETURN
  3020 POKE517,0:RETURN
  3030 PRINT"TURN ON TAPE PLAYER
  3040 POKE515,255:RETURN
  3050 INPUT"IS TAPE RECORDER
- ON"; A\$
  3060 POKE517,1:RETURN
  3070 POKE517,0:RETURN

It will also be necessary to delete line 1960. Does anyone out there know the location of the , & : terminators in BASIC-in-ROM?

I hope that this program will be as much value to you as it has been to me. I find that it is used much more than I expected, since I can now turn out letters and reports with no erasures or obvious corrections.





3 USERS-80 Mega Bytes —\$999000\* INTRODUCTORY

WITH DUAL FLOPPIES SPECIAL 1 YEAR WARRANTY ON HARD DISK! BRAND NEW -

• 90 Days on Power Supply, Floppy Drives — Circuit Boards.

Configured for Time-Share @ 2 MHZ

 Includes: 2 Serial Printer Ports with Handshake, Improved Cooling, and Ball Bearing Roller Chassis Rails

ALSO AVAILABLE WITH 3 MULTI-PROCESSOR

Denver Boards with 64K each user and Centronics Parallel Printer Port at

\$10.990.<sup>00</sup>

\*DEALER DISCOUNTS AVAILABLE

# 8" HARD DISK SYSTEMS

SINGLE BOX TABLE TOP WITH IMPROVED COOLING 10 M/B HARD DISK AND 8" FLOPPY DISK 2 USERS AND 2 SERIAL PRINTER PORTS \$**5990**\_00

AS ABOVE WITH 2 MULTI-PROCESSOR 64K DENVER BOARDS PLUS CENTRONIC PARALLEL INTERFACE \$6990.00



\$6490.00 1 USER w/ Contronics Printer Port \$6990.00 2 USER w/ 2 Serial Printer Ports

\$7790.00 2 USER w/Centronics Printer Port

MULTI
PROC. 3 USER w/Centronics & Serial Printer Ports \$8990.00 **MULTI-PROCESSOR** EVELOPMENT SYSTEM SPECIAL

5 M/B Hard Disk-1 8" Floppy

• 1 Serial Printer Port, 1 Modem Port ONLY \$5990.00

• 2 DB-1 Multi-Processors

• Complete Programmer Manual and Software Overlays

DEALERS — We have lots of OSI machines and can build virtually any combination you need. Appropriate dealer discounts.

Please Give Us a Call!

WHERE WE STILL LOVE OS-65U — AND SUPPORT IT!

22991 LA CADENA DRIVE, LAGUNA HILLS, CALIFORNIA 92653

ORDER TODAY

1714) 951-4648

**SOME QUANTITIES LIMITED** 

THE 2716, PHASE 2 AND OTHER MYTHS.

By: Paul C. of TOSIE

Almost every hardware question I am asked involves an EPROM, a simple device that continues to confuse a lot of people. It also confused OSI since even they have made design errors using this chip. I don't want to talk about EPROM theory, since it is covered in any electronics book, but I do want to point out what might have been missed. The pinouts are very straight-forward, the 'A' lines are the address lines, the 'O' or 'D' lines are the data lines, GND is ground and Vcc is the supply voltage typ. +5 volts. The pins that cause problems for most beginners are 18 (CE), 20 (OE) and 21 (Vpp). Vpp (program voltage) is used for just that, programming the chip, in normal use (when not programming) this pin should be equal to Vcc (+5V). Grounding this pin will not harm the chip but it is not a logical state and most 2716 will not work if it is. What confuses people is that some makes do work but they shouldn't. In short the Vpp should be +5V and then forget about it. It should also be noted that during application of power, care must be taken to assure that Vcc is applied before or simultaneously with Vpp and Vpp must be removed before or simultaneously with Vpc or you may be buying a new chip.

Pins 18 and 20 are both active low, when the pin is equal to Vcc the function is disabled and enabled when the pin is grounded. Chip Enable (CE) is the power control, and must be used to select the device. Output Enable (OE) controls only the ouput stages and must be used to gate data to the output pins. The common use of these chips with a 6502 system is to activate CE as soon as the address lines indicate the chip is to be selected and use phase 2 to gate the data. The time required for data to be available for output from the time the chip is selected (CE=O) is known as the access time. Typical access times are shown in table 1. The max. access time for a system at 1MHz is approx. 700 to 1000 nanoseconds depending on how the 2716 is used in the circuit. A common mistake involving access times is to use phase 2 in the address decoding that controls the Chip Enable line. This means

2080 IFF=4THEN775 2090 IFF<3THENPRINT"YOUR NAME?":GOSUB10:N\$=A\$

2110 PRINT"YOUR STREET ADDRESS":GOSUB10:C1\$=A\$
2130 PRINT"YOUR TOWN & STATE":GOSUB10:S1\$=A\$
2140 PRINT"PRINT"DATE?":GOSUB10:D\$=A\$
2150 IFF<\1CRF>2THEN775

3040 DISK OPEN,6,A\$; REJURN
3050 DISK OPEN,6,A\$; REJURN
3060 DISK OPEN,6,A\$; REJURN
3070 DISK PUT:DISK CLOSE,6; REJURN

2150 IFF</ORF>ZHEM775
2160 PRINT:PRINT"NAME OF RECIPIENT?":GOSUBl0:B\$=A\$
2170 PRINT:PRINT"STREET ADDRESS?":GOSUBl0:C\$=A\$
2180 PRINT:PRINT"TUMN & STATE?"\*GOSUBl0:S\$=A\$:GOTO775
3000 POKE9803,33:PRINT:POKE9803,0:RETURN:\*\*\*\*SCREEN CLEAR\*\*\*
3010 DISK!"10 ,03":NULL6:RETURN:PRINTER ON
3020 DISK!"10 ,02":RETURN:PRINTER OFF
3030 INFUTTGET WHICH MEMORY";A\$:A\$="FILE"+A\$

2100 IFF>2THEN2140 2110 IFN\$="BOSS"ORN\$="boss"THEN850 that the chip is not even selected till the CPU is half-way through its cycle and the max. allowable access time is therefore divided by two. An example can be seen in the decoding of the BASIC ROMs on the Superboard Rev D.

Even though this is a mistake, the Superboard does still work, the problem is if you try to go to 2MBz the ROMs appear too slow for the system. There are two solutions if you have problems with slow access times, you can buy faster EPROMs which cost more or you can try the following: If CE is active at all times and OE is used to

select the ROMs the output enable to output delay time is typically 120 nanoseconds, more than fast enough for most uses. The drawback of this application is that the EPROM will be drawing full power at all times. Power dissipation when selected is 1.0W, and when in standby mode (CE=1) 0.132W.

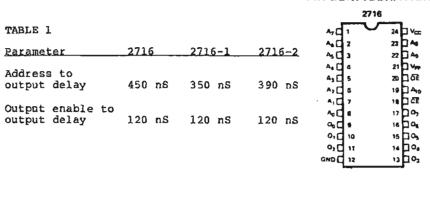
The other common problem arises when substituting 2716s for the OSI BASIC ROMs or monitor ROM. In many models OSI used the 2316; the difference is that this chip is programmed during manufacturing of the chip and that pins 18, 20 and 21 are all

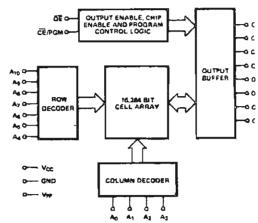
programmable chip select lines. In typical OSI fashion, they had these lines programmed differently from the 2716, i.e., they may or may not be active low. Therefore, when replacing these ROMs you may have to invert signals to some of these pins.

After rereading this article, I realize two things: I'm not a technical writer, and no wonder people get confused. I hope this article helps somebody but if you still have questions all you have to do is ask. Values used were taken from the Synertek Data Book.

### BLOCK DIAGRAM

## PIN CONFIGURATION





### RELOCATING WP-6502 Part 2

By: John T. Roecker 5141 Thomas Ave., S. Minneapolis, MN 55410

I was overjoyed after purchasing an Epson MX80 printer. I had relocated WP-6502 in order to use it with my ClP with a ClS monitor ROM. I knew WP-6502 was working because I could create a tape of an article or letter and take the tape to a friend who had a printer to have it printed. This was inconvenient and also could possibly tax a friend-ship. I had the RS-232 interface populated so all I had to do was connect the Epson to it. I ran a quick test in BASIC and the printer worked fine. However, when I attempted to use WP-6502, the printer did not! After much head scratching, I remembered that I had to use a different output routine in my ClS monitor ROM in order to output to tapes using WP-6502. I surmised I would have to use this routine to output to the printer also. Those of you

who have C1E/C2Es can rest easy because these ROMs appear to use the standard output routines. A quick test checked my idea, WP-6502 would drive the printer with another modification.

My modification to WP-6502 was to add a new command, the Print command, to the WP-6502 repertoire and to disable using the View command to output to the printer. Those of you with standard OSIs or with ClE/C2E monitor ROMs may find this new command useful.

I used the following steps to add the Print command to WP-6502. All address locations mentioned are the original addresses from your nonrelocated tape version of WP-6502. All instructions with a \* behind them will have to have their address fields modified to suit your relocation. All instructions with a behind them are new instructions which have been added.

1. Expand the WP-6502 menu so that the Print command may be added. The menu plus other

words outputted to the screen are located at memory locations \$070D through \$0783. I modified the menu to have it look like this:

---WP 6502
Type
View
Blk View
G/Edit
L/Edit
Move
Print
Zap
R/Tape
W/Tape

I used the OSI Extended Monitor to relocate locations \$0736 through the end of the cold start code of WP-6502, \$0FDO, by 5 bytes. Then I added the new Print command at \$0736:

\$0732 4D6F76E5 Move \$0736 5072696EF4 Print \$073B 5A61F0 Zap

2. Contract the View command code to eliminate the Pr? after View. The View command code is located at memory

locations \$0795 through \$09F9. I dropped the instructions located at \$0798, \$079A, \$079D and \$079F by relocating \$07Al to \$0798.

3. In the process of performing these two relocations, I managed to destroy two instructions. One of them stopped L/Edit and G/Edit from working from the menu. The instruction which was destroyed for this problem was located at \$078F. The address field of this instruction should be modified by the relocations performed in the previous steps.

# \$078F 20940A JSR \$0A94 \*

The second instruction which was destroyed caused an insert at the 'End of Text' to operate improperly. This instruction should be:

## \$0C46 4C5A0F JMP \$0F5A \*

4. Any references which index into the WP-6502 menu data area may have to be corrected because we added a new command. References for commands and data after the new Print command will have to be modified by adding 5 bytes to the immediate data:

\$03FA LDY #\$52 LDY #\$40 LDY #\$5B A052 \$0465 \$06AD A040 A05B LDY #\$55 LDY #\$43 CPY #\$4E LDY #\$50 \$0787 A055 \$07B4 A043 S07BB C04E \$07E2 A050 \$09B0 A06D LDY #\$6D LDY #\$6D \$09F0 A06D LDY #\$60 A060 A03E A071 #\$3E #\$71 SOB1 2 LDY \$0D01 A060 LDY #\$60 LDY #\$60 SOD44 A066 LDY #\$66 LDY #\$6D \$0EA9 A06D

5. The warm start code will have to be modified so it will recognize the Print command. Those of you with ClS/C2S monitor ROMs will have to add this check after the switch to the new output routine which was added in the last article.

This is what the code looked like before the change:

\$0F8F E057 CPX #\$57 Check for W/Tape \$0F91 D003 BNE \$0F96 \$0F93 20F30E JSR \$0EF3 \$0F96 4C6504 JMP \$0465 Not legal

command

This is after the change:

\$0F8F E057 CPX \$\$57 Check for W/Tape

\$0F91 D003 BNE \$0F96 \$0F93 20F30E JSR \$0EF3 \* \$0F96 E050 CPX #\$50 & Check for Print \$0F98 D003 BNE \$0F9D & \$0F9A 4C9807 JMP \$0798 &\* Print

\$0F9D 4C6504 JMP \$0465 \*

\$0FA0 5D00 \$0FA2 40

Starting text location

output

Not legal command

6. The cold start code will have to be modified to use the proper data/text starting location. I have indicated this location in step 5 above; in this case \$0FA2. The amounts of the relocations will have to modify this address. I have reproduced all the cold start code below for clarity. I have placed my cold start code at \$1024.

Store \$1026 8500 \$1028 A90F STA SOO warm LDA #\$OF start amuir \$102A 8502 STA \$02 \$102C A90B LDA #\$0B instruction \$102E 8501 STA SOL \$1030 A924 LDA #\$24 \$1032 8503 STA \$03 cold start \$1034 A910 LDA #\$10 address \$1036 8536 STA \$04 \$1038 A90F \$103A 8D4202 STA \$0242 \* starting \$103D A9A2 LDA #\$A2 Address \$103F SD4102 STA \$0241 \* \$1042 A900 LDA #\$00 \$1044 8546 STA \$46 \$1046 ADE2FF LDA \$FFE2 Test for CIP \$1049 DOOA BNE \$1055 Branch if not \$104B A914 LDA #\$14 \$104D 8D3602 STA \$0236 \* \$1050 A9FF LDA #\$FF \$1052 8D4002 STA \$0240 \* Jump to warm start \$1055 4C0000 JMP \$0000

The immediate data at \$1039 and \$103E will have to be modified to point to your starting text address.

Now, after much blood, sweat, and some tears, those of us with nonstandard monitor ROMs installed may use WP-6502. To eliminate all this work, all machine language / Assembler code should start at a suitably high address. The Assembler/Editor starts at location \$0240. I feel this would be a good starting address to enable anyone with standard or nonstandard monitor ROMs to use your program.

I have made additions to WP-6502 to utilize some of the features of my ClE monitor ROM. These additions will be the third article in this series.

### HOOKS INTO BASIC V1.8

by: Rick Trethewey 8 Duran Court Pacifica, CA 94044

In September 1980, OSI published a method of adding keywords into BASIC under OS-65D. Since that time, I have been expanding on that theme. My first attempts were published in the April 1981 issue of MICRO and a subsequent version was published in the book "MICRO on the OSI". Unfortunately, MICRO sat on the version they put in their book for about a year and even since then I have continued to expand on the code. My feeling has always been that despite the obvious shortcomings of BASIC, it remains the most used language on OSI systems and thus anything that enhances that language is a worthwhile project. I chose OS-65D V3.3 because it is the latest (and perhaps last) "official" release of the operating system and it is virtually identical in both the 8 inch and minifloppy versions. The last point makes it possible for the same code to run on everything from a C4P-MF to a full-blown C-3. However, there is a price to be paid for these additions in lost memory. The object code for "Hooks" requires 10 pages of memory. That's fine for a 48K system and barely acceptable on a 32K system. But if you only have 24K, Hooks will only leave you with an 8K workspace unless you delete some parts of the code.

Hooks' primary latch into BASIC is in the code that handles equations. When BASIC starts to look at a line of code, it checks the first character to see if it's a BASIC token representing a command. If it is a token, the appropriate command is executed. If it is not a token, BASIC assumes that an equation is in progress and begins to handle a variable name. Hooks intercepts the variable name handler and checks to see if the text is one of the new keywords and if it is not, control is passed back to BASIC. If Hooks does recognize a "keyword" it retains control and executes one of the new commands. There are two "gotcha's" involved here. First of all, there is a reduction in the speed of BASIC. The speed overhead isn't that much and since BASIC isn't noted for speed anyway, this should not become a problem. The second is that all of the new Hooks





# Resolution Color Graphics

Our new Color Plus board provides 256 x 192 high-resolution graphics with 15 colors. Two 8-bit resolution joystick interfaces are included. Software extensions to OS65-D BASIC provide a superset of APPLE II® graphics instructions.

Color Plus connects to the standard 48-pin bus or the 16 pin bus.

Pricing:

\$195 CP-8 for C8 or C3 computers: CP-4 for C4 computers (5V only): \$245 CP-bare Bare board with software: \$ 75

# **Generos Operating Syst**

Generos is a new operating system for OSI computers. Features include descriptive error messages, optimized disk usage, device independence, and:

- ASSEM Fast disk based assembler generates relocatable code.
- BASIC Basic Interpreter
- DDT Machine language debugger with single step, trace, more.
  TECO — Most powerful and widely used
- text editor ever.

Currently available for 48K 8-inch systems. Call or write about availability for other configurations. Cost: \$85

# Low Power Memory Board

# Our popular MEM + board is ideal for:

- Partitions for multi-user systems
- 64K CP/M systems when combined with the D&N-80 CPU board
- Upgrading systems where backplane space, low power consumption, and/or low heat dissipation is required

## **Options** include:

- OSI compatible floppy disk controller protects against disk crashes caused by power failures
- Real time clock/calendar Date and time with battery backup
- Centronics parallel printer interface -Supported by software that automatically patches OS65D and OS65U
- One year warranty

VISA, MasterCard, personal checks and C.O.D.s all accepted. Add \$5 per board for shipping and handling.

To order, or for more information, contact:

Fial Computer 5221 S.W. Corbett Portland, Oregon 97201

(503) 227-7083

# MEM + includes the following features:

- Memory chips in quality production sockets; high reliability machine screw sockets optional
- Low power consumption
- Uses 2K x 8-bit memory chips will accept 2716-type EPROMS.
- Versatile addressing

# **NEW LOW PRICES!!**

Bare — Solder masked and silkscreened \$75 Software and documentation provided

16K	\$200		
24K	\$250	Disk controller	\$85
32K	\$300		
40 K	\$350	Real time clock	\$65
48K	\$390		
52K	\$415	Centronics interface	\$45
56K	\$440		
64K	\$490	High-rel sockets add	15%



5740 S.E. 18th Ave. Portland, OR 97202



Continued from page 8.

keywords are also reserved and cannot be used as variable names so some of your old programs may require editing.

To begin, you'll have to in the source file for Hooks. Without all the comments, this takes about 9 tracks on an 8" Hooks system. Clearly, systems with less than 48K are going to run into trouble using the OSI Assembler/Editor here. For those people, I suggest buying the OSI Assembler/Editor Manual for \$5.95. That book desual for \$5.95. That book describes the method of carrying over the symbol table from one source file to the next, thus allowing the linking of source files. The alternative would be to simply break up the source code into separate files and make sure that each file includes the address labels it needs. Both of these methods are a bit tedious and will make you very aware of the benefits of assemblers that automatically link source files. You will also have to make two changes assemblers that automatically link source files. You will also have to make two changes to the source file if you have less than 48K. The first is the origin address. You should make this \$7600 if you have 32K and \$5600 if you have 24K. The second change is another origin address that is set just before the code that allows BASIC to understand EEX in expressions. This should allows BASIC to understand HEX in expressions. This should be changed to \$7ElB on 32K systems and \$5ElB on 24K systems. Except for these two changes, both the source code and the BEXEC\* program will automatically configure themselves to your system's disk and memory sizes. Before you selves to your system's disk and memory sizes. Before you assemble the Hooks code, you should prepare an OS-65D V3.3 diskette with two files of 1 track each for 8" systems and 2 tracks each on minifloppies. Name the first file "BEXEC\*" and the second file "BASIC+". The file "BASIC+" will hold the Hooks object code. When you have assembled and saved the Hooks code to disk, enter and save the BEXEC\* program listed here. This BEXEC\* will automatically make the necessary changes to This BEXEC\* will automatically make the necessary changes to BASIC to incorporate Hooks on the initial boot-up and will display the diskette's directory and a menu of options.

Now let's talk about what all that typing will gain for you. The following "keywords" are supported by Hooks:

C2

Clears the 540 black and white video.

### B\*exp

Fills the 540's color background with the value of the expression following the asterisk and sets the cursor backround to the same color.

### Q\*exp

PORES 56832 with the value of the expression following the asterisk to set the 540's color, sound, and character size.

R

Enables "NEW", "LIST" , and <CTRL>'C'.

T\*

Toggles the BASIC line trace on and off just as if the program "TRACE" had been run.

S\*a\$

Selects the disk drive corresponding to the string (literal or variable) that follows the asterisk. This string must be only one character long and in the range of A-D (upper or lower case) or a SYNTAX ERROR will result.

D\*

Prints the directory of the currently selected drive. You'll note that the BEXEC\* tabs the "\* Directory \*" message to center it a bit, but that the code will usually left-justify this.

ASM

Invokes the Assembler/Editor.

en

Invokes the Extended Monitor.

LOAD"FNAME

Loads the file "FNAME" into the workspace.

SAVE"FNAME"

Writes the current workspace contents to the file "FNAME". If "FNAME" already exists, then the current file length is checked to make sure that the workspace contents will fit BEFORE the attempt is made to write out the file. This prevents an "ERR #D ERROR" from trashing the current file contents. If the disk file is not big enough to hold the workspace contents, an error message is displayed saying so. If "FNAME" is not found in the directory, Books automatically creates a file of

sufficient size to hold the current workspace contents. Again, if there are not enough contiguous free tracks on the disk to make such a file, an error message is displayed. "SAVE" can be used on both BASIC programs or Assembler source files loaded while in BASIC. However, Hooks looks at location \$3A7E to determine if the file is a BASIC or Assembler file. This will provide correct results unless a BASIC program with a buffer at the start of the workspace. To make sure of proper results, POKE \$3A7E,0 (remember, that's legal now!) before using SAVE on these files. Yes, this logic will also fail on Assembler files whose first line number is an even multiple of 256, but I don't know of anyone using such numbering.

# MAKE"FNAME", exp

Creates a file named "FNAME" whose length is the value of the expression that follows the comma. Variables can be used for the file name and the desired length can be the result of a calculation.

RENAME "OLDNAM" TO "NEWNAM"

No mystery here, the file "OLDNAM" is renamed "NEWNAM". The code checks for duplicate and legal file names.

KILL"FNAME" (,"F2NAM",...)

Deletes all of the file names following the keyword "KILL" separated by commas.

PACK

Frees up all unused space on the disk by moving all files to the lowest possible track number. This makes all unused disk space available in a contiguous block at the end of the disk. This command requires a disk buffer in the user's workspace the size of one track. PACK tries to find space between where arrays and strings are stored in the workspace. If there is room, the command is executed. If not, an "OM ERROR" is displayed. I suggest that if you incorporate PACK in a program you do a FRE(X) immediately before invoking PACK. This makes the maximum amount of memory available. You shouldn't have any problems with PRE(X) causing a system crash here since utility programs use few string arrays, yet 24K systems may well need the extra RAM. \* CAUTION \* PACK can cause disastrous results







if a disk error occurs during the packing process. The headaches caused by a bad disk are never worth the price of a new disk.

# VIEW

VIEW displays the values of all non-subscripted variables. The variable names are marked for integer or string types. types. within Control characters control characters within strings are displayed with a caret ("^") followed by the letter of the control character. That is "^C" would be <CTRL>'C'.

CALL executes the machine code routine located at the value of the expression following

# WAIT addr.,expl (,exp2)

This is the standard WAIT command from BASIC under

As you can see, a large portion of Books is dedicated to making disk file management commands a part of the language so that they're "on-line" instead of requiring special utility programs. The idea here is to make the system work for you instead of against you. For example, how many times have you, in the midst of developing a program, after adding several lines of code to a program, tried to use DISK!"PU FNAME" only to get the dreaded "ERR #D ERROR"? Unless you were wise enough to maintain a special disk containing a scratch file, you would be really stuck in this situation. You not only have no place to put the latest version of your program, you've also managed to trash the original file as well. Using SAVE, KILL, and PACK will totally alleviate these problems. SAVE also does the track allocation check before attempting to write the file to disk to prevent it from ever trashing the original file.

While we're on the subject of errors.

While we're on the subject while we're on the subject of errors, it should be noted that Hooks uses BASIC's error handling wherever possible. This allows use of the TRAP command under OS-65D V3.3. Where there are no proper error messages in BASIC, Hooks provides it's own.

Hooks also makes changes the language itself. The big-gest change is that BASIC will understand hexadecimal

numbers in numeric expressions. Thus "DIRBUF=\$2E79" and "POKE \$8000,2" are now legal. This does not apply to INPUTS or DATA statements. INPUTS or DATA statements. BASIC can now print in either HEX or decimal as well. To have a value printed in HEX, preceed the expression with "\$,". "PRINT \$,11897" would display "\$2E79". This function is fully compatible with 3.3's cursor addressing, but not PRINT USING since only integer values are printed. 3.3's cursor addressing, but not PRINT USING since only integer values are printed. There is a difference between the HEX inputs and outputs in accuracy. Only 16-bit inputs are legal, but the output has a 32-bit accuracy.

With Books installed, GOTO's and GOSUB's will accept either and GOSUB'S Will accept either line numbers or variables. This can make your programs more readable, but you'll have to remember to change the values of the variables if you RSEQ your program. This function is NOT available in ONxGOTO's or ONxGOSUB's.

Finally, BASIC will now allow a limited IF...THEN...ELSE. Books does this by putting the keyword "REM" to an extra use. If an IF statement is evaluated as TRUE, all proceeds normally. But if the statement is FALSE, the line containing the IF statement is ment is FALSE, the line containing the IF statement is scanned for a "REM". If a "REM" is found, it is treated as a GOTO and control is passed to the line number (or variable!) following the REM. If no REM is found, control is passed to the next statement. For example, consider "100 IF X=2 THEN Y=3:REM 200". Under Hooks, if X does indeed equal 2, Y is set to 3 and control is passed to the next line in the program. But if X does not equal 2, control would be is passed to the next line in the program. But if X does not equal 2, control would be passed to line number 200. This allows a truer expression of the programmer's intentions sometimes than the normal IF statement would allow. This does mean that you'll have to be careful where you put your REM's though.

I expect that a lot of people will wonder if this code can be run under OS-65D V3.2. The answer is yes. You will have to make 3 changes to the source code for 3.2. The first is to change the label "CASECK" so that it points to an "RTS". CASECK is the routine in 3.3 that converts lower case alphabetic characters to upper case. The second change is to the label "SRCSTR". On mini-floppies this address should be charged expect that a lot of people "SRCSTR". On mini-floppies this address should be changed

Article continued on page 14

# DISK DRIVE RECONDITIONING

# WINCHESTER DRIVES

FLAT RATE CLEAN ROOM SERVICE. (parts & labor included) Shugart SA1002 5meg \$390.00 Shugart SA1004 10meg \$450.00

# FLOPPY DRIVE FLAT RATES Parts & Labor Included (Missing parts extra)

8" Double Sided Stemans 8" Single Sided Slemens 8" Doubte Sided Remex \$225.00 8" Single Sided Shugart 8" Double Sided Shugart \$190,00 514 M.P.L Single Sided 514 M.P.L Double Sided

# ONE WEEK TURN AROUND TYPICAL

- Il be notified of —

  1. The date we received your drive.
  2. Any delays & estimated completion date.
  3. Date drive was shipped from our plant.
  4. Repairs performed on your drive.
  5. Parts used (#and description).

90 day warranty — Write or call for detailed brochure We sell emergency parts Phone: (417) 485-2501



FESSENDEN COMPUTERS
116 N. 3RD STREET
OZARK, MO 65721

"Computer Business Software

"CBS"

· INTEGRATED Business system .

# – FEATURING –

- Accounts Receivable
- Inventory Control
- Order Entry/Invoicing
- · Accounts Payable
- General Ledger
- Payroll

# • BUSI-CALC

"An electronic worksheet"

# - FEATURING

- Local and General Formatting
- Replication
- Variable Column Widths
- EditingInsertion/Deletion
- of Rows and Columns Protected Entries
- · Help Screen
- Flexible Printing
- Complete User Manual



Madolyn()Sioux Fails, SD 5 **1-800-842-9858** 

### TIME & TASK PLANNER A REVIEW

PERK(65)
By: Edward T. Gieske, Jr.

Time & Task Planner's name either tells you a lot or very little - depending upon where you come from. Those who are already in the habit of keeping diaries, appointment books and the like will have a good idea, but I doubt that any would have guessed the completeness of this work. John Huntley (Mr. Gander Software) sure was organized when he put this one together. One of the beauties is that, although the programming is very complex, the user is presented with a very simple and straightforward program to "use". Remember, TTP is out of the same mold as the Financial Planner reviewed in the May issue of PEEK(65).

Very nice, but what's it all about? A better mouse trap? Yes! Start off with the premise that you are not as organized as you ought to be. Run down to the stationers and pick out a desk calendar that comes close to your usual schedule. Put up with its lack of room to make all the entries you need, not to mention, searching the streets for another one like it for next year. Scribble little notes on little scraps of paper about the things you must do - next week, month, or year (forget that - the paper will be gone long before then). Rewrite the list again and again trying to put the important items at the top. Oops! Where did that note go about the estimated taxes due?

Well, you get the picture. Whether its a doctor's daily appointment schedule, Johnny's birthday or the date of your annual check-up, chances are you won't have the information at hand when you need it, much less know that the 18th falls on a Sunday in 1984 when you had planned the meeting for next summer with your stock broker.

Now let's look at it in a positive attitude. TTP can give you the "Winner's Edge". With TTP you will have the opportunity to define and set wour goals and thus give you a clear and immediate challenge and the ability to track your performance. Once organized, it is amazing how much more you can get done in a day. In fact, just recording your goals helps to define and clarify them. The priority as-

signment makes sure that things are done in the proper order. Just try it! You will either get more done or have time left over for things you never had time for before. In any case, it will be worth many times the cost of the software.

That may sound like a dream, but it's fact. If you don't believe it, just take Gander up on their free trial offer. What have you got to lose? Nothing and everything, depending upon how you look at it.

Let's just assume that you have taken Gander up on their offer, and the package has arrived. Installation is routine using INSTAL for hard disks and floppy disk is almost "plug-n-go". For those who suffer compuphobia, there is a 38 page manual that approaches the ideal model. It's clear and concise with all of the user instructions at the beginning and sufficient technical stuff at the back. It is liberally sprinkled with screen printouts and also loaded with practical hints not only for the operation of the system, but also for practical use of its output.

The primary menu is broken down into five major areas.

- 1. An Appointment Scheduler. This looks just like an appointment book on your screen or paper. The difference is that you set the times (18 slots in all) on the schedule to your convenience not the publishers. On top of that, your files will keep 60 days worth on line at all times.
- 2. A "To Do List". This is your reminder; up to 60 items to which you assign a date and a priority number. The list is sorted by either date or priority.
- 3. A "Future Planning List". Again 60 items, sorted by date. This is primarily for long range planning beyond the "To Do" list. The nice part of this list is that, because both Julian and Georgian calendars are used, it always reports the number of days till the appointment, or for slow pokes, the number of days overdue.
- 4. A transfer program. Now we are getting to the good stuff! This ditty works its way through either the To Do or Future Lists and allows you to "post" it into a time slot for any day's Appointment Sche-

dule. The screening makes it all so easy and obvious that it is hardly worth reading the manual - but do it anyway.

5. A Calendar Program. Julian calendar to the rescue! A calendar for any month or year from 1910 - 2399. A whole year on a page or one of those nice block planning calendars for one month. Just tell it Screen or Printer.

With the user in mind, there are two means of getting into the above areas of the system. The normal route is through the Master Scheduler which allows one into all phases of a users lists. But, if you want to add something to a group of users, then you may enter either the To Do or Future areas directly from the menu and jump from person to person. Redundancy, yes, but it is convenient.

Lastly, there is a Print Utilities menu selection. This allows direct printing of just about anything in the system, either one day or a range of days and very importantly, prints Work Sheets on which to make notes as things come up.

What makes it all so nice is the beautiful screening - everything is there - all the ESC sequences listed that let you back up a question, get out, ignore last entry, etc. If that feature is not available at that point, it won't appear on the screen. The Edit, Add, Delete like the ESCs are all entered in a command description line at the bottom and are predictable and put the cursor where expected. Full use is made of the required OS65-U vl.43+ operating system. Most non-ANSI terminals are supported and provisions are provided for setting up most others. The printer selected shows up on the screen every time you enter the system. If you change it, it's stored and will be there the next time you enter the system. Just for the record, it is DMS compatible and will keep schedules for five named people. Need more than five, just run a duplicate system.

All the Utilities are here too. The sub-menu for System Maintenance allows simple selection for back-up disk initialization and file creation, down loading and up loading. You can back up just your files or everyone's. Of course, there's an Initialize the System so that you can set





things back to square one when you have finished playing and are ready to get down to business.

That about covers the things that you might expect to find in this kind of package, but there's more — from the programmers point of view, much more. Try a few of these on for size and keep in mind the ease of use. You have lists of 60 items and only 15 to a screen. Hence, forward or backward screening to your choice of screen not to mention automatic switch when you run off the bottom. A (L) ocate function that presents the right page with the cursor on the item. (A) dd an item, tells it the date, and the day of the week appears right next to it. Change the date during an (E) dit and the day of the week changes as does the number of days to the event in the Future list. The (S) ort always keeps things pushed up to the front of the file, so there is no need of a delete and repack routine. It takes a few seconds to do its thing, so a flag is set to prohibit an already sorted file from resorting.

I could go on, but by now you should have the picture. This is indeed a well thought out

system. They haven't missed a trick, but most importantly, it is simple to use. Read the manual and within 15 minutes you will be doing meaningful work.

Gripes! There was one question on the screen that I felt should have been prepacked by INPS and I like my Exits to go back to the base system directory rather than BEXEC\*. Programmers seem to prefer BEXEC\* but clutz users see that as one more step to get to SYSDIR. Then too, not all machines are set up with sub systems.

When it gets right down to it, this is the kind of program that should be on every machine capable of running OS-U. There isn't one of us who could not profit from TTP. When I look out in the market place and see all of the "bundled" software that comes with machines these days, it seems that someone missed the boat by not having a TTP in the stable. But now OSI dealers can be one up.

I almost forgot! What about support? I don't think that you will need any! The whole idea is to make the package plug and go. I cannot say that I did not talk to Mr.

EPROM POWER SUPPLY

12-5=26

Provides 26 volts from available +12 and -5 volts Sufficient to drive programmer for one Eprom

Bare board \$2.50
Bag of parts \$3.50
Post & Bandling \$3.00

Md. residents add 5% tax. Send U.S. dollars (drawn on a U.S. Bank to:

PEEK (65) P.O. BOX 347 Owings Mills, Md. 21117 (301) 363-3268

Euntley about the preliminary versions. I did, and found help came fast and to the point. At this point, John's calls are going to be for orders, not help.



From Gander Software

The Ultimate Personal Planner

# TIME & TASK PLANNER

30 DAY FREE TRIAL - IF NOT SATISFIED, FULL REFUND UPON RETURN

- "Daily Appointment Schedule"
- Work Sheets for all Aspects
- "Future Planning List" sorted
- Year & Month Printed Calendar
- "To Do List" by rank or date
- Transfers to Daily Schedule

A SIMPLE BUT POWERFUL TOOL FOR SUCCESS

Put the two most effective success techniques to work for you — every day of every year. Just five to ten minutes a day allows your mind and dreams to take charge of your life.

Set Your Goals: To reach a goal, you have to know where you are going. Just enter your goals or future appointments and let your computer remind you.

Set Your Priorities: Success depends upon doing first things first. Assign priorities (1-99) to your "To Do" list, let the computer keep them ranked by date or priority, and then get to work. When the time comes, the computer will help you transfer items to your choice of time on the daily Appointment Scheduler.

Teahnicalities - Appointment Scheduler: 18 time slots per day (you define) for 60 days. To Do List 60 items ranked by date or priority. Future Planning: 80 long range Items, date sorted; days to event or days overdue. Transfer to Scheduler: just tell it the date and time. Printed Calendars: Year on a page and one month box planning; any month, any year. System uses both Julian and Georgian calendars to handle dates from 1910-2399 and produce day of the week. Screen and menu driven; DMS Keybase compatible files. Detailed 38 page manual. Simple installation: FD to Multi HD. Files for 6 users=5.400 appointments. Unlimited Warranty.

HARDWARE: 48K OSI, 8" floppy or hard disk, serial terminal system, OS-66U v. 1.3 or later.

FEATURES: package allows configuration to ANSI standard and almost all non-ANSI terminals, AND user specification of printer port.

PRICE: \$150.00 (User Manual, \$25.00, credited toward TTP purchase). Michigan residents add 4% sales tax.

DEALERS: Your Inquiries are invited. This program should be on every 65U machine, including your own. At dealer prices, you could bundle this superior package as a sales incentive.

GANDER SOFTWARE

3223 Bross Road "The Ponds" Hastings, MI 49058



to \$3279 and on 8" systems it's \$3179. The last change is that you'll have to delete the line in "BCODE" that says "STX CRSCLR" since there is no cursor color in 3.2. The BEXEC\* program that installs Hooks automatically handles the only other change needed for 3.2.

If you want to experiment with adding your own keywords to BASIC, the Hooks source code will adjust automatically. First add the keyword itself to the CMDTBL and then add the label of the code that handles your keyword to the tables JTBL and JTBH like the others are listed. Just make sure

that you add your keyword and addresses AFTER the current ones. You may find that you'll have to change the origin address of the source code if you try this, but there is still a small amount of memory in the space currently allotted for Hooks. The main thing is to enjoy!



## LISTING 1

# by:Rick Trethewey

10 ; HOOKS INTO BASIC U	660	PNIL	=\$E1	Z-PAGE	POINTER LSB	
20 ; REV 1.8 9/23,	/83	670	PATTA	=SE2	2-DAGE	DOTABLED WCB
30 ;						
40 *≈\$B600		690	ADRIL	=\$FE	DISK R	/W ADDRESS LSB
50 ;		700	ADRH	=\$PP	DISK R	/W ADDRESS MSB
60; BASIC EXTERNAL	£	710	MAXMEM	=\$2300	HIGHES	t available ram page
70 ;		720	SECT	=\$265E	DISK R	/W SECTOR
80 POKER =\$19 2-PAG	E SCRAICH LOCATION	730	PAGES	=\$265F	# PAGE	s read or to write
90 VARIAB =\$/A SI'ART	OF ADDAY MADE	740	ADRLX	=\$2660	DISK R	/W ADDRESS LSB (NON-
100 ARRIAD = 5/C SIARI	OF ARRAY TABLE	750	AUKUX	-63663 -5700T	DIEK K	/W ADDRESS MSB VOLITILE)
130 STOCON -\$20 STADT	C VECTOR & ANTHORNO	770	TANGER	-62663	ע שמינים	AM DIMENT TWOCK #
130 EXT.THE =\$86 CHRE	OF LIME & REING EXECUTED	780	SEEKA	-92003 -82646	MOVE D	BLANG UNDOKAN
140 COTOTE =\$88 "COTO	Ou DOKEN	790	SEEK	=\$26BC	MOVE D	RIVE TO BCD TRK. # IN ACC.
150 REMIK =\$8E "REM"	TOKEN	800	LOAD	=\$2754	LOAD D	RIVE HEAD
160 VARNAM =\$92 CURRE	NT VARIABLE NAME	810	DNLOAD	=\$2761	UNLOAD	DRIVE HEAD
170 VARPNT =\$94 POINT	ER TO VARIABLE	820	SAVEX	=\$27D7	DISK W	RITE FROM ADRLX/ADRHX
180 FORPNT =\$96 MULTI	-DSE POINTER	830	FIND	=\$28C4	FIND S	ECTOR ON TRACK
190 TOIX = \$9D "TO"	TOKEN	840	CALLX	=\$295D	DISK R	EAD TO ADRLX/ADREX
200 THENTK = SAO "THEN	I TOKEN	850	DUMRED	=\$2998	READ D	isk - Thron away contents
210 MOLUK =\$A5 "*" TO	OREN	860	SELECT	=\$2906	SELECT	DISK DRIVE
220 EQUATE SAB "=" T	UKEN	870	ERHOR	=\$2A4B	65D ER	ROR REPORT ROTTINE
240 BYCEAD ~CYE E B	NACTIMITY NEXTS EASTMENTS	200	aspir	=\$ZADE	TUAOKR	ASSEMBLER/EDITOR
250 FACET -SAE F D	ACCIM MCR	000	ENI ENI	-63D37	TWANKE	EXITADED MONITOR
250 FACHI -SRO F P	ACCIM NIMSR	900	ETT.GM	=\$ZDM/ =\$7RM	DOND S	SAIDAE EILE
270 FACMED =\$B1 F.P.	ACCTM_ NLSB	920	SRCSTZ	=\$2BE9	SCHREE	FITE STAF IN PAGES
280 FACLO =\$B2 F.P.	ACCUM. LSB	930	CRLF	=\$2D6A	EXECUI	E (CR>CLF)
290 FACEGN =\$B3 F.P.	ACCIM. SIGN	940	SAVEM	=\$2C28	SECTOR	WRITE ROUTINE
300 CHRGET =\$CO PETCH	NEXT CHAR. IN PROGRAM	950	TINO	=\$2CEC	FEICH	COMMAND BYTE
310 CHRGOT =\$C6 RE-FE	FICH CHAR. AT CURRENT PIR.	960	SWAP	=\$2CF7	PAGE 0	/1 SWAP ROUTINE
320 TXTPIR =\$C7 POINT	TER TO PROGRAM TEXT	970	PRBYTE	=\$2D92	PRINT	ACC. CONTENTS (NUMERIC)
330 OMERR =\$044C OUT O	OF MEMORY ERROR	980	FNDNUM	=\$2DA6	PUT/IC	DAD TRACK FINDER
340 TYPERR =\$0462 DISPL	E SCRATCH LOCATION OF VARIABLE TABLE OF ARRAY TABLE OF ARRAY & VARIABLES OF STRING ARRAYS ANT LINE # BEING EXECUTED OF TOKEN ON TOKEN ON TOKEN ON TOKEN ON TOKEN ON TOKEN ON TOKEN OKEN OKEN OKEN OKEN OKEN OKEN OKEN	990	FNDNAM	=\$2DBE	FILE N	IAME LOOK-UP
350 GOID =\$UBA6 CODE	FOR "GOIO"	T000	DIKIRK	=\$2DC4	RCD DI	RECIORY TRACK NUMBER
320 DEM 2003C CODE	E IXIPIR IU IXIPIRTI	1020 1010	COLCH	=\$2343 =\$2573	EKTIMI.	ACC. CONTENTS
380 T.TATERY =\$095C COMPR	PER ASCIT INE TO BINARY	1030 1040	STROOT	-62E1E	EKTIVI.	MANU JEAN BUBBED
390 (780) =\$0A73 EXPOI	TITE (CR>CLF)	1040	DIBBUR	=\$2E78	DIBEAL	ADA BURKED ADA BURKED
400 BASPRT =\$0ACC OUTPU	FT STRING	1050	CRSCLR	=\$32E3	3.3 CI	RESOR BACKROUND COLOR
410 OUTDO -SOAEE OUTPU	IT A CHARACTER	1060	CASECK	=\$3A5F	CONVER	ET LOWER TO UPPER CASE
420 CHKTYP =\$0CBC CHECK	K FOR NUMERIC VARIABLE	1070	SRCSTR	=\$3A79	SOURCE	FILE START ADDRESS
430 CHKSTR =\$0CBE CHECK	K FOR STRING VARIABLE	1080;	;			
440 FRMEVL =\$0CCD EVALU	IATE EXPRESSION	1090		LDY #\$01		INIZ POINTER
450 CHROOM =\$0EL3 CHECK	FOR COMMA	1100		LDA (TXT	PIR),Y	LOOK 1 CHARACTER AHEAD
460 CBKCHR =\$0EL5 CHECK	FOR CHAR. IN ACC.	1110		CMP #MDL	TK	IS IT AN ASTERISK ?
470 SNERK =SUELE SYNTA	AX ERROR CUDE	112V		BINE CHK		VOOL BACK OR ONE
AND BALES -SALSE TYON	TON CALL ERROR	11 VV TT20		TDA (UNIO	v /cma	DESERVED TO COMPANY OF THE PROPERTY OF THE PRO
500 GWAYR =\$1218 GWE	A-V PATR TO P.P. ACCIM.	1150		נבגו מענו ובאו מענו	<u>טוג</u> בזע"ו•ז	MACK OFF LOWED CASE
510 FREFAC =\$1520 GET P	COINTER TO STRING	1160		CAD AIR	Lall.	CLEAD COLLEGE COM
520 GIBYTC =\$1615 DO "C	HRGET" & EVAL. EXP. < 256	1170		BERG COOLD	R	Canada Designata :
530 GETBYT =\$1618 EVAL.	EXPRESSION < 256	1180		CMP #'B	_	SET BACKROUND COLOR ?
540 GETVAR =\$1A9D FUT V	Variable in FACC.	1190		BEQ BOOD	E	
550 FLOAT =\$1B44 CONVE	ERT INTERGER TO F.P. TYPE	1200		CMP #'Q		SET SCREEN STATE ?
TOG GOTTIL -ATDOC COMAD	TO ALL TO THE TOTAL			2-0-	E	
570 ASCFP =\$1BEE CONVE		1220		CMP #'R	_	RESTORE NEW, LIST, & ^C ?
580 ASCII =\$1CEC CONVE		1230		BEQ RCOD	ß	POTENTI INTERCEDORUS S
590 PNUMBR =\$1CDC PRINT		1240 1250		CMP #'D	9	PRINT DIRECTORY ?
610 NONDAR =\$211C COPY 610 NONDAR =\$213A POINT		1260		BEQ DCOD CMP #'S	<b>ن</b> ا	SELECT DISK DRIVE ?
620 ;		1270		BEO SCOD	E	Present by program !
630; OS-65D EXTERNA		1280		CMP #'T	~	TOGGLE TRACE ?
640 ;		1290		BEQ TOOD	E	THE PARTY OF THE P
650 TMP = SEO TEMPO						Continued on page 16
•==						







# MICRO PRODUC

3702 N. Wells St. -Fort Wayne, Ind. 46808

# COMPUTER

MICRO-80 C	OMPUTER	į
------------	---------	---

Z-80A CPU with 4Mhz clock and CP/M 2.2 operating system. 64K low power static memory. Centronics parallel printer port. 3 serial ports. 4" cooling fan. Two 8" single or double sided floppy disk drives. IBM single density 3740 format for 243K or storage, double density format for 604K of storage. Double sided drives allow 1.2 meg on each drive. Satin finish extruded aluminum with vinyl woodgrain decorative finish. 8 slot backplane, 48 pin buss compatible with OSI boards.

MODEL 80-1200	\$2995
28" Single sided drives	
MODEL 80-2400	\$3495

28" Double sided drives

# **MICRO-65 COMPUTER**

6502 CPU with 2Mhz clock and DOS-65 operating system. 48K of low power static memory. 2 serial ports and 1 Centronics parallel port. 2 8" single or double sided drives. Satin finish extruded aluminum with vinyl woodgrain finish. 8 slot backplane, 48 pin buss compatible with OSI. Will run OSI 65D and 65U software. Includes Basic E/65 a compiled BASIC for

6502 CPU. MODEL 65-1	\$2995
2 8" Single sided drives MODEL 65-2	\$3495
28" Double sided drives BP-5808 Slot Backplane	\$ 47

# OSI 48 pin Buss compatible MEM-CM9 MEMORY/ FLOPPY CONTROLLER 24K memory/floppy controller card uses 2114 memory chips, 1 8K and

1 16K partition. Supports OSI type disk interface 24MEM-CM9 .....\$325 16MEM-CM9 .....\$260

Controller on assembled unit add.....\$ 90

BIO-1600 Bare IO card . . . . . . \$ 50 Supports 8K of memory, 2 16 bit parallel ports, 5 serial ports, with manual and Molex connectors.

# PRINTERS

# Okidata

ML82A, 120 cps, 10" . \$409 ML83A, 120 cps, 15" \$895 ML84 Parallel, 200 caps, 15". \$1150 C. loth

8510AP Prowriter, parallel ...\$419 120 cps, correspondence quality 8510APD Prowriter, serial . . . . \$585 F10-40PU Starwriter, parallel \$1319 Letter quality daisy wheel F10-40RU Starwriter, serial . \$1319

F10-55PU Printmaster .... \$1610 parallel, Letter quality daisy

F10-65RU Printmaster, serial \$1610 DISK DRIVES AND CABLES 8" Shugart SA801 .....\$385

single sided 8" Shugart SA851 double sided

FLC-66ft cable from D&N . or OSI disk controller to 8" drive 51/4" MPI B51 disk drive with . . \$450 cable, power supply and cabinet. Specify computer type. FLC-51/4 cable for connection . \$75

to 51/4 drive and D&N or OSI controller, with data separator and disk switch. Specify computer type

# HARDWARE : OSI COMPATIBLE

IO-CA10X Serial Printer Port . . \$125 Specify Device #3 or #8
IO-CA9 Parallel Printer Port ...\$150 **CMOS-MEM** 

64K CMOS static memory board, uses 6116 chips, 3 16K, 1 8K and 2 4K blocks, Partitionable for multiuser, OSI type disk controller, 2 IO mapped serial ports for use with D&N-80 CPU. Ideal way to upgrade from cassette to disk.

64R CMOS-MEM	490
48K CMOS-MEM	390
24K CMOS-MEM	250
16K CMOS-MEM	200
BARECMOS-MEM\$	
Controller add.\$	
210 mapped serial ports add. \$	125
and the second s	

on assembled memory board Z80-IO 2 IO mapped serial . . . . \$160 ports for use with D&N-80 CPU

FL470 Disk Controller . . . . Specify 51/4 or 8" drive

# STANDARD -CP/M FOR OSI

# D&N-80 CPU CARD

The D&N-80 CPU allows the owner of an OSI static memory computer to convert to Industrial Standard IBM 3740 single density disk format and CP/M operating system. Double density disk operation is also supported for 608K of storage on an 8" diskette. When used with a 5%" disk system 200K of storage is provided. Includes parallel printer and real time clock. Also available for polled keyboard and video systems. Compatible with C2, C3, C4 and 200 series OSi computers.

D&N-80- P · · · · · · · · · · · · · · · · · ·	\$349
CP/ M 2.2 ······	\$150
64K CMOS-MEM with D&	

HARD DISK DRIVER Allows D&N-80 CPU board to control OSI 40 or 80 meg hard disk unit. Will not destroy OSI files. Will also allow for a true 56K CP/M system. Specify 40 or 80 meg drive.

BUSSTRANSFER Allows for D&N-80 and OSI CPU to be in the computer at the same time. Toggle switch provides for alternate CPU operation.

DISKTRANSFER Utility program to transfer OSI CP/M format disk to IBM 3740 single density format. Will also transfer IBM to OSI format. SYSTEM HARDWARE REQUIREMENTS

D&N-80 CPU, D&N FL470 or OSI 470 controller, 48K memory at 0000-BFFF, 4K memory at D000-DFFF, two disk drive cables.

FORMATTRANSFER You supply software on 8" diskette D&N will transfer OSI CP/M format to IBM 3740 CP/M format. Can also transfer IBM 3740 CP/M format to OSI CP/M format. Original diskette





1200	TO SACTE	700	DOMESTIC STATE	TOTAL A TRANSPORTER OF THE PROPERTY.	2110				•
			PIRGET	IT'S A VARIABLE! EXECUTE	2110		7000	Offen Owner.	
1310			FORPNT	REPLACED CODE IN BASIC	2120	ט	USK.	STROOT	DISPLAY MESSAGE
1320			FORPNT+1	GIVE VARIABLE'S ADDR.	2130		PELL	#coo	cory *', \$D, \$A, \$A, \$0
1330		RIS		IT'S A VARIABLE! EXECUTE REPLACED CODE IN BASIC GIVE VARIABLE'S ADDR. TO BASIC AND GO BACK SEE IF IT'S A NEW KEYWORD	2140			#\$00	INIZ
1340	; ~~~	~~	Oraton.	ACC TO TOLD & NOT THE YOUR	2120		27.1	FIFTH	INIZ ENTRY COUNTER
1350	CBK	JUL	CHKR	JUMP TO SELECT CODE  JUMP TO TRACE TOGGLE  LOAD A <sp> INIZ PAGE COUNTER  INIZ PAGE COUNTER  LOOP TO PAGE END  BUMP PAGE INDEX  LOOP TO PAGE END  BUMP ADDRESS MSB  DECREMENT PAGE COUNTER  LOOP 'TIL DONE  GET ORIGINAL MSB  RESTORE POINTER  THROW AWAY ASTERISK  AND THE NEXT CHARACTER TOO  CANCEL JSR TO HOOKS  AND GO BACK TO BASIC  &amp; CURSOR COLOR  THROW AWAY ASTERISK  EVALUATE FOLLOWING EXPRESSION  MAKE VALUE NEW CURSOR COLOR  PUT IN ACC.  INIZ  INIZ MEMORY PAGE COUNTER  CHANGE COLOR  BUMP FOINTER  LOOP TO PAGE END</sp>	2120		INY	COCO	+1 (=1)
1360	;		2022	TRO MO STEET CORP.	21/0		211	SECT	SET SECTOR #
13/0	SCODE	JMP	SCOD0	JUMP TO SELECT CODE	7180		JSR	DIKIN	READ DIRECTORY SECTOR
1380	;				2190		JSR	DI	DISPLAY CONTENTS
1390	TOODE	JMP	TCODO	JUMP TO TRACE TOGGLE	2200		INC	SECT	BUMP SECTOR #
1400	·				2210		JSR	DIRIN	READ AND FALL THROUGH
1410	; CLE	IR 54	O BLACK & W	HITE VIDEO	2220	D1	LDX	#\$00	INIZ DIRBUF INDEX
1420	;				2230		LDX	#\$00	INIZ ENTRY INDEX
1430	CCODE	LDA	<b>#</b> 1	LOAD A <sp></sp>	2240	D2	LDA	DIRBUF, Y	FETCH CHARACTER FROM DIRBUF
1440		$\mathbf{L}\mathbf{D}\mathbf{X}$	#\$08	INIZ PAGE COUNTER	2250		CMP	# • #	PETCH CHARACTER FROM DIRBUF ENPTY ENTRY?  NO! DISPLAY NAME! => D3  MAYBE, "#" IS 1ST CHARACTER?  YES! SKIP ENTRY! => D4  PRINT NAME CHARACTER  BUMP DIRBUF INDEX  BUMP ENTRY INDEX  BUMP ENTRY INDEX  PRINT ENTIRE NAME YET?  NO! LOOP! => D2  YES! DISPLAY TRACK RANGE!  PUT DIRBUF INDEX IN ACC.
1450		<b>LDY</b>	#\$00	INIZ PAGE INDEX	2260		BNE	D3	NO! DISPLAY NAME! ==> D3
1460	$\alpha$	STA	\$D000,Y	CLEAR A CELL	2270		CPX	#\$00	MAYBE, "#" IS 1ST CHARACTER?
1470		IMA		BUMP PAGE INDEX	2280		BEQ	D4	YES! SKIP ENTRY! ==> D4
1480		BNE	a	LOOP TO PAGE END	2290	D3	JSR	CUTCH	PRINT NAME CHARACTER
1490		INC	C1+2	BUMP ADDRESS MSB	2300		INY		BUMP DIRBUF INDEX
1500		DEX		DECREMENT PAGE COUNTER	2310		<b>IXX</b>		BUMP ENTRY INDEX
1510		BNE	Cl	LOOP 'TIL DONE	2320		CPX	#\$06	PRINT ENTIRE NAME YET?
1520		LDA	#\$D0	GET ORIGINAL MSB	2330		BNE	D2	NO! LOOP! => D2
1530		STA	Cl+2	RESTORE POINTER	2340		JSR	TKOUT	YES! DISPLAY TRACK RANGE!
15401	PDATE	JSR	CHRGET	THROW AWAY ASTERISK	2350	D4	TYA		PUT DIRBUF INDEX IN ACC.
1550		JSR	CHRGET	AND THE NEXT CHARACTER TOO	2360		AND	#\$F8	MASK TO 8'S
1560	IJP7	PLA		CANCEL JSR TO BOOKS	2370		CLC		
1570		PLA		destinate and its require	2380		ADC	#\$08	ADD ENTRY LENGTE
7 58N		PTC		AND CO BACK TO BASIC	2390		TAY	-100	PUT RESULT BACK IN Y
1500		KZS		AND GO BALK TO BASIC	2400		BNE	D2-2	LOOP TO BUFFER END!
1600	, टाइसा (	<b>3</b> 77 (1)	DIAMONTAIN C	CURCOR COLOR	2410		T.DA	SEX all	DONE! FEICH CURRENT SECTOR #
1610	, DEL (		DACKROUND	& CURSON COLON	2/20		CMD	4¢02	DONE BOTH?
1010	2000		Annorm:	TODAY MAN NAMED OF	2420		DATE	# 402	NOVE BOTTLE
1620	BODDE	JSR	CHRGET	THROW AWAY ASTERISK	2430		D1412	CDT D	NOI QUIT!
1630		JSR	GIBYIC	EVALUATE FOLLOWING EXPRESSION	2440	_	שניונ	CKTL	YES! DO CLEAN UP & QUIT
1640		STX	CRSCLR	MAKE VALUE NEW CURSOR COLOR	2430	mrzoram.	CT 17		
1650		TXA		PUT IN ACC.	2400	TOOT	TXA		PUT DIRBUF INDEX IN ACC.
1660		LDY	<b>#\$0</b> 0	INIZ	24/0		PBA		SAVE ON STACK
1670		LDX	#\$08	INIZ MEMORY PAGE COUNTER	2480		LDA	# "	GET A <sp></sp>
1680	BC01	STA	\$E000,Y	CHANGE COLOR	2490		JSR	OUTCH	PRINT FOR SEPARATION
1690		INY		BUMP POINTER	2500		LDA	DIRBUF,Y	FETCH START TRACK #
1700		BNE	BCOI	LOOP TO PAGE END	2510		JSR	PRBYTE	PRINT IT
1710		INC	BCO1+2	BUMP PAGE ADDRESS ABOVE	2520		LDA	#!-	GET "-"
1720		DEX		DECREMENT PAGE COUNT	2530		JSR	OUTCH	PRINT IT
1730		BNE	BCO1	LOOP 'TIL DONE	2540		PLA		RETRIEVE DIRBUF INDEX
1740		LDA	#SEO	FEICH ORIGINAL PAGE ADDRESS	2550		PHA		PUT IT BACK FOR LATER
1750		STA	BCO1+2	RESTORE IT ABOVE	2560		TAY		PUT INDEX IN Y AGAIN
1760		BME	וקוו	AND EXTT TO BASIC	2570		LDA	DIRBUF+1,Y	FETCH ENDING TRACK #
1770	•	247	0.2		2580		JSR	PRBYTE	PRINT IT
	, दास्य	340 1	DISPLAY STAT	TE.	2590		LDA	Pifth	FETCH # ON THIS LINE
1790	, war .	,40 ,	JIDIDAY DIA	INIZ INIZ MEMORY PAGE COUNTER CHANGE COLOR BUMP POINTER LOOP TO PAGE END BUMP PAGE ADDRESS ABOVE DECREMENT PAGE COUNT LOOP 'TIL DONE FEICH CRIGINAL PAGE ADDRESS RESTORE IT ABOVE AND EXIT TO BASIC  THROW AWAY ASTERISK EVALUATE FOLLOWING EXPRESSION SET SCREEN STATE WITH RESULT	2600		CMP		DONE 4 YET?
	, ()	TED	CHRGET	TUDEM MADY ACTIVITIES	2610		BEO	TROU2	YES! => TKOU2
1810	QCCDE	JOIN GPT.	CITEVITY	EVALUATE FOLLOWING EXPRESSION	2620		JSR	STROUT	NOI PRINT 2 SPACES
1820		COLV	\$DE00	SET SCREEN STATE WITH RESULT	2630		BYI	E'',\$00	
1830		TMD	UP1	AND GO BACK TO BASIC	2640		TNC	FIFTH	BUMP # ON THIS LINE
1840		QPL-	OFI	HID GO BACK TO BASIC		TKOUL			RETRIEVE DIRBUF INDEX
		NDD	Particular No. T.C.	ויין איזיין איזיין איזיין איזיין איזיין איזיין	2660		TAY		PUT IT BACK IN Y
		JKE.	NEW , LIS	C", AND <ctrl>'C'</ctrl>	2670		RTS		AND GO BACK
1860		T 538	A76	THECE NUMBERS OFFICE		TROU2		CRI.F	DO CLEAN-UP <cr><lf></lf></cr>
	RCODE			THESE NUMBERS SHOULD	2690	~~~		#\$00	INIZ
1880			741	*VERY* FAMILIAR	2700			FIFTH	CLEAR LINE COUNTER
1890			#78		2710			TKOU1	AND LOOP!
1900			750		2720		DLAZ	TVCOT	AND LOOF!
1910			#173				TCD	CHRGET	OTTOCK NAME A CONCENT COM
1920			2073		2740	30000		CHRGET	THROW AWAY ASTERISK
1930		BNE	UPDATE						FETCH NEXT CHARACTER
1940	;				2750			FRMEVL	EVALUATE EXPRESSION
1950	; D* (	CAN )	Be made avai	LABLE FROM OS-65D BY	2760			CHKSTR	MAKE SURE IT'S A STRING
			G 65D AS FOI		2770			FREFAC	FIND STRING
1970	; \$2E	$= \alpha$	\$2A \$2E3E	= \$B7  \$2E3F = \$B6	2780			SCOD1+1	
1980					2790		SIY	SCOD1+2	
1990	DCODE	JSR	SWAP	* DOS CONTEXT *	2800			#\$01	CHECK LENGTH
2000		JSR	D	* DOS CONTEXT * PRINT DIRECTORY * LANGUAGE CONTEXT * AND GO BACK TO BASIC	2810			SCOD3	BAD! ERROR!
2010			SWAP	* LANGUAGE CONTEXT *		SCOD1			MODIFIED CODE!
2020			UPDATE	AND GO BACK TO BASIC	2830			CASECK	MAKE IT ALL-CAPS
2030			<del></del>		2840		CMP		CHECK FOR LEGAL DRIVE #
		LDZ	#DLRBUF	GET DIRBUF LSB	2850		BCC	SCOD3	
2050			ADRLX	GIVE IT TO 65D	2860		CMP	#'D+1	
2050			#DIRBUF/256	<b>;</b>	2870		BCS	SCOD3	
2070				HANDLE MED TOO	2880			TEMP	SAVE IT
			ADREX	GET BCD DIRECTORY TRACK #	2890			SWAP	* DOS CONTEXT *
2080			DIRTRK		2900			TEMP	PEICH DRIVE REQUEST
2090			SEEK	MOVE HEAD TO TRACK					
2100		NUL	READ+3	READ DIRECTORY SECTOR					Listing continued





2910	and #\$of	MASK TO LOW NYBBLE	3320	CMP #JTBH-JTB	t.+2
2920	JSR SELECT	SELECT DRIVE	3330	BNE NXTI	NO! LOOP! => NECTL
2930	BCS SCOD2	DRIVE NOT READY? => SCOD2	3340	JMP BACK	YES! RETURN TO BASIC
2940	JSR BOMEO	BOME DRIVE	3350 NXII	LDA CMOTBL,X	CHECK REYWORD CHARACTER
	JSR SWAP	* LANGUAGE CONTEXT *	3360	BEO NXT2	at end of word? => nxt2
2960	JMP UPL	AND QUIT	3370	INX	NO! BUMP COMMAND INDEX
2970 SCOD2		GET 'DRIVE NOT READY'	3380	BNE NXII	AND LOOP!
2980	JMP ERROR	DSE 65D'S ERROR ROUTINE & QUIT		INX	MOVE 1 PAST TERMINATOR
2990 SCOD3		ERROR!	3400	LDY #\$00	RESTORE TEXT INDEX
3000 ;	DATE L'ESTAT	LAWORS	3410	BEO CHKI	AND LOOP!
	ו זאר פולדו	E REYWORDS BELOW ARE	3420;	20g	
		CANNOT* BE USED AS VARIABLES	3430 XCMD	LDA COUNT	GET COMMAND #
3030 ;	TOPOSITO PRID		3440	CMP #\$02	ASM OR EM ?
	BYTE 'ASM' .O		3450	BCS RUNNER	NO! => RUNNER
3050	BYTE 'EM',0		3460	JSR SWAP	YES! * DOS CONTEXT *
3060	BYTE 'SAVE',0		3470	LDA #\$11	GET TXTBUF LENGTE
3070	.BYTE 'LOAD',0		3480	STA TINO+1	RESET FOR TINCH
3080	BYTE 'PACK'.0		3490	LDA COUNT	CHECK COMMAND
3090	.BYTE 'VIEW', O		3500	BEO GOASM	ASM? => GOASM
3100	BYTE 'CALL'.0		3510	JAP EM	EMI DO ITI
3110	BYTE 'KILL',0		3520;		2
3120	.BYTE 'MAKE',0			JMP ASMR	ASMI DO ITI
3130	BYTE 'RENAME'	•	3540;	UPA PADEAN	1311 20 21
3140	.BYTE 'WAIT' .0	,u	3550 RUNNE	ο πλα	PUT COMMAND # IN X
3140 3150			3560 ADMAL	DEX	-1
3160	BYTE 'FILE',0		3570	DEX	-1 -2
				LDA JIBL,X	FETCH ADDRESS LSB
3180 CHKR	TOV ACOO	INIZ CMD. TABLE INDEX INIZ CMD. COUNTER INIZ TEXT INDEX FEICH TEXT CHARACTER MAKE IT ALL CAPS COMPARE TO KEYWORD	3590	STA RUNL+1	SAVE IT
3190 Carr	TIN HOU	TATE CAD CONTRETE LADEX	3600	LDA JIBH.X	
3200	PIN COOMI.	INIZ CMD. COUNTER	3610	SIA RUNI.+2	SAVE IT TOO
3210 CBKI	ער למשוויים אולים אינים	TMIX ALEXA TIMBRY	3620 RUNIL	JMP SFFFF	
3220	TED CACIOCA	MAPP THE ALL CHICAGIES	3630 ;	OHE SERV	DO CONTRADO.
3230	CARD CARDECT	CARDIDE IN MERICOR	3640 JTBL	BYTE SAVIT-I	ODIT, PACKIT, VIEWIT
3240	BNE NXTCMD	AN MARKE TO KEXWORD	3650		IL, MAKER, RENAME, WAIT
3250	INX	COMPARE TO KEYWORD  NO MATCH? => NXTCMD  YES! BUMP COMMAND INDEX  BUMP TEXT INDEX  CHECK FOR KEYWORD END  YES! EXECUTE COMMAND! =>  NO! LOOP!	3660	BYTE FILGET	
3260	INA TAY	TEST BUILD CLARAND LINDEX	3670;	*DIAH LAMUA	
3270	LDA CMDIBL,X	DUNG TOXY TANDEY	3680 JTBH	BALL CALLERY	256 ,LODIT/256 ,PACKIT/256
3280	BEO XCMD	CORA PUR REXIMOND END	3690		/256,CALR/256,KILL/256
3290	BNE CBKI	NOI LOOP!	3700		256, RENAME/256, WAIT/256
3300NXTCMD		NO. BOOK:	3710	BYTE FILGET	
3310	LDA COUNT	BUMP COMMAND COUNTER FETCH IT	3720 ;	"DITE RINGEL	AU
3310	DIE CANINI	EBICH II	3120 1		Continued

# DBI ANNOUNCES ANOTHER FIRST FOR THE OSI\*\* MACHINE

THE SAME PEOPLE WHO BROUGHT YOU

THE REVOLUTIONARY DB-1 MULTIPROCESSING ENCHANCEMENT INTRODUCES THE

# DS-1 SCSI HOST ADAPTER

WITH BATTERY BACKED REAL TIME CLOCK, 100 YEAR DAY DATE CALENDAR AND 5K RAM

The DS-1 allows for many new disk technologies. For example, the IOMEGA $^\dagger$  Alpha 10, a 10 megabyte formatted removable disk, or the 5% inch Winchestors.

The combination of the **DS-1** and **Alpha 10**<sup>1</sup> are a perfect upgrade for all OSI "machines using the 48 pin bus and OS-65U<sup>††</sup> Operating Systems. This combination can also be used for additional storage and backup on hard disk models.

For Further Information Contact:

1 ALPHA 1U AND IUMEGA ARE TRADEMARKS OF IOMEGA CORP 17 USI AND US-88U ARE TRADEMARKS OF OHIO SCIENTIFIC. INC



p.o. box 7276 denver, co 80207 inc. (303) 364-6987 Dealer Inquires Invited

3730	LODIT	JSR	ADD4	BUMP TXTPIR PAST "LOAD" GIVE FILE NAME/# TO 65D LOAD IT INTO WORKSPACE RESTORE LANGUAGE CONTEXT COPY OUT FILE SPECS TO BASIC GO BACK TO BASIC
3740		JSR	NONUMR	GIVE FILE NAME/# TO 65D
3750		JSR	LOADER	LOAD IT INTO WORKSPACE
3760		JSR	SWAP	RESTORE LANGUAGE CONTEXT
3770		JSR	\$2271	COPY OUT FILE SPECS TO BASIC
3780		JMP	00143	GO BACK TO BASIC
3000	ECUDAD 1	27.7	8001	GO BACK TO BASIC  INIZ  START WITH SECTOR 1  READ DIRECTORY SECTOR  MARK USED TRACKS  BUMP SECTOR #  READ SECTOR AND FALL THROUGH  INIZ DIRBUF INDEX  PETCH ENTRY 1ST CHARACTER  EMPTY ENTRY?  YES1 => SORTZ  NO, FUT INDEX IN ACC.  POINT TO STARTING TRACK #  PUT INDEX BACK IN Y  FETCH STARTING TRACK #  MAKE IT HEX  SAVE AS START TRACK  FETCH ENDING TRACK #  MAKE IT HEX  SAVE IT TOO  MARK TRACK RANGE "IN USE"  PUT DIRBUF INDEX IN ACC.  MASK TO 8'S  ADD ENTRY LEXSTH  PUT RESULT BACK IN Y  LOOP TO PAGE END  AND QUIT
3810	BUKI	STA	#9UL	CHADA PATHE CENTROD 1
3820		JER	DISIN	PEAD DIRECTORY CECTOR
3830		JSR	SORTO	MARK USED TRACKS
3840		INC	SECT	BUMP SECTOR #
3850		JSR	DIRIN	READ SECTOR AND FALL THROUGH
3860	SORTO	$\mathbf{r}_{\mathbf{D}\mathbf{X}}$	#\$00	INIZ DIRBUF INDEX
3870	SORT	LDA	DIRBUP, Y	FETCH ENTRY 1ST CHARACTER
3880		CMP	# *#	EMPTY ENTRY?
2830		Reñ	SORIZ	YES! ==> SORIZ
3910		CLC		NO, FUT INDEX IN ACC.
3920		ADC	#S06	POTENT TO STARTING TRACK #
3930		TAY	4700	PUT INDEX BACK IN Y
3940		LDA	DIRBOF, Y	FETCH STARTING TRACK #
3 <b>9</b> 50		JSR	BCDH	MAKE IT HEX
3960		STA	STIK	SAVE AS START TRACK
3970		LDA	DIRBUF+1,Y	FETCH ENDING TRACK #
3980		JSR	ECDH	MAKE IT HEX
3990		SIA	ENDIK	MADE GOACE DANCE CON DOOR
4030	CODOD)	NOW.	REDR	THE DEPOSIT TARGET IN USE
4020	SURIZ	TXM	ASTR .	MICK WO 8'S
4030		CLC.	"TLO	1401 10 0 0
4040		ADC	#\$08	ADD ENTRY LENGTH
4050		TAY		PUT RESULT BACK IN Y
4060		BNE	SORCI	LOOP TO PAGE END
4070	SORT3	RIS		AND QUIT
4080;	Baan	* ***	-	H. C. C. L. C.
47.00	DECEN	LUX	STIK	USE STIK AS ORIGIN
4110	RESKI	TINC	LTSI.'Y	MARK "LIST" ENTRY
4120		BED	STRING	VEST OUTTI-S SORTS
4130		אמנ		NO. RIMP TNDEX
4140		BNE	RESRI.	AND LOOP
41.50	;			USE STTK AS ORIGIN MARK "LIST" ENTRY MARKED ALL TRACKS? YES! QUIT!=> SORT3 NO, BUMP INDEX AND LOOP
4160F	ACKIT	JSR	ADD4	MOVE PAST "PACK" GET SOURCE FILE SIZE +2 FOR 8" MAXIMUM SAVE RESULT CHECK AVAILABLE RAM
4170		LDA	SRCSIZ	GET SOURCE FILE SIZE
4180		CLC		a as all manage
4190		ADC	#\$02	+2 FOR B" MAXIMUM
4200		STA	STRUIA	SAAE KESORI.
4220		SEC	DIGGEATI	CILCA AVAILABLE KAN
4230		SBC	STRFLG	
4240		CMP	ENDTAB+1	
4260		JMP	OMERR	NOT ENOUGH MEMORY!
4270	PACK1	STA	STRFLG	SAVE BUFFER ADDR. MSB
4280		JSR	SWAP	NOT ENOUGH MEMORY! SAVE BUFFER ADDR. MSB * DOS CONTEXT *
4230		JSK	PAKK	PACK DISKETTE
4300		าผน	OUL	RETURN TO BASIC
4310;	משמם: בשמם	TCD	CT DI CT	מ באם ווכבה שסארא ו זכש
4330	FEMA	JSR ART.	SORT	CLEAR USED TRACK LIST MARK USED TRACKS IN LIST
4340	GAP	LDA	EMIMIM+)	GET FLOPPY MAX. TRK. #
4350			BCDH	MAKE IT HEX
4360		TAY		PUT IN Y
4370		INA		+11
4380		STY		SAVE AS MAX.
4390			#\$00	INIZ
	GARS	LUA	LIST, Y	CHECK LIST
4410 4420		IMX REÖ	CAP4	CLEAR TRACK ? => NO! BUMP POINTER TO LIST
4430			MAXVAL	AT END OF DISK ?
4440		BNE	GAP3	NOI LOGP! => GAP3
4450		RTS	•	YES! OUTT! (DISK FULL)
4460	GAP4	STY	STGAP	SAVE 1ST CLEAR TK. #
4470	GAP5	LDA	LIST, Y	SAVE 1ST CLEAR TK. # FIND LENGTH OF "EMPTINESS"
4480		RIVE	GALTO	USED? => GAPO
4490		IMY		NOI BUMP POINTER!
4500		CTA	MAXVAL	AT END OF DISK?
4510 4520		RIS		NO! LOOP! => GAP5 YES! OUIT (DISK CLEAR TO END
マンムひ		$\alpha$		TIME COST CHEST CHEST IN DAIL

# LAST

FOR THE BEST DENVER BOARD UTILITIES AVAILABLE, CLOSE-OUT OF DOCUMENTED PACKAGES IN STOCK . . .

Professional OSI programmer (5 years developing specialized packages nationwide) — recently contracted to design operating system utilities for IBM PC.

NOW OFFERING to OSI end-users: complete system maintenance and applications utilities for OSI Deriver Board systems...

# QF BOSS:

Ties any applications package to all utilities.

# QF UTIL:

Copies, creates, deletes, edits, etc.

# **QF LOAD:**

- Assembly-language, report & key-file loader.
- With comparison testing.

# QF SORT:

- Assembly-language, fixed-length record sort.
- Fastest OSI sort on the market.
- No work or merge files required.

# COMPLETE PACKAGE INCLUDES ALL ABOVE PLUS:

- Package includes over 26 programs.
- Over 100 sample report and sort specifications.
- Access to all basic source code.
- All reports & sorts can be saved for re-use.
- Fully documented with 232-page manual.
- OSI/DMS compatible.

# PROVEN RELIABLE FOR OVER 3 YEARS!! ... Ask some of our delighted users:

DBI, Inc. (Denver, CO) 303/428-0222

Browning Publications (Atlanta, GA) 404/455-3430 Progressive Casualty Ins. (Cleveland, OH) 216/461-5000

Bethphage Mission (Axtell, NE) 308/743-2401 Union Credit Corporation (Albany, GA) 912/435-1381

SEND CHECK OR MONEY-ORDER TODAY! FULL MONEYBACK SATISFACTION GUARANTEED

**QUICK FILES** 

P. O. BOX 56552 ATLANTA, GA 30343

404/523-5229





RIS Listing continued YES! QUIT (DISK CLEAR TO END)

4530 GAP6 STY STBLK 4540 STY POINT SAVE AGAIN - PROTECTED 4550 TYA 4560 SEC 4570 SEC STGAP 4580 STA GAPLEN 4590 GAP7 INY 4600 LDA LIST, Y 4610 BEQ GAP8 4520 CPY MAXVAL 4630 BER GAP7 NO! LOOP! ⇒ GAP8 4640 GAP8 DEY 4650 STY ENBLR 4660 NOVE LDA STBLK 4670 STA TRAKX 4670 STA TRAKX 4700 STA LIST, Y 4680 LDA \$SDLK 4710 JER SEEKX 4720 JSR CNTS 4730 LDA \$FIFTH GET SECTOR ON TRACK 4720 JSR CNTS 4750 LDA \$901 INIZ 4750 LDA \$901 INIZ 4760 STA RAKX 4770 MOV2 JER READ 4780 LDA STGAP 4780 GET OPEN TRACK 4810 JSR SEEKX 4790 STA TRAKX 4810 JSR WRITE 4820 LDA SIGAP 4830 CMF FIFTH END OF TRACK 4810 JSR SEEKX 4810 JSR WRITE 4820 LDA SIGAP 4830 CMF FIFTH END OF TRACK 4810 JSR SEEKX 4810 JSR SEEKX 4810 JSR WRITE 4820 LDA SIGAP 4830 CMF FIFTH END OF TRACK 4810 JSR SEEKX 4810 JSR SEEKX 4810 JSR WRITE 4820 LDA SIGAP 4830 CMF FIFTH END OF TRACK 4810 JSR SEEKX 4810 JSR SEEKX 4820 LDA SECT 4830 CMF FIFTH END OF TRACK 4840 BEQ MOV3 4850 LDA STBLK 4870 STA TRAKX 4870 STA TRAKX 4890 JMP MOV2 4900 MOV3 LDY STGAP 4890 JNR MOV2 4910 LDA \$TBLK 4890 JNR MOV2 4900 MOV3 LDY STGAP 4910 LDA \$TBLK 4890 JNR MOV2 4900 MOV3 LDY STGAP 4910 LDA \$TBLK 4890 JNR MOV2 4900 MOV3 LDY STGAP 4910 LDA \$TBLK 4890 JNR MOV2 4900 MOV3 LDY STGAP 4910 LDA \$TBLK 4890 JNR MOV2 4900 MOV3 LDY STGAP 4910 LDA \$TBLK 4890 JNR MOV2 4900 MOV3 LDY STGAP 4910 LDA \$TBLK 4890 JNR MOV2 4900 MOV3 LDY STGAP 4910 LDA \$TBLK 4890 JNR MOV2 4900 MOV4 LDA \$\$DL 5000 STA SECT 5010 MOV4 LDA \$\$DL 5000 MOV5 LDA DIRBUF, Y FEICH DIR. ENTRY			
## ## ## ## ## ## ## ## ## ## ## ## ##	4530 GAP6	STY STRLK	SAVE FOOD OF GAP + 1
## A550   TYA   PUT IX. # IN ACC.   ## A560   SEC   ## A570   SEC   STGAP   SUBTRACT   1ST FREE IX. #   ## A580   STA   GAPLEN   SAVE LENGTH OF GAP   ## A680   LDA   LIST, Y   BUMP POINTER   ## A610   LDA   LIST, Y   BUMP POINTER   ## A620   CPY   MAXVAL   NO, END OF DISK?   ## A630   SEC   GAP8   EMPLY?   SACK   LEO   ## A640   GAP8   DEY   BACK   UP ONE   ## A650   STY   ENBLK   GET TRACK #   TO MOVE   ## A660   MOVE   LDA   SUBLK   GET TRACK #   TO MOVE   ## A670   STA   LIST, Y   SHOW TRACK   CLEAR   NOW   ## A680   TAY   SHOW   TRACK   CLEAR   NOW   ## A710   JER   SEERK   MOVE   HEAD TO TRACK   ## A720   JSR   CNTS   COUNT   SECTORS   ON TRACK   ## A730   LDA   FIFTH   GET   SECTOR   COUNT   ## A740   BED   MOV3   IF 0   MOV3   ## A750   LDA   \$501   INIZ   ## A750   LDA   \$501   INIZ   ## A770   MOV2   JSR   READ   READ   RECTOR   ## A770   ROV2   JSR   READ   READ   RECTOR   ## A780   LDA   STGAP   GET OPEN   TRACK   ## A800   JSR   SEETK   MOVE   HEAD TO TRACK   ## A810   JSR   WRITE   END OF TRACK   ## A820   LDA   STGAP   GET OPEN   TRACK   ## A830   CMP   FIFTH   END OF TRACK   ## A840   BED   MOV3   YES!   MOV3   ## A850   INC   SECT   RO, BUMP   SECTOR   ## A840   BED   MOV3   YES!   MOV4   ## A850   JSR   SEEKK   MOVE   HEAD TO TRACK   ## A840   JSR   SEEKK   MOVE   HEAD TO TRACK   ## A840   BED   MOV3   YES!   MOV3   ## A850   JSR   SEEKK   MOVE   HEAD TO TRACK   ## A860   LDA   STBLK   GET ORIGIN   TRACK   ## A870   STA   LIST, Y   SHOW   TRACK   NUSE   NOW   ## A930   LDA   STBLK   GET ORIGIN   TRACK   ## A940   CMP   STGAP   ROD   ROD   ## A940   CMP   STGAP   ROD   ROD   ## A940   CMP   STGAP   ROD   ROD   ## A940   CMP   STBLK   GET ORIGIN   TRACK   ## A950   BHO   MOV4   LDA   \$501   ROD   ## A960   MOV5   LDA   DIRBUF, Y   ## FIFTH	4540	STY POINT	SAVE AGAIN - PROTECTED
4550 SEC STGAP 4570 SEC STGAP 4580 SEC STGAP 4600 LDA LIST,Y 4600 LDA LIST,Y 4610 BEQ GAP8 4620 CPY MANVAL 4630 BEC GAP7 4640 GAP8 4650 SEY 4650 SEY 4650 SEY 4650 SEY 4660 MOVE LDA STBLK 4660 MOVE LDA STBLK 4670 SEA TRAKX 4680 TAY 4710 JER SEEKK 4720 JER ALIST,Y 4710 JER SEEKK 4720 JER CATS 4730 LDA FIFTE GET SECTOR COUNT 4740 BEQ MOV3 4750 LDA FIFTE GET SECTOR COUNT 4740 BEQ MOV3 4750 LDA STGAP 4760 SEA TRAKX 4770 MOV2 4780 LDA STGAP 4800 JER BEEKK 4810 JER WRITE 4820 LDA STGAP 4820 LDA STGAP 4830 COP FIFTH 4830 COP FIFTH 4830 COP FIFTH 4840 BEQ MOV3 4850 INC SECT 4850 LOA STBLK 4860 JER SEEKK 4870 SECT 4860 LOA STBLK 4870 SECT 4880 JER SEEKK 4890 MOV3 4850 INC SECT 4860 LOA STBLK 4870 SECT 4880 JER SEEKK 4890 MOV3 4850 INC SECT 4860 LOA STBLK 4870 SECT 4880 JER SEEKK 4890 MOV3 4850 INC SECT 4860 LOA STBLK 4870 SECT 4880 JER SEEKK 4890 MOV3 4850 INC SECT 4860 LOA STBLK 4870 SECT 4880 JER SEEKK 4890 MOV3 4850 INC SECT 4870 SECT 4880 JER SEEKK 4890 MOV3 4850 INC SECT 4870 SECT 4880 JER SEEKK 4890 MOV3 4850 INC SECT 4880 JER SEEKK 4890 MOV3 4850 INC SECT 4890 MOV4 4960 INC STGAP 4990 MOV4 4960 INC STGAP 4980 BNE MOVE 4990 MOV4 4960 INC STGAP 4900 INC STGAP 4900 INC STGAP 4900 IN	4550	TŸA	PIT TX. # TN ACC.
## ## ## ## ## ## ## ## ## ## ## ## ##	4560	SEC	102 MI
## S80	4570	SBC STGAP	SUBTRACT 1ST FREE TK. #
## ## ## ## ## ## ## ## ## ## ## ## ##	4580	STA CAPLEN	SAVE LENGTH OF GAP
4600 LDA LIST, Y CHECK LIST  4610 BEQ GAP8 EMPTY: => GAP8  4620 CPY MAXVAL  4630 BNE GAP7 NO, END OF DISK?  4640 GAP8 DEY BACK UP ONE  4650 STY ENBLE SAVE END OF BLOCK TE. #  4660 MOVE LDA STIBLE GET TRACK # TO MOVE  4650 TAY BACK UP ONE  4660 TAY BACK UP ONE  4660 TAY BACK UP ONE  4670 STA TRAEX GIVE IT TO 65D  4680 TAY PUT IN Y AS INDEX  4690 LDA #\$00 INIZ  4700 STA LIST, Y SHOW TRACK CLEAR NOW  4710 JER SEEKK MOVE HEAD TO TRACK  4720 JER CNTS COUNT SECTORS ON TRACK  4730 LDA FIFTH GET SECTOR COUNT  4750 LDA #\$01 INIZ  4760 STA SECT START AT SECTOR 1  4770 MOV2 JER READ  4780 LDA STGAP GET OPEN TRACK #  4790 STA TRAKX GIVE IT TO 65D  4800 JER SEEKK MOVE HEAD TO TRACK  4810 JER WRITE WRITE SECTOR  4810 JER WRITE WRITE SECTOR  4820 LDA SECT CHECK CURRENT SECTOR #  4830 CMP FIFTH END OF TRACK #  4840 BEQ MOV3 YES! => MOV3  4850 LDA STBLE GET ORIGIN TRACK #  4860 LDA STBLE GET ORIGIN TRACK #  4870 STA TRAKX GIVE TO 65D  4880 JER SEEKX MOVE HEAD TO TRACK  4890 JER SEEKX MOVE HEAD TO TRACK #  4870 STA TRAKX GIVE TO 65D  4880 JER SEEKX MOVE HEAD TO TRACK #  4890 MOV3 LDY STGAP GET ORIGIN TRACK #  4890 JER SEEKX MOVE HEAD TO TRACK #  4890 LDA \$TBLE GET DEST. TE. #  4910 LDA \$501 INIZ  4920 STA LIST, Y SHOW TRACK IN USE NOW  4930 LDA STBLE GET ORIGIN TE. # TOO  4940 CMP ENBLE AT END OF BLOCK?  4950 BIR MOVE AND LOOP! => MOVE  4960 BIR MOVE AND LOOP! => MOVE  4960 BIR MOVE AND LOOP! => MOVE  4970 INC STBLE BEMP ORIGIN TE. # TOO  4980 BIR MOVE AND LOOP! => MOVE  4990 MOV4 LDA #\$01 INIZ  5000 STA GAPLEN GET BLOCK LEACHH  5010 LDA GAPLEN GET BLOCK LEACHH  5020 JER BECD MAKE IT BCOI  5030 MOV9 JER DIRIN READ IN DIRECTORY SECTOR  5040 MOV9 JER DIRIN READ IN DIRECTORY SECTOR  5050 LDY #\$00 INIZ	4590 GAP7	TNY	BUMP POTNITIER
### ### ### ### ### ### ### ### ### ##	4600	TAPA TATET. Y	CHECK LIGHT
### ### ### ### ### ### ### ### ### ##	4610	BEO GAPS	EMPTY? => CADR
### ### #### #########################	4620	CDV MAYOTAT.	MU EMD OF DIERS
## ## ## ## ## ## ## ## ## ## ## ## ##	4630	BATE CAP?	NO LOOP - CAP
4650 MOVE LDA SIBLK GET TRACK # TO MOVE 4670 ST# TRAKX GIVE IT TO 65D 4680 TAY 4690 LDA #\$00 INIZ 4700 ST# LIST, Y SHOW TRACK CLEAR NOW 4710 JER SEEKK MOVE HEAD TO TRACK 4720 JER CNTS COUNT SECTORS ON TRACK 4730 LDA FTFTH GET SECTOR COUNT 4740 BEQ MOV3 IF 0 \$\Rightarrow\$ MOV2 4750 LDA #\$01 INIZ 4760 ST# SECT ST#RT AT SECTOR 1 4770 MOV2 JER READ READ SECTOR 4780 LDA SIGAP GET OPEN TRACK # 4790 ST# TRAKX GIVE IT TO 65D 4800 JER SEEKX MOVE HEAD TO TRACK 4810 JER WRITE WRITE SECTOR 4820 LDA SECT CHECK CURRENT SECTOR # 4820 LDA SECT CHECK CURRENT SECTOR # 4830 CMP FIFTH END OF TRACK ? 4840 BEQ MOV3 YES! \$\Rightarrow\$ MOV3 4850 LNC SECT CHECK CURRENT SECTOR # 4860 LDA STBLK GET ORIGIN TRACK # 4870 ST# TRAKX GIVE TO 65D 4880 JER SEEKX MOVE HEAD TO TRACK # 4890 JMP MOV2 AND LOOP! \$\Rightarrow\$ MOV2 4900 MOV3 LDY STGAP GET ORIGIN TRACK # 4910 LDA STBLK GET ORIGIN TR. # AGAIN 4940 CMP ENBLK GET ORIGIN TR. # AGAIN 4940 CMP ENBLK GET ORIGIN TR. # AGAIN 4940 CMP ENBLK GET ORIGIN TR. # AGAIN 4940 LDA STBLK GET ORIGIN TR. # AGAIN 4950 BED MOV4 YES! \$\Rightarrow\$ MOV4 4960 LNC STGAP NOI BUMP DEST. TR. # 4970 LNC STGAP NOI BUMP DEST. TR. # 4990 MOV4 LDA #\$01 INIZ 5000 STA SECT SET SECTOR 1 5010 LDA GAPLEN GET BLOCK LENGTH 5020 JER BECD MAKE IT BCO! 5030 STA GAPLEN GET BLOCK LENGTH 5040 MOV9 JER DIRIN READ IN DIRECTORY SECTOR 5050 LDY #\$00 INIZ 5060 MOV5 LDA DIRBUF, Y FETCH DIR. ENTRY	4640 CAPR	DEV.	BACK LID ONE
### 4660 MOVE LIA STBLK ### 4670 STA TRAKX ### 4680 TAY FUT IN Y AS INDEX ### 4680 TAY FUT IN Y AS INDEX ### 4680 TAY FUT IN Y AS INDEX #### 4680 TAY FUT IN Y AS INDEX #### 4680 TAY FUT IN Y AS INDEX #### 4700 STA LIST, Y SHOW TRACK CLEAR NOW #### 4710 JER SEERK MOVE HEAD TO TRACK #### 4720 JER CNTS COUNT SECTORS ON TRACK #### 4730 LDA FITTH GET SECTOR COUNT #### 4740 BEQ MOV3 IF 0 \$\Rightarrow\$ MOV3 ### 4750 LDA ### 501 INIZ #### 4760 STA SECT START AT SECTOR 1 #### 4770 MOV2 JER READ READ SECTOR ### 4780 LDA STEAP GET OFFEN TRACK # ### 4790 STA TRAKX GIVE IT TO 65D ### 4810 JER WRITE WRITE SECTOR ### 4820 LDA SECT CHECK CURRENT SECTOR # ### 4830 CMP FIFTH END OF TRACK? ### 4840 BEQ MOV3 YES! \$\Rightarrow\$ MOV3 ### 4850 INC SECT KO, BUMP SECTOR # ### 4860 LDA STBLK GET ORIGIN TRACK # ### 4870 STA TRAKX GIVE TO 65D ### 4880 JER SEERX MOVE HEAD TO TRACK ### 4890 JER SEERX ### 4910 LDA ## 501 INIZ ### 4920 STA LIST, Y SHOW TRACK IN USE NOW ### 4930 LDA STBLK AT END OF BLOCK? ### 4950 BEQ MOV4 YES! \$\Rightarrow\$ MOV4 ### 4960 INC STGAP NO! BUMP DEST, TR. # ### 4970 INC STGAP NO! BUMP DEST, TR. # ### 4970 INC STGAP NO! BUMP DEST, TR. # ### 4970 INC STGAP NO! BUMP DEST, TR. # ### 4970 INC STGAP NO! BUMP DEST, TR. # ### 4970 JER SECT SET SECTOR 1 ### 5010 LDA GAPLEN GET BLOCK LEASTH ### 5020 JER BECD MAKE IT BCO! ### 5030 STA GAPLEN GET BLOCK LEASTH ### 5040 MOV9 JER BITN READ IN DIRECTORY SECTOR ### 5040 MOV9 JER DIRIN READ IN DIRECTORY SECTOR ### 5040 MOV9 JER DIRIN READ IN DIRECTORY SECTOR ### 5040 MOV9 JER DIRIN READ IN DIRECTORY SECTOR ### 5040 MOV9 JER DIRIN READ IN DIRECTORY SECTOR	4650	COUNTY PARTS	CALLS END UP DIVAS US #
4670 STR TRAKX GIVE IT TO 65D 4680 TAY PUT IN Y AS INDEX 4690 LDA #\$00 INIZ 4700 STA LIST, Y SHOW TRACK CLEAR NOW 4710 JER SEEKX MOVE HEAD TO TRACK 4710 JER CNTS COUNT SECTORS ON TRACK 4710 JER CNTS COUNT SECTORS ON TRACK 4710 JER CNTS COUNT SECTOR COUNT 4710 JER SEEKX MOVE HEAD TO TRACK 4720 JER CHEAD INIZ 4730 LDA FIFTH GET SECTOR COUNT 4740 BED MOV3 IF 0 \$ROV3 4750 LDA #\$01 INIZ 4760 MOV2 JER READ READ SECTOR 4770 MOV2 JER READ READ SECTOR 4780 LDA SIGAP GET OPEN TRACK # 4790 STA TRAKX GIVE IT TO 65D 4880 JER SEEKX MOVE HEAD TO TRACK 4810 JER WRITE WRITE SECTOR # 4820 LDA SECT CHECK CURRENT SECTOR # 4830 CMP FIFTH END OF TRACK? 4840 BED MOV3 YES! \$ROV3 4850 INC SECT NO, BUNP SECTOR # 4860 LDA STBLK GET CRIGIN TRACK # 4870 STA TRAKX GIVE TO 65D 4880 JER SEEKX MOVE HEAD TO TRACK 4890 JMP MOV2 AND LOOP! \$ROV2 4900 MOV3 LDY STGAP GET DEST. TK. # 4910 LDA #\$01 INIZ 4920 STA LIST, Y SHOW TRACK IN USE NOW 4930 LDA STBLK GET CRIGIN TK. # AGAIN 4940 CMP ENBLK AT END OF BLOCK? 4950 BED MOV4 YES! \$ROV4 4960 INC STGAP NO! BUNP DEST. TK. # 4970 INC STBLK GET CRIGIN TK. # AGAIN 4940 CMP ENBLK AT END OF BLOCK? 4950 BED MOV4 YES! \$ROV4 4960 INC STBLK BUMP ORIGIN TK. # TOO 4990 MOV4 LDA #\$01 INIZ 5000 STA SECT SET SECTOR 1 5010 LDA GAPLEN GET BLOCK LEXCTH 5020 JER BCO MAKE IT BCO! 5030 STA GAPLEN AND PUT IT BACK 5040 MOV9 JER DIRIN READ IN DIRECTORY SECTOR 5050 LDY #\$00 INIZ 5060 MOV5 LDA DIRBUF, Y FETCH DIR. ENTRY	4660 MONTE	T.DO CHOLK	CERT TIDACE & TO MOUTE
### ##################################	4670	CALLY ALDYSAA	CITIE TO TO CED
### ### ### ### ### ### ### ### ### ##	4680	DIG TONEY	GIVE IX XO GOD
4700 STA LIST, Y SHOW TRACK CLEAR NOW 4710 JER SEEKK MOVE HEAD TO TRACK 4720 JER CNTS COUNT SECTORS ON TRACK 4730 LDA FIFTE GET SECTOR COUNT 4740 BEQ MOV3 IF 0 \$\Rightarrow\$ MOV3 4750 LDA \$\frac{1}{9}\$01 INIZ 4760 STA SECT START AT SECTOR 1 4770 MOV2 JER READ READ SECTOR 4780 LDA \$TGAP GET OPEN TRACK \$\rightarrow\$ 4790 STA TRAKX GIVE IT TO 65D 4800 JER SEEKX MOVE HEAD TO TRACK 4810 JER WRITE WRITE SECTOR \$\rightarrow\$ 4820 LDA SECT CHECK CURRENT SECTOR \$\rightarrow\$ 4820 LDA SECT CHECK CURRENT SECTOR \$\rightarrow\$ 4830 CMP FIFTH END OF TRACK \$\rightarrow\$ 4840 BEQ MOV3 YES! \$\Rightarrow\$ MOV3 4850 INC SECT NO, BUMP SECTOR \$\rightarrow\$ 4860 LDA STBLK GET ORIGIN TRACK \$\rightarrow\$ 4870 STA TRAKX GIVE TO 65D 4880 JER SEEKX MOVE HEAD TO TRACK 4890 JMP MOV2 AND LOOP! \$\Rightarrow\$ MOV2 4900 MOV3 LDY STGAP GET DEST. TK. \$\rightarrow\$ 4910 LDA STBLK GET ORIGIN TR. \$\rightarrow\$ AGAIN 4920 STA LIST, Y SHON TRACK IN USE NOW 4930 LDA STBLK GET ORIGIN TK. \$\rightarrow\$ AGAIN 4940 CMP ENBLK AT END OF BLOCK? 4950 BEQ MOV4 YES! \$\Rightarrow\$ MOV4 4960 INC STGAP NO! BUMP DEST. TR. \$\rightarrow\$ 4970 LNC STGAP NO! BUMP DEST. TR. \$\rightarrow\$ 4970 LNC STGAP NO! BUMP DEST. TR. \$\rightarrow\$ 4990 MOV4 LDA \$\rightarrow\$ INIZ 5000 STA SECT SET SECTOR \$\bar{1}\$ 5010 LDA GAPLEN GET BLOCK LEARTH 5020 JER BECD MAKE IT BCD! 5030 STA GAPLEN AND PUT IT BACK 5040 MOV9 JER DIRIN READ IN DIRECTORY SECTOR 5050 LDY \$\rightarrow\$ OO INIZ 5060 MOV5 LDA DIRBUF, Y FEICH DIR. ENTRY	4600	LDV 4600	TAITS AS TANKY
47100 JER SEERK MCVE HEAD TO TRACK 4720 JER CNTS COUNT SECTORS ON TRACK 4730 LDA FIFTE GET SECTOR COUNT 4740 BEQ MOV3 IF 0 \$\implies \text{MOVE}\$ HEAD TO TRACK 4750 LDA \$\forall \text{SD1}\$ INU 4750 LDA \$\forall \text{SD1}\$ INU 4760 STA \$\forall \text{SECT}\$ START AT SECTOR 1 4770 MOV2 JER READ READ SECTOR 4780 LDA \$\forall \text{START}\$ READ SECTOR 4780 LDA \$\forall \text{START}\$ GET OPEN TRACK \$\pi\$ 4790 STA TRAKX GIVE IT TO 65D 4800 JER SEEKX MOVE HEAD TO TRACK 4810 JER WRITE WRITE SECTOR \$\pi\$ 4820 LDA \$\forall \text{SECT}\$ CHECK CURRENT SECTOR \$\pi\$ 4830 CMP FIFTH END OF TRACK \$\pi\$ 4840 BEQ MOV3 YES! \$\Rightarrow \text{MOV3}\$ 4850 INC SECT RO, BUMP SECTOR \$\pi\$ 4860 LDA \$\forall \text{SELK}\$ GET ORIGIN TRACK \$\pi\$ 4870 STA TRAKX GIVE TO 65D 4880 JER SEEKX MOVE HEAD TO TRACK 4890 JMP MOV2 AND LOOP! \$\Rightarrow \text{MOV2}\$ 4910 MOV3 LDY \$\forall \text{STGAP}\$ GET DEST. TR. \$\pi\$ 4910 LDA \$\forall \text{STGAP}\$ GET ORIGIN TR. \$\pi \text{ aGAIN}\$ 4920 STA LIST,Y SHOW TRACK IN USE NOW 4930 LDA \$\forall \text{STELK}\$ GET ORIGIN TR. \$\pi \text{ aGAIN}\$ 4940 CMP ENBLK AT END OF BLOCK? 4950 BEQ MOV4 YES! \$\Rightarrow \text{ MOVE}\$ 4950 BIO MOV4 YES! \$\Rightarrow \text{ MOVE}\$ 4960 INC \$\forall \text{STGAP}\$ NOI BUMP DEST. TR. \$\pi\$ 4970 INC \$\forall \text{STELK}\$ BUMP ORIGIN TR. \$\pi \text{ TOO}\$ 4990 MOV4 LDA \$\pi \text{ IDA }\text{ SPOI}\$ INIZ 5000 STA SECT SET SECTOR 1 5010 LDA \$\forall \text{ CET DICK LEMTH}\$ 5020 JER BECD MAKE IT BCDI 5030 STA GAPLEN GET ELOCK LEMTH 5040 MOV9 JER DIRIN READ IN DIRECTORY SECTOR 5050 LDY \$\pi \pi \text{ IDA }\text{ INIZ }\$ 5060 MOV5 LDA DIRBUF,Y FETCH DIR. ENTRY	4700	CALLY ILCON AL	TIME
4720 JSR CNTS COUNT SECTORS ON TRACK 4730 LDA FIFTE GET SECTOR COUNT 4740 BEQ MOV3 IF 0 => MOV3 4750 LDA \$901 INIZ 4760 STA SECT START AT SECTOR 1 4770 MOV2 JSR READ READ SECTOR 4780 LDA STGAP GET OPEN TRACK # 4790 STA TRAKX GIVE IT TO 65D 4800 JSR SEEKX MOVE HEAD TO TRACK 4810 JSR WRITE WRITE SECTOR # 4820 LDA SECT CHECK CURRENT SECTOR # 4830 CMP FIFTH END OF TRACK ? 4840 BEQ MOV3 YES! => MOV3 4850 INC SECT NO, BUMP SECTOR # 4860 LDA STELK GET CRIGIN TRACK # 4870 STA TRAKX GIVE TO 65D 4860 JSR SEEKX MOVE HEAD TO TRACK # 4870 STA TRAKX GIVE TO 65D 4860 JSR SEEKX MOVE HEAD TO TRACK # 4890 JMP MOV2 AND LOOP! => MOV2 4910 MOV3 LDY STGAP GET DEST. TR. # 4910 LDA \$501 INIZ 4920 STA LIST,Y SHOW TRACK IN USE NOW 4930 LDA STELK GET CRIGIN TR. # AGAIN 4940 CMP ENBLK AT END OF BLOCK? 4950 BEQ MOV4 YES! => MOV4 4960 INC STGAP NO! BUMP DEST. TR. # 4970 LIC STELK BUMP ORIGIN TR. # TOO 4960 BNE MOVE AND LOOP! => MOVE 4990 MOV4 LDA \$501 INIZ 5000 STA SECT SET SECTOR 1 5010 LDA GAPLEN GET BLOCK LENGTH 5020 JSR GAPLEN MAKE IT BCDI 5030 STA GAPLEN MAKE IT BCDI 5040 MOV9 JSR DIRIN READ IN DIRECTORY SECTOR 5050 LDY \$500 INIZ	4710	DED OFFISA	MOTE TELL DO TELLO
4730 LDA FIFTH GET SECTOR COUNT 4740 BEQ MOV3 IF 0 → MOV3 4750 LDA \$\$01 INIZ 4760 STA SECT START AT SECTOR 1 4770 MOV2 JER READ READ SECTOR 4780 LDA STGAP GET OPEN TRACK \$ 4790 STA TRAKK GIVE IT TO 65D 4800 JER SEEKK MOVE HEAD TO TRACK 4810 JER WRITE WRITE SECTOR \$ 4820 LDA SECT CHECK CURRENT SECTOR \$ 4830 CMP FIFTH END OF TRACK ? 4840 BEQ MOV3 YES! → MOV3 4850 INC SECT NO, BUMP SECTOR \$ 4860 LDA STELK GET CRIGIN TRACK \$ 4870 SIA TRAKK GIVE TO 65D 4880 JER SEEKX MOVE HEAD TO TRACK \$ 4890 MOV3 LDY STGAP GET DEST. TK. \$ 4910 MOV3 LDY STGAP GET DEST. TK. \$ 4920 STA LIST, Y SHOW TRACK IN USE NOW \$ 4930 LDA STELK GET ORIGIN TK. \$ AGAIN \$ 4940 CMP EMBLK AT END OF BLOCK? 4950 BEQ MOV4 YES! → MOV4 4960 INC STGAP NOI BUMP DEST. TK. \$ 4970 INC STGAP NOI BUMP DEST. TK. \$ 5010 LDA GAPLEN GET BLOCK LENGTH 5020 JER BECD MAKE IT BCDI 5030 STA GAPLEN AND PUT IT BACK 5040 MOV9 JER DIRIN READ IN DIRECTORY SECTOR 5050 LDY \$\$00 INIZ	4720	TOD MITTO	MOVE HEAD TO TRACK
4730 BEQ MOV3 IF 0 ⇒ MOV3 4750 LDA \$\$01 INIZ 4760 STA SECT START AT SECTOR 1 4770 MOV2 JER READ READ SECTOR 4780 LDA STGAP GET OPEN TRACK \$ 4790 STA TRAKX GIVE IT TO 65D 48810 JER WRITE WRITE SECTOR \$ 4820 LDA SECT CHECK CURRENT SECTOR \$ 4820 LDA SECT CHECK CURRENT SECTOR \$ 4820 LDA SECT CHECK CURRENT SECTOR \$ 4840 BEQ MOV3 YES! ⇒ MOV3 4850 INC SECT NO, BUMP SECTOR \$ 4860 LDA STBLK GET ORIGIN TRACK \$ 4870 STA TRAKX GIVE TO 65D 4880 JER SEEKX MOVE HEAD TO TRACK \$ 4890 JMP MOV2 AND LOOP! ⇒ MOV2 4900 MOV3 LDY STGAP GET DEST. TX. \$ 4910 LDA \$501 INIZ 4920 STA LIST,Y SHOW TRACK IN USE NOW \$ 4930 LDA STBLK GET ORIGIN TR. \$ AGAIN \$ 4940 CMP ENBLK AT END OF BLOCK? \$ 4950 BEQ MOV4 YES! ⇒ MOV4 4960 INC STGAP NO1 BUMP DEST. TR. \$ 4970 INC STGAP NO1 BUMP DEST. TR. \$ 5010 LDA GAPLEN GET BLOCK LENGTH SO20 JER BECD MAKE IT BCD1 5030 STA GAPLEN READ IN DIRECTORY SECTOR \$ 5040 MOV9 JER DIRIN READ IN DIRECTORY SECTOR \$ 5050 LDY \$\$00 INIZ	4720	DOV CATE	COUNT SECTIONS ON TRACK
4750 LDA \$\$01 INIZ 4760 STA SECT STERT AT SECTOR 1 4770 MOV2 JER READ READ SECTOR 4780 LDA STGAP GET OPEN TRACK # 4790 STA TRAKX GIVE IT TO 65D 4800 JER SEEKX MOVE HEAD TO TRACK 4810 JER WRITE WRITE SECTOR # 4820 LDA SECT CHECK CURRENT SECTOR # 4830 CMP FIFTH END OF TRACK ? 4840 BEQ MOV3 YES! => MOV3 4850 INC SECT NO, BUMP SECTOR # 4860 LDA STBLK GET ORIGIN TRACK # 4870 STA TRAKX GIVE TO 65D 4880 JER SEEKX MOVE HEAD TO TRACK 4890 JMP MOV2 AND LOOP! => MOV2 4900 MOV3 LDY STGAP GET DEST. TK. # 4910 LDA STBLK GET ORIGIN TR. # AGAIN 4920 STA LIST,Y SHON TRACK IN USE NOW 4930 LDA STBLK GET ORIGIN TK. # AGAIN 4940 CMP ENBLK AT END OF BLOCK? 4950 BEQ MOV4 YES! => MOV4 4960 INC STGAP NO! BUMP DEST. TK. # 4970 LNC STGAP NO! BUMP DEST. TK. # 4970 LNC STGAP NO! BUMP DEST. TK. # 4970 LNC STGAP NO! BUMP DEST. TK. # 4990 MOV4 LDA \$\$01 INIZ 5000 STA SECT SET SECTOR 1 5010 LDA GAPLEN GET BLOCK LEASTH 5020 JER BECD MAKE IT BCD! 5030 STA GAPLEN READ IN DIRECTORY SECTOR 5040 MOV9 JER DIRIN READ IN DIRECTORY SECTOR 5050 LDY \$\$00 INIZ	47.40	TAM LILID	GET SECTOR COUNT
4750 4760 578 SECT 578 SECT 578 SECT 4770 MOV2 JER READ READ SECTOR 4780 LDA STGAP GET OPEN TRACK # 4790 STA TRAKX GIVE IT TO 65D 4800 JER SEEKX MOVE HEAD TO TRACK 4810 JER WRITE WRITE SECTOR 4820 LDA SECT CHECK CURRENT SECTOR # 4830 CMP FIFTH END OF TRACK? 4840 BEQ MOV3 YES! => MOV3 4850 INC SECT RO, BUMP SECTOR # 4860 LDA STBLK GET ORIGIN TRACK # 4870 STA TRAKX GIVE TO 65D JER SEEKX MOVE HEAD TO TRACK 4890 JMP MOV2 AND LOOP! => MOV2 4910 MOV3 LDY STGAP GET DEST. TK. # 4910 LDA \$SO1 INIZ 4920 STA LIST,Y SHOW TRACK IN USE NOW 4930 LDA STBLK GET ORIGIN TK. # AGAIN 4940 CMP ENBLK GET ORIGIN TK. # AGAIN 4940 CMP ENBLK GET ORIGIN TK. # 4950 BEQ MOV4 YES! => MOV4 4960 INC STGAP NOI BUMP DEST. TK. # 4970 INC STGAP NOI BUMP DEST. TK. # 4970 INC STGAP NOI BUMP DEST. TK. # 4970 LDA SEBLK BUMP ORIGIN TK. # TOO AGAILEN 4990 MOV4 LDA \$SO1 INIZ 5000 STA SECT SET SECTOR 1 5010 LDA GAPLEN GET BLOCK LEMTH 5020 JER BECD MAKE IT BCDI 5030 STA GAPLEN SEAD INIZ 5040 MOV9 JER DIRIN READ IN DIRECTORY SECTOR 5050 LDY \$\$00 INIZ	47 40 47 50	TON ACOL	THE O => MOV3
4770 MOV2 JER READ READ SECTOR 4780 LDA STGAP GET OPEN TRACK # 4790 SIA TRAKX GIVE IT TO 65D 4800 JER SEEKX MOVE HEAD TO TRACK 4810 JER WRITE WRITE SECTOR 4820 LDA SECT CHECK CURRENT SECTOR # 4830 CMP FIFTH END OF TRACK ? 4840 BEQ MOV3 YES! => MOV3 4850 INC SECT NO, BUMP SECTOR # 4860 LDA STELK GET CRIGIN TRACK # 4870 SIA TRAKX GIVE TO 65D 4880 JER SEEKX MOVE HEAD TO TRACK # 4890 JMP MOV2 AND LOOP! => MOV2 4900 MOV3 LDY STGAP GET DEST. TK. # 4910 LDA #\$01 INIZ 4920 STA LIST,Y SHOW TRACK IN USE NOW 4930 LDA STELK GET CRIGIN TK. # AGAIN 4940 CMP ENBLK GET CRIGIN TK. # AGAIN 4940 CMP ENBLK GET CRIGIN TK. # AGAIN 4940 CMP ENBLK GET CRIGIN TK. # AGAIN 4950 BEQ MOVA YES! => MOVA 4960 INC STGAP NO! BUMP DEST. TK. # 4970 LNC STELK BUMP CRIGIN TK. # TOO 4980 BNE MOVE AND LOOP! => MOVE 4990 MOV4 LDA #\$01 INIZ 5000 STA SECT SET SECTOR 1 5010 LDA GAPLEN GET ELOCK LENGTH 5020 JER BECD MAKE IT BCD 5030 STA GAPLEN AND PUT IT BACK 5040 MOV9 JER DIRIN READ IN DIRECTORY SECTOR 5050 LDY #\$00 INIZ	4730	TINA #20T	INIZ
4780 LDA STIGAP GET OPEN TRACK # 4790 STA TRAKK GIVE IT TO 65D 4800 JER SEEKK MOVE HEAD TO TRACK 4810 JER WRITE WRITE SECTOR 4820 LDA SECT CHECK CURRENT SECTOR # 4830 CMP FIFTH END OF TRACK ? 4840 BEQ MOV3 YES! -> MOV3 4850 INC SECT NO, BUMP SECTOR # 4870 STA TRAKK GIVE TO 65D 4886 JER SEEKK MOVE HEAD TO TRACK # 4870 STA TRAKK GIVE TO 65D 4886 JER SEEKK MOVE HEAD TO TRACK # 4890 JMP MOV2 AND LOOP! -> MOV2 4900 MOV3 LDY STGAP GET DEST. TK. # 4910 LDA #SOI INIZ 4920 STA LIST, Y SHOW TRACK IN USE NOW 4930 LDA STBLK GET ORIGIN TK. # AGAIN 4940 CMP EMBLK AT END OF BLOCK? 4950 BEQ MOV4 YES! -> MOV4 4950 INC STGAP NOI BUMP DEST. TK. # 4970 INC STGAP NOI BUMP DEST. TK. # 5010 STA SECT SET SECTOR 1 5010 LDA GAPLEN GET BLOCK LENGTH 5020 JER BECD MAKE IT BCDI 5030 STA GAPLEN AND PUT IT BACK 5040 MOV9 JER DIRIN READ IN DIRECTORY SECTOR 5050 LDY #\$00 INIZ	4/60	SIA SECT	START AT SECTOR 1
4790 SIA TRAKX GIVE IT TO 65D  4800 JSR SEEKX MOVE HEAD TO TRACK  4810 JSR WRITE WRITE SECTOR  4820 LDA SECT CHECK CURRENT SECTOR #  4830 CMP FIFTH END OF TRACK?  4840 BEQ MOV3 YES! -> MOV3  4850 INC SECT NO, BUMP SECTOR #  4860 LDA STBLK GET ORIGIN TRACK #  4870 SIA TRAKX GIVE TO 65D  4880 JSR SEEKX MOVE HEAD TO TRACK  4890 JMP MOV2 AND LOOP! -> MOV2  4900 MOV3 LDY STGAP GET DEST. TX. #  4910 LDA #\$01 INIZ  4920 STA LIET,Y SHON TRACK IN USE NOW  4930 LDA STBLK GET ORIGIN TR. # AGAIN  4940 CMP ENBLK AT END OF BLOCK?  4950 BEQ MOV4 YES! -> MOV4  4960 INC STGAP NO1 BUMP DEST. TR. #  4970 INC STGAP SECTOR 1  5010 LDA GAPLEN GET BLOCK LEASTH  5020 JSR GECT SET SECTOR 1  5030 STA GAPLEN GET BLOCK LEASTH  5020 JSR GAPLEN AND PUT IT BACK  5040 MOV9 JSR DIRIN READ IN DIRECTORY SECTOR  5050 LDY #\$00 INIZ  5060 MOV5 LDA DIRBUF,Y FETCH DIR. ENTRY	4770 19072	JEK READ	READ SECTOR
4800 JER SEEKX MOVE HEAD TO TRACK 4810 JER WRITE WRITE SECTOR 4820 LDA SECT CHECK CURRENT SECTOR # 4830 CMP FIFTH END OF TRACK ? 4840 BEQ MOV3 YES! => MOV3 4850 INC SECT NO, BUMP SECTOR # 4860 LDA STBLK GET ORIGIN TRACK # 4870 STA TRAKX GIVE TO 65D 4880 JER SEEKX MOVE HEAD TO TRACK 4890 JMP MOV2 AND LOOP! => MOV2 4900 MOV3 LDY STGAP GET DEST. TK. # 4910 LDA STBLK GET ORIGIN TR. # 4910 LDA STBLK GET ORIGIN TK. # 4920 STA LIST,Y SHON TRACK IN USE NOW 4930 LDA STBLK GET ORIGIN TK. # AGAIN 4940 CMP ENBLK AT END OF BLOCK? 4950 BEQ MOV4 YES! => MOV4 4960 INC STGAP NO! BUMP DEST. TK. # 4970 INC STGAP NO! BUMP DEST. TK. # 4970 INC STBLK BUMP ORIGIN TK. # TOO 4980 BNE MOVE AND LOOP! => MOVE 4990 MOV4 LDA \$501 INIZ 5000 STA SECT SET SECTOR 1 5010 LDA GAPLEN GET BLOCK LENGTH 5020 JER BECD MAKE IT BCD! 5030 STA GAPLEN READ IN DIRECTORY SECTOR 5040 MOV9 JER DIRIN READ IN DIRECTORY SECTOR 5050 LDY \$\$00 INIZ	4760	LUA SIGAP	GET OPEN TRACK #
### ### ### ### ### ### ### ### ### ##	4/90	SIA IRAKX	GIVE IT TO 65D
4820 LDA SECT CHECK CURRENT SECTOR # 4830 CMP FIFTH END OF TRACK ? 4840 BEQ MOV3 YES! => MOV3 4850 INC SECT RO, BUMP SECTOR # 4860 LDA STBLK GET ORIGIN TRACK # 4870 STA TRAKX GIVE TO 65D 4880 JSR SEEKX MOVE HEAD TO TRACK # 4890 JMP MOV2 AND LOOP! => MOV2 4900 MOV3 LDY STGAP GET DEST. TK. # 4910 LDA #\$01 INIZ 4920 STA LIST,Y SHOW TRACK IN USE NOW 4930 LDA STBLK GET ORIGIN TK. # AGAIN 4940 CMP EMBLK AT END OF BLOCK? 4950 BEQ MOV4 YES! => MOV4 4950 INC STGAP NO! BUMP DEST. TK. # 4970 INC STGAP NO! BUMP DEST. TK. # 4970 INC STGAP NO! BUMP DEST. TK. # 4970 INC STBLK BUMP ORIGIN TK. # TOO 4980 MOV4 LDA #\$01 INIZ 5000 STA SECT SET SECTOR 1 5010 LDA GAPLEN GET BLOCK LENGTH 5020 JSR BECD MAKE IT BCD! 5030 STA GAPLEN AND PUT IT BACK 5040 MOV9 JSR DIRIN READ IN DIRECTORY SECTOR 5050 LDY #\$00 INIZ	4800	JER SEEKX	MOVE HEAD TO TRACK
4820 LDA SECT CHECK CURRENT SECTOR # 4830 CMP FIFTH END OF TRACK ? 4840 BEQ MOV3 YES! => MOV3 4850 INC SECT NO, BUMP SECTOR # 4860 LDA STELK GET ORIGIN TRACK # 4870 SIA TRAKX GIVE TO 65D 4880 JER SEEKX MOVE HEAD TO TRACK 4890 JMP MOV2 AND LOOP! => MOV2 4900 MOV3 LDY STGAP GET DEST. TK. # 4910 LDA #SOI INIZ 4920 STA LIST, Y SHOW TRACK IN USE NOW 4930 LDA STELK GET ORIGIN TK. # AGAIN 4940 CMP EMBLK AT END OF BLOCK? 4950 BEQ MOV4 YES! => MOV4 4950 INC STGAP NOI BUMP DEST. TK. # 4970 INC STGAP NOI BUMP DEST. TK. # 4970 INC STGAP NOI BUMP DEST. TK. # 4970 INC STGAP NOI BUMP DEST. TK. # 5010 STA SECT SET SECTOR 1 5010 LDA GAPLEN GET BLOCK LENGTH 5020 JER BECD MAKE IT BCDI 5030 STA GAPLEN READ IN DIRECTORY SECTOR 5040 MOV9 JER DIRIN READ IN DIRECTORY SECTOR 5050 LDY #\$00 INIZ	4810	JSR WRITE	Write Sector
4830 CMP FIFTH END OF TRACK ? 4840 BEQ MOV3 YES! => MOV3 4850 INC SECT RO, BUMP SECTOR # 4860 LDA STBLK GET ORIGIN TRACK # 4870 SIA TRAKX GIVE TO 65D 4880 JER SEEKX MOVE HEAD TO TRACK 4890 JMP MOV2 AND LOOP! => MOV2 4900 MOV3 LDY STGAP GET DEST. TX. # 4910 LDA #\$01 INIZ 4920 STA LIST,Y SHOW TRACK IN USE NOW 4930 LDA STBLK GET ORIGIN TX. # AGAIN 4940 CMP ENBLK AT END OF BLOCK? 4950 BEQ MOV4 YES! => MOV4 4960 INC STGAP NO1 BUMP DEST. TX. # 4970 INC STGAP NO1 BUMP DEST. TX. # 4970 INC STGAP NO1 BUMP DEST. TX. # 4970 INC STGAP NO1 BUMP DEST. TX. # 5010 STA SECT SET SECTOR 1 5010 LDA GAPLEN GET BLOCK LENGTH 5020 JER BECD MAKE IT BCD1 5030 STA GAPLEN READ IN DIRECTORY SECTOR 5040 MOV9 JER DIRIN READ IN DIRECTORY SECTOR 5050 LDY #\$00 INIZ	4820	LDA SECT	CHECK CURRENT SECTOR #
4840 BEQ MOV3 YES! => MOV3  4850 INC SECT NO, BUMP SECTOR #  4860 LDA STBLK GET ORIGIN TRACK #  4870 STA TRAKK GIVE TO 65D  4880 JER SEEKK MOVE HEAD TO TRACK  4890 JMP MOV2 AND LOOP! => MOV2  4900 MOV3 LDY STGAP GET DEST. TK. #  4910 LDA #SO1 INIZ  4920 STA LIST,Y SHOW TRACK IN USE NOW  4930 LDA STBLK GET ORIGIN TK. # AGAIN  4940 CMP ENBLK AT END OF BLOCK?  4950 BEQ MOV4 YES! => MOV4  4960 INC STGAP NO! BUMP DEST. TK. #  4970 INC STGAP NO! BUMP DEST. TK. #  4970 INC STGAP NO! BUMP ORIGIN TK. # TOO  4980 BNE MOVE AND LOOP! => MOVE  4990 MOV4 LDA #\$O1 INIZ  5000 STA SECT SET SECTOR 1  5010 LDA GAPLEN GET BLOCK LENGTH  5020 JER BECD MAKE IT BCD!  5030 STA GAPLEN AND PUT IT BACK  5040 MOV9 JER DIRIN READ IN DIRECTORY SECTOR  5050 LDY #\$00 INIZ	4830	CMP FIFTH	END OF TRACK ?
4850 INC SECT NO, BUMP SECTOR # 4860 LDA STBLK GET ORIGIN TRACK # 4870 STA TRAKX GIVE HEAD TO TRACK 4880 JSR SEEKX MOVE HEAD TO TRACK 4890 JMP MOV2 AND LOOP! => MOV2 4900 MOV3 LDY STGAP GET DEST. TK. # 4910 LDA #\$01 INIZ 4920 STA LIST,Y SHOW TRACK IN USE NOW 4930 LDA STBLK GET ORIGIN TK. # AGAIN 4940 CMP ENBLK AT END OF BLOCK? 4950 BED MOV4 YES! => MOV4 4960 INC STGAP NO! BUMP DEST. TK. # 4970 LNC STBLK BUMP ORIGIN TK. # TOO 4980 BNE MOVE AND LOOP! => MOVE 4990 MOV4 LDA #\$01 INIZ 5000 STA SECT SET SECTOR 1 5010 LDA GAPLEN GET BLOCK LENGTH 5020 JSR BECD MAKE IT BCD! 5030 STA GAPLEN GET BLOCK LENGTH 5040 MOV9 JSR DIRIN READ IN DIRECTORY SECTOR 5050 LDY #\$00 INIZ	4840	BEO MOV3	YES! => MOV3
4850 LDA STELK GET ORIGIN TRACK # 4870 STA TRAKX GIVE TO 65D  4880 JSR SEEKX MOVE HEAD TO TRACK 4890 JMP MOV2 AND LOOP! => MOV2 4900 MOV3 LDY STGAP GET DEST. TK. # 4910 LDA #\$01 INIZ 4920 STA LIST,Y SHOW TRACK IN USE NOW 4930 LDA STELK GET ORIGIN TK. # AGAIN 4940 CMP ENBLK AT END OF BLOCK? 4950 BEQ MOV4 YES! => MOV4 4960 INC STGAP NO! BUMP DEST. TK. # 4970 INC STGAP NO! BUMP DEST. TK. # 4970 INC STELK BUMP ORIGIN TK. # TOO 4980 BNE MOVE AND LOOP! => MOVE 4990 MOV4 LDA #\$01 INIZ 5000 STA SECT SET SECTOR 1 5010 LDA GAPLEN GET BLOCK LENGTH 5020 JSR BECD MAKE IT BCD! 5030 STA GAPLEN AND FUT IT BACK 5040 MOV9 JSR DIRIN READ IN DIRECTORY SECTOR 5050 LDY #\$00 INIZ 5060 MOV5 LDA DIRBUF,Y FETCH DIR. ENTRY	4850	INC SECT	NO, BUMP SECTOR #
4870 STA TRAKX GIVE TO 65D  4880 JER SEEKX MOVE HEAD TO TRACK  4890 JMP MOV2 AND LOOP! => MOV2  4900 MOV3 LDY STGAP GET DEST. TK. #  4910 LDA #SOL INIZ  4920 STA LIST,Y SHOW TRACK IN USE NOW  4930 LDA STBLK GET ORIGIN TK. # AGAIN  4940 CMP EMBLK AT END OF BLOCK?  4950 BEQ MOV4 YES! => MOV4  4960 INC STGAP NOL BUMP DEST. TK. #  4970 INC STGAP NOL BUMP DEST. TK. #  4970 INC STGAP NOL BUMP DEST. TK. #  4970 INC STBLK BUMP ORIGIN TK. # TOO  4980 BNE MOVE AND LOOP! => MOVE  4990 MOV4 LDA #\$OL INIZ  5000 STA SECT SET SECTOR 1  5010 LDA GAPLEN GET BLOCK LENGTH  5020 JER BECD MAKE IT BCD!  5030 STA GAPLEN AND PUT IT BACK  5040 MOV9 JER DIRIN READ IN DIRECTORY SECTOR  5050 LDY #\$00 INIZ  5060 MOV5 LDA DIRBUF,Y FETCH DIR. ENTRY	4860	LDA STBLK	GET ORIGIN TRACK #
4880 JSR SEEKX MOVE HEAD TO TRACK 4890 JMP MOV2 AND LOOP! ==> MOV2 4900 MOV3 LDY STGAP GET DEST. TK. # 4910 LDA #\$01 INIZ 4920 STA LLET,Y SHOW TRACK IN USE NOW 4930 LDA STBLK GET ORIGIN TK. # AGAIN 4940 CMP ENBLK AT END OF BLOCK? 4950 BBQ MOV4 YES! ==> MOV4 4960 INC STGAP NO1 BUMP DEST. TK. # 4970 INC STGAP NO1 BUMP DEST. TK. # 4970 INC STGAP NO1 BUMP DEST. TK. # 4980 BNE MOVE AND LOOP! ==> MOVE 4990 MOV4 LDA #\$01 INIZ 5000 STA SECT SET SECTOR 1 5010 LDA GAPLEN GET BLOCK LENGTH 5020 JER BECD MAKE IT BCD1 5030 STA GAPLEN AND PUT IT BACK 5040 MOV9 JER DIRIN READ IN DIRECTORY SECTOR 5050 LDY #\$00 INIZ 5060 MOV5 LDA DIRBUF,Y FETCH DIR. ENTRY	4870	STA TRAKX	GIVE TO 65D
4890 JMP MOV2 AND LOOP! ==> MOV2 4900 MOV3 LDY STGAP GET DEST. TK. # 4910 LDA #\$01 INIZ 4920 STA LIBT,Y SHOW TRACK IN USE NOW 4930 LDA STBLK GET ORIGIN TK. # AGAIN 4940 CMP ENBLK AT END OF BLOCK? 4950 BBQ MOV4 YES! ==> MOV4 4960 INC STGAP NOI BUMP DEST. TK. # 4970 INC STBLK BUMP ORIGIN TK. # TOO 4980 BNE MOVE AND LOOP! ==> MOVE 4990 MOV4 LDA #\$01 INIZ 5000 STA SECT SET SECTOR 1 5010 LDA GAPLEN GET BLOCK LEAGTH 5020 JER BECD MAKE IT BCD1 5030 STA GAPLEN AND PUT IT BACK 5040 MOV9 JER DIRIN READ IN DIRECTORY SECTOR 5050 LDY #\$00 INIZ 5060 MOV5 LDA DIRBUF,Y FETCH DIR. ENTRY	4880	JER SEEKX	MOVE HEAD TO TRACK
4900 MOV3 LDY STGAP GET DEST. TR. #  4910 LDA \$501 INIZ  4920 STA LIST,Y SHOW TRACK IN USE NOW  4930 LDA STBLK GET ORIGIN TR. # AGAIN  4940 CMP EMBLK AT END OF BLOCK?  4950 BEQ MOV4 YES! => MOV4  4960 INC STGAP NO! BUMP DEST. TR. #  4970 LNC STBLK BUMP ORIGIN TK. # TOO  4980 BNE MOVE AND LOOP! => MOVE  4990 MOV4 LDA \$501 INIZ  5000 STA SECT SET SECTOR 1  5010 LDA GAPLEN GET BLOCK LENGTH  5020 JER BECD MAKE IT BCD!  5030 STA GAPLEN AND PUT IT BACK  5040 MOV9 JER DIRIN READ IN DIRECTORY SECTOR  5050 LDY \$500 INIZ  5060 MOV5 LDA DIRBUF,Y FETCH DIR. ENTRY	4890	JMP MOV2	AND LOOP! ==> MOV2
4910 LDA #\$01 INIZ 4920 STA LIST,Y SHOW TRACK IN USE NOW 4930 LDA STBLK GET ORIGIN TK. # AGAIN 4940 CMP ENBLK AT END OF BLOCK? 4950 BBQ MCV4 YES! -> MCV4 4960 INC STGAP NOI BUMP DEST. TK. # 4970 INC STBLK BUMP ORIGIN TK. # TOO 4980 BNE MCVE AND LOOP! -> MCVE 4990 MCV4 LDA #\$01 INIZ 5000 STA SECT SET SECTOR 1 5010 LDA GAPLEN GET BLOCK LENGTH 5020 JER BBCD MARE IT BCD1 5030 STA GAPLEN AND FUT IT BACK 5040 MCV9 JER DIRIN READ IN DIRECTORY SECTOR 5050 LDY #\$00 INIZ 5060 MCV5 LDA DIRBUF,Y FETCH DIR. ENTRY	4900 MOV3	LDY STGAP	GET DEST. TR. #
4920 STA LIST,Y  4930 LDA STBLK GET ORIGIN TR. # AGAIN  4940 CMP ENBLK AT END OF BLOCK?  4950 BBQ MOV4 YES! => MOV4  4960 INC STGAP NO! BUMP DEST. TR. #  4970 INC STBLK BUMP ORIGIN TR. # TOO  4980 BNE MOVE AND LOOP! => MOVE  4990 MOV4 LDA #\$01 INIZ  5000 STA SECT SET SECTOR 1  5010 LDA GAPLEN GET BLOCK LENGTH  5020 JSR BECD MAKE IT BCD!  5030 STA GAPLEN AND PUT IT BACK  5040 MOV9 JSR DIRIN READ IN DIRECTORY SECTOR  5050 LDY #\$00 INIZ  5060 MOV5 LDA DIRBUF,Y FETCH DIR. ENTRY	4910	LDA #\$01	INIZ
4930 LDA STBLK GET ORIGIN TK. # AGAIN 4940 CMP EMBLK AT END OF BLOCK? 4950 BEQ MOV4 YES! => MOV4 4960 INC STGAP NOI BUMP DEST. TK. # 4970 INC STBLK BUMP ORIGIN TK. # TOO 4980 BNE MOVE AND LOOP! => MOVE 4990 MOV4 LDA #\$01 INIZ 5000 STA SECT SET SECTOR 1 5010 LDA GAPLEN GET BLOCK LEAGTH 5020 JER BECD MAKE IT BCD! 5030 STA GAPLEN AND PUT IT BACK 5040 MOV9 JER DIRIN READ IN DIRECTORY SECTOR 5050 LDY #\$00 INIZ 5060 MOV5 LDA DIRBUF,Y FETCH DIR. ENTRY	4920	STA LIST, Y	SHOW TRACK IN USE NOW
4940 CMP ENBLK AT END OF BLOCK?  4950 BBQ MCV4 YES! => MCV4  4960 INC STGAP NOI BUMP DEST. TR. #  4970 INC STBLK BUMP ORIGIN TR. # TOO  4980 BNE MCVE AND LOOP! => MCVE  4990 MCV4 LDA #\$01 INIZ  5000 STA SECT SET SECTOR 1  5010 LDA GAPLEN GET BLOCK LEAGTH  5020 JER BECD MAKE IT BCD1  5030 STA GAPLEN AND PUT IT BACK  5040 MCV9 JER DIRIN READ IN DIRECTORY SECTOR  5050 LDY #\$00 INIZ  5060 MCV5 LDA DIRBUF, Y FETCH DIR. ENTRY	4930	LDA STBLK	GET ORIGIN TR. # AGAIN
4950 BBQ MOV4 YES! => MOV4 4960 INC STGAP NO! BUMP DEST. TR. # 4970 INC STBLK BUMP ORIGIN TR. # TOO 4980 BNE MOVE AND LOOP! => MOVE 4990 MOV4 LDA #\$01 INIT 5000 STA SECT SET SECTOR ! 5010 LDA GAPLEN GET BLOCK LENGTH 5020 JER HBCD MAKE IT BCD! 5030 STA GAPLEN AND PUT IT BACK 5040 MOV9 JER DIRIN READ IN DIRECTORY SECTOR 5050 LDY #\$00 INIZ 5060 MOV5 LDA DIRBUF, Y FETCH DIR. ENTRY	4940	CMP ENBLK	AT END OF BLOCK?
4960 INC STGAP NOI BUMP DEST. TR. # 4970 INC STBLK BUMP ORIGIN TR. # TOO 4980 BNE MOVE AND LOOP1 => MOVE 4990 MOV4 LDA #\$01 INTE 5000 STA SECT SET SECTOR 1 5010 LDA GAPLEN GET BLOCK LENGTH 5020 JER BECD MAKE IT BCD1 5030 STA GAPLEN AND PUT IT BACK 5040 MOV9 JER DIRIN READ IN DIRECTORY SECTOR 5050 LDY #\$00 INIZ 5060 MOV5 LDA DIRBUF, Y FETCH DIR. ENTRY	4950	BBO MOV4	YES! ==> MOV4
4970 INC STBLK BIMP ORIGIN TK. # TOO 4980 BNE MOVE AND LOOP1 => MOVE 4990 MOV4 LDA #\$01 INTZ 5000 STA SECT SET SECTOR 1 5010 LDA GAPLEN GET BLOCK LENGTH 5020 JSR BECD MAKE IT BCD1 5030 STA GAPLEN AND PUT IT BACK 5040 MOV9 JSR DIRIN READ IN DIRECTORY SECTOR 5050 LDY #\$00 INIZ 5060 MOV5 LDA DIRBUF,Y FETCH DIR. ENTRY	4960	INC STGAP	MOL BUMP DEST. TR. #
4980 BNE MOVE AND LOOP1 => MOVE 4990 MOV4 LDA #\$01 INIZ 5000 STA SECT SET SECTOR 1 5010 LDA GAPLEN GET BLOCK LENGTH 5020 JER EBCD MAKE IT BCD1 5030 STA GAPLEN AND PUT IT BACK 5040 MOV9 JER DIRIN READ IN DIRECTORY SECTOR 5050 LDY #\$00 INIZ 5060 MOV5 LDA DIRBUF, Y FETCH DIR. ENTRY	4970	INC STRLK	BOMP ORIGIN TK. # 7000
4990 MOV4 LDA #\$01 INTE 5000 STA SECT SET SECTOR 1 5010 LDA GAPLEN GET BLOCK LEAGTH 5020 JER HBCD MAKE IT BCD1 5030 STA GAFLEN AND PUT IT BACK 5040 MOV9 JER DIRIN READ IN DIRECTORY SECTOR 5050 LDY #\$00 INTE 5060 MOV5 LDA DIRBUF, Y FETCH DIR. ENTRY	4980	BNE MOVE	AND LOOP! ==> MONE
5000 STA SECT SET SECTOR 1 5010 LDA GAPLEN GET BLOCK LENGTH 5020 JER HBCD MAKE IT BCD1 5030 STA GAFLEN AND PUT IT BACK 5040 MOV9 JER DIRIN READ IN DIRECTORY SECTOR 5050 LDY #\$00 INIZ 5060 MOV5 LDA DIRBUF,Y FETCH DIR. ENTRY	4990 MOV4	IDA #SOI	TNT7
501.0 LDA GAPLEN GET BLOCK LENGTH 5020 JER HBCD MAKE IT BCD1 5030 STA GAPLEN AND PUT IT BACK 5040 MOV9 JER DIRIN READ IN DIRECTORY SECTOR 5050 LDY #\$00 INIZ 5060 MOV5 LDA DIRBUF,Y FETCH DIR. ENTRY	5000	STEA SECTO	इस्य दक्ष्याता ।
5020 JER HECD MARE IT ECD1 5030 STA GAPLEN AND FUT IT BACK 5040 MOV9 JER DIRIN READ IN DIRECTORY SECTOR 5050 LDY #\$00 INIZ 5060 MOV5 LDA DIRBUF,Y FETCH DIR. ENTRY	5010	LDA CAPLEN	CET BLOCK LENGTH
5030 STA GAPLEN AND PUT IT BACK 5040 MOV9 JER DIRIN READ IN DIRECTORY SECTOR 5050 LDY #\$00 INIZ 5060 MOV5 LDA DIRBUF,Y FETCH DIR. ENTRY	5020	TER HRCD	MAYE TO DON'T
5040 MOV9 JER DIRIN READ IN DIRECTORY SECTOR 5050 LDY #\$00 INIZ 5060 MOV5 LDA DIRBUF,Y FETCH DIR. ENTRY	5030	OTA CAPTEM	TANK TAND TANDERS OF THE PARTY
5050 LDY #\$00 INIZ 5060 MOV5 LDA DIRBUF,Y FEICH DIR. ENTRY	50/0 MOK7A	אבה טויבנינע	PEAD TH PEDECEMBER COMMON
5060 MOV5 LDA DIRBUF, Y FEICH DIR. ENTRY	5050 17079	TUN 46VV	THE THE PROPERTY SECTOR
DOOR BOAD TWW DIRECL'X REICH DIE WALKA	EDEO MONTE	TOY DIDDING	TNT
	בעראון המחר	TOWN DIRECTORY	EDICH DIK. ENTRY

5070 5080 5090		CMP #1# BBQ MOV6 TYA	EMPTY EMIRY? YES! => MOV6 NO! PUT INDEX IN ACC.  POINT TO 1ST TRACK # PUT BACK IN Y FETCH 1ST TK. OF FILE MAKE IT HEX COMPARE TO START OF BLOCK SAME! => MOV11 LESS THAN! => MOV6  COMPARE TO END OF BLOCK SAME! => MOV11 GREATER THAN! => MOV6 SET BCD MATH MODE REFERCH TRACK #
5110		ADC #\$06	POINT TO 1ST TRACK #
5120		TAY	PUT BACK IN Y
5130		LDA DIRBUF.Y	FETCH 1ST TK. OF FILE
5140		JSR BCDB	MAKE IT HEX
51.50		CMP POINT	COMPARE TO START OF BLOCK
5160		BEO MOVII	SAME! ==> MOVII
5170		BCC MOV6	LESS THAN! => MOVE
5180		CLC	
51.90		CMP EMBLK -	COMPARE TO END OF BLOCK
5200		BEO MOVII	SAME! => MOVI)
521.0		BCS MOV6	GREATER TEAN! => MOV6
5220	MOV11	SED	SET BCD MATH MODE
5230		LDA DIRBUF.Y	REFETCH TRACK #
5240		SEC	
5250		SBC GAPLEN	SUBTRACT GAP LENGTH
5260		STA DIRBUF, Y	PUT RESULT BACK IN ENTRY
5270		LDA DIRBUF+1,Y	SUBTRACT GAP LENGTH PUT RESULT BACK IN ENTRY FETCH LAST TRACK #
5280		SEC	SUBTRACT GAP LENGTH PUT BACK RESET TO NORMAL MATH MOVE TO NEXT ENTRY
5290		SEC GAPLEN	SUBTRACT GAP LENGTH
5300		STA DIRBUF+1, Y	PUT BACK
5310		CTD	RESET TO NORMAL MATH
5320	MOV6	TYA	MOVE TO NEXT ENTRY
5330		AND #\$F8	
53 40		AND #\$F8 CLC	
5350		ADC #\$08	
5360		TAY	AT END OF DIRBUF? NO! LOOP! => MOV5 YES! WRITE REVISED DIR. CHECK DIRECTORY SECTOR # DONE BOTH ?
5370		BNE MOVS	NO! LOOP! => MOV5
5380	MOV7	JSR WRITE+3	YES! WRITE REVISED DIR.
5390		LDA SECT	CHECK DIRECTORY SECTOR #
5400		CMP \$\$02	DONE BOTH ?
5410		BEO MOVE	ARRI ==> MONR
5420		INC SECT	NO! BUMP SECTOR #
5430		BNE MOV9	AND LOCP! ≈> MOV9
5440	8VOM	JMP GAP	NO! BUMP SECTIOR # AND LOOP! => MOV9 LOOP TO ALLOCATION CHECKEF
5450	:		LISTING NEXT MONTE

# <u>LETTERS</u>

BD 2

This program is for the OSI ClP or by changing addresses 0239 & 023A, it can run on other OSI machines. I have been using this program myself for quite a while and find it very useful.

# MnM Software Technologies, Inc.

· といかははないからはないないない。 とうまとれる はんかい か

416 Hungerford Drive, Suite 216 Rockville, Maryland 20850

# INTRODUCING OUR

The missing tools for the OS-65U system. Our products are written in 6502 native code and are compatible with 65U, single, timeshare or network modes. Floppy or hard disk systems.

Ky. ASM V1.1-ASSEMBLER (Virtual source files, superfast, many extra features including a label table) ...\$129 (manual \$25)(50 pgs.)

Ky. COM V1.5-COMPILER (Configures Itself to V1.2 or 1.42, dynamic variables and arrays DIM A (N), supports machine language routines at hex6000, last 2 pages in high memory accessible, debug with interpreter and compile in 2-3 minutes. Protect your valuable source routines, gain as much as 2-10 times on average programs in execution speed. Supports 'INPUT[and 'PRINT[on the 1.42 system. ....\$395 (manual \$25)(110 pgs.)

Ky. DEV I-ASSEMBLER AND COMPILER TOGETHER....\$474(manual \$40)

KEYMASTER I V1.0-The word processing missing link for OS-65U based systems. KEYMASTER I is screen oriented, menu driven, simple to use yet highly advanced. KEYMASTER I contains most of the best features only found in dedicated work processing systems. Ask for the features you have been looking for and the answer will most likely be "YES!" To be released in February...Introductory price \$475 (Manual \$25)

All software comes with license agreement, registration card, manual, bloder, diskette holder and 8" diskette. Manuals are available by themselves and are deductible from full purchase price of software within 60 days after purchase., Foreign orders must be paid in U.S. dollars and drawn on a U.S. bank or international money order.

ALLOW 2 WEEKS FOR DELIVERY AFTER RECEIPT OF CHECK OR MONEY ORDER

CALL 301/279-2225



I know there have been a lot of search programs listed in many magazines, but most were in Basic, so here is mine. This one can be put in through the machine monitor and left in the unused portion of page 2 until it is needed.

The program can be called up by 0222 G then the variable or strings that you wish to find are typed in and a carriage return will produce the line numbers in the Basic program that contains what was typed in.

When you have typed, and after the carriage return, the program looks for the question mark (3F) that is printed out when anything is input without a line number. When the 3F is found, whatever is in the line buffer is stored and then compared with the Basic program until a match is found. At that time a line number is printed and the comparing goes on until the entire basic program is covered.

0222	A937	LDA	#\$37
0224	8504	STA	\$04
0226	A902	LDA	<b>#\$02</b>
0228		STA	
022A	A901	LDA	
022C		STA	
022F		LDA	#\$03
		STA	
	8DB902		
0234	4C74A2	JMP	\$A274
0237		PHA	AB 0 C F
0238		LDA	
023B	C93F		#\$3F
023D		BEQ	\$0243
023F		PLA	
0240	4CC3 A8	JMP	\$A8C3
0243	4CC3A8 A900	LDA	#\$00
0245 0248	8D0302	STA	\$0203
0248	A200	LDX	#\$00
024A	B513	LDA	\$13,X
024C	B513 9DE702	LDA STA	\$02E7,X
024F	C900	CMP	#\$00
0251	C900 F005	BEO	\$\$00 \$0258
0253	E0	INX	40230
0233	E8 E020		#\$20
0254	0200		
0256	DOF2	BNE	
	8EE602	STX	
	206 CA8	JER	
025E	A200	<b>LDX</b>	
0260	A000	LDA	
0262	B9E702	LDA	
0265	8DF402	STA	
0268		JSR	
026B	20AB02	JSR	\$02AB
026E	CDE402	CMP	\$02E4
0271	DOED	BNE	\$0260
0273	C8	INY	
0274	CCE602	CPY	\$02E6
0277		BNE	
	E004 .	CPX	
027B	30E5	BMI	
027D	A5F2	LDA	
027E		STA	
	8587	LDA	
0281	A5F3		
0283	8388	STA	\$88
0285	8A	TXA	
0286	48	PHA	
0287	98	AYT	
0288	48	PHA	

0289	205AB9	JSR	\$B95A
028C	68	PLA	
028D	A8	TAY	
028E	68	PLA	
028F	AA	TAX	
0290	4C6002	JMP	\$0260
0293	A9C3	LDA	#\$C3
0295	8504	STA	<b>\$04</b>
0297	A9AB	LDA	\$\$A8
0299	8505	STA	\$05
029B	4C74A2	JMP	\$A274
029E	ADB802	LDA	\$02B8
02Al	C57B	CMP	\$7B
02A3	ADB902	LDA	\$02B9
02A6		SBC	\$7C
02A8	BOE9	BCS	\$0293
02AA	60	RTS	
02AB	E000	CPX	#\$00
02AD	F008	BEQ	\$02B7
02AF	EEB802	INC	\$02B8
02B2	D003	BNE	\$02B7
02B4	EEB902	INC	\$02B9
02B7	ADFFFF	LDA	\$FFFF
02BA	8DE502	STA	\$02E5
02BD	E004	CPX	#\$04
02BF	1003	BPL	\$02C4
02C1	95F0	STA	\$F0,X
02C3	E8	INX	***
02C4	20D102	JSR	\$02D1
02C7	ADE502	LDA	\$02E5
02CA	38 E930	SEC	4020
02CB 02CD	38	SBC	<b>#</b> \$30
02CE		SEC	ACDO
UZCE	E9D0	SBC	#\$D0

02D0 60 RTS LDA \$02B8 CMP \$F0 02D1 ADB802 02D4 C5F0 02D6 F001 BEQ \$02D9 02D8 60 RTS 02D9 ADB902 02DC C5F1 02DE F001 LDA \$02B9 CMP SFI BEQ \$02E1 02E0 60 RYS 02E1 A200 LDX #\$00 02E3 60 02E4 00 02E5 00 RTS BRK 02F6 00

Robert Pendt Poestenkill, NY 12140

\* \* \* \* \*

## ED:

First things first; I have a C4 with a 8 slot backplane and power supply, a D&N CM9 memory (24K) & floppy controller, a 527 board (24K), a D&N IO-1600 and Radio Shack Line Printer 1 (i.e. Centronics 779) and dual MPI minifloppies.

I originally bought a 8K cassette C2-8P so I could learn more about computers by expanding it myself. (How's that line go, "if it wasn't for bad luck I wouldn't have any luck at all").

Adding memory was easy enough, but the disk was trouble. I couldn't read what I had just written to the disk. Eventually, I sent the drive, cable, and D&N controller back to D&N. They added a couple of capacitors to counteract a

ringing effect they felt was caused by the metal shield on the drive cable. Finally, everything seemed to be in order. Then I got a deal on a second drive, a D&N IO-1600 board and a RS LPl Printer. I put up with double line feeds for a long time until after rereading an old PEEK(65) and someone said that the Radio Shack basic interpreter didn't provide a line feed. Therefore, I figured it had to be on the printer (I had thought it was a problem in the OSI DOS). Sure enough, with the help of a service manual, I found a jumper wire that selects an automatic linefeed function. To disable it, change the jumper from El & E2 to El & E3. Hope this helps somebody.

Now for the problem at hand. I've been working with Planner Plus VI.1 (which runs under OS-65D v 3.2) inputting all my utility bills, etc. for the past seven years (trying to see where all the money goes), and time after time, the thing will start giving me error messages. I've tried new disks (and starting all over), I've tried typing real slow and I don't even dare touch the keyboard when its closing or accessing the files for fear it will get upset. I can hear the disk looking back and forth and then before the error is masked by the trap routine, I see an error I (parity error). I've had the drive checked and aligned, I've run memory tests afterwards and read all the old PEEK(65)s. Can anyone help me?

Where can I get a set of the OSI Tech Notes?

Does video Ram have to be as fast as memory Ram to run at 2Mhz?

By the way, I program like I write letters. First, I get an idea, then I write a couple of lines, then I rewrite, then I get a pencil and paper, flowchart it, and start over.

Thank heavens PEEK(65) is independent of OSI, keep up the good work.

Craig S. Borst Holland, MI 49423

Craig:

Contact your local dealer or OSI for a set of the Tech Notes.

Al

\* \* \* \* \*

1

# ISOTRON, INC.

PROUDLY ANNOUNCES THE ACQUISITION OF

# OHIO SCIENTIFIC, INC.

# A MESSAGE FROM THE PRESIDENT

I am happy to report that, even at this very early date, ISOTRON, Inc. is fully committed to: build upon the dedicated base of OSI users and resellers, guarantee a continuing supply of machines, parts and service, and to press confidently toward the release of new products and software.

By next month, we will be in a position to advise you, in a more explicit fashion, of the details which have so excited our team.

We look forward to working with and for you to put OSI back at the technological forefront of the micro computer world.

Merry Christmas and a Happy New Year.

Robert Lewis

# ISOTRON, INC.

6515 MAIN STREET TRUMBULL, CT 06611 (203) 268-3116 I own an OSI C2-D with 52K memory, 7MB hard disk, single floppy, a Bazeltine 1420 terminal, and a NEC 5500-D letter quality printer all of which runs under 65U.

I would like to have it run CP/M but I don't know where to begin. I realize that I will need to purchase some new boards etc., but I don't know which ones to buy, where to buy them, how much they will cost, etc.. In addition, I would require a version of CP/M that will support the hard disk and the letter quality printer. Where will I get that? Any help you can give me will be appreciated.

John Beamish Ontario, Canada M6B 4A3

John:

You have several problems:

1) CP/M runs on the 8080/8085/Z80 CPU chip, which the C2-D does not have.

2) CP/M requires static RAM, which most C2-D's don't have.

3) CP/M and OS-65U store information in different disk formats, so files from the two systems can't be intermixed freely on the same disk.

There are several solutions: If you want to run both CP/M and OS-65U, you must buy a new 510 CPU board and at least supplement your RAM if not replace it completely, then buy the CP/M and OS-U which are compatible with each other from OSI. Dick McGuire uses this system every day.

If you don't want to run OS-U, you can buy the D&N-80 board from D&N Micro Products, which will let you run standard CP/M, read and write IBM 3740 format floppy disks — but won't let you use your hard disk at all ... yet. D&N has drivers written for the 37 and 74 Mbyte disks for their CP/M, and my guess is they will soon have one for the 7 Mbyte disk. Check with them. I use a D&N-80 board, with a D&N 64K RAM board and a D&N 1600 serial I/O board every day, and they work just fine.

A.

\* \* \* \* \*

# A LETTER TO OSIO & PEEK(65)

As OSI struggles to survive, those of us users who have not switched to other systems must band together for support. Our group in the Boston area has decided to rename our-

selves "OSI Users/Boston" to better define our status. We are now proceeding to search out others in the same situation.

We would like to explore the possibility of establishing a national Users Group to expand on the exchange of software and hardware. Preliminary thoughts indicate that it would be a federation of local groups, perhaps using PEEK(65) and Compuserve for information exchange. With gradually decreasing resources, we should not waste efforts supporting extensive local newsletters or other activities, but should pool our capabilities to the maximum extent possible.

In order to increase our ability to keep together, OSI Users/Boston is conducting a campaign to equip all our machines with modems. Our hardware types are working on a club project for an inexpensive home-built modem, while others are purchasing some of the cheap modems now available on the market.

We are not aware of local groups other than OSIO. There must be many who could join in this effort. If you have such a list, we would like your assistance in contacting them. A copy of this letter will go to PEEK(65), which is an essential ingredient of any such plan.

Please respond to this letter, either directly or in PEEK (65). We need discussion, followed by action!

William R. Hutchins OSI/boston 21 Winthrop Road Lexington, MA 02173

William:

We agree! Although it will take considerable effort on someone's part to get the ball rolling and the "someone" needs to be found, we concur with the principle.

PEEK(65) can, hopefully, help to make the road easier by being the forum and vehicle of communications. To be more specific, we can provide a User Group Column in which to print the following: 1. The name, address and contact of all groups, 2. List new groups as they occur, 3. Describe the special interests and projects of each group, and 4. Disseminate information from the National Group. Additionally, we will be happy to maintain this list and inform callers of groups in their area.

As regards the dissemination of general information and articles, we already have "swap" arrangements with several User Group News Letters and their material and group are given appropriate credit.

We don't know all of the groups. Others we know of, but not how to contact them. So we make a plea for information. With your help we can develope that list of groups and make it available to both readers and callers. Most important, let us hear from you.

PEER(65) Staff

ED:

Responding to the letter of Mr. Kent on pg 22 of Sept '83, I bypass the floppy drive disable switch with a 0.1 MF (or MFD), 600 V capacitor. This prevents system crash.

I'm a consultant, and I have two OSI C3-OEM's that I use to develop hardware and software for 6800, 6502, and 280 systems. I also have an IBM PC, Apple II+, Apple IIe, Quasar HHC, and Kim 1.

Rick Miller Elgin, IL 60120

# AD\$

FOR SALE: C4PDMF with 48k RAM and 2 - 5 1/4" drives. Lots of OS-65D V3.2 software and utilities. OSI, Sams and V3.2 listing manuals. Very low hours on this like-new system. Asking \$700. Excellent 19" color monitor included if you pay for shipping of all. Bob Curran, RD 2, Box 35, Mohawk, NY 13407. Phone (315) 866-7271.

\* \* \* \* \*

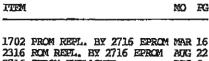
Small eastern Iowa OSI dealer going out of business for personal computers. Everything must go. Best offer takes all or part. Call Steve at (319) 396-2415 after 4 Central Time or all day Saturday.

FOR SALE: OSI C2-OEM computer, 48k, dual 8" floppy. Televideo 920c terminal w/built-in modem. 65D 3.3 and 65U 1.43 operating systems w/manuals and extra system documentation. Misc. programs, extra diskettes w/cases. Asking \$1600. Will pay shipping. Phone 703-942-2702.

\* \* \* \* \*

32K ClP Series 2 Single Disk AD\$ continued on page 23

MO PG



CIP, SBII ADD 8K RAM

C2-4 MONITOR UPGRADE 2716 C2-4P MOD FOR 540 BOARD C8P USER MANUAL ERRATA

CBASIC - THE FUZZY THEORY CBS LABEL OPTION PAK REVIEW

CD-28 CYL. WRAP-AROUND FIX CLOCK FIX FOR 2 MBZ

FEB 12 JAN 12

JUL 21 APR 16

MAR 8

SEP 19 OCT 19 PROGRAMMING TIPS

REGRESSION PROGRAM

REVIEW 300 SERIES APR 5
REVIEW DBI MULTI PROC. BOARDS SEP 11

R65C02

COMPUSERVE ASCII FIX COMPUSERVE OSI-SIG INFO COMPUSERVE PROTOCOL CP/M CHANGING PRINTER PORTS JUN 23 CP/M, ETE/ACK
CP/M INCREASING TO 64K NEM MAY 13 MAY 23 CURSOR ADDRESSING - 65U V1.3+ DIR IMPROVED PROGRAM MAR 12 2716 EPROM EXPLAINED 2716 EPROM MONITOR UPGRADE MAR 10 DEC 6 DISK FORMATS XLATE OSI/IBM/ DISK STORAGE UNDER ROM BASIC SEP 22 JAN 22 JAN 12 2716 EPROM SWAP FM 1702 PROM MAR 16 2716 EPROM SWAP FM 2316 ROM AUG 22 DMS, CUSTOMIZED KEY FILE DUMP FEB 17 65D 3.2 IMPROVED RANDOM FILES JAN 18 65D BASIC, BLOCK DELETE FUNCT FEB 2 65D BEXEC\* IN 1 TRACK IMPROVD JUL 10 65D BEXEC\* IN 1 TRACK PROGRAM JUN 16 DMS OLD TO NEW FILE LOADER PR JUL 11 DMS REPORT WRITER - TITLE MAR 6 OCT 23 DOS/65 COMMENTS EDMAFL ADDS A NEW FEATURE JAN 20 65D DIR, SORTED PROGRAM 65D DIRECTORY RESTORER EDMAFL ADDIT PRINTER FIX EDMAFL LABEL SEARCH TAN 21 FEB 9 EPROM POWER SUPFLY
EPROM PROGRAMMER, CLP PROJECT NOV
EX/MON ADDITIONS, MORE
SEP 65D DISABLE COLON, COMMA FIX 65D DISK INDENTIFICATION SEP 23 APR 15 65D DISK INDENTIFICATION, MORE JUL 22
65D DISK OP. SYSTEM NOTES JUN 6
65D EXECUTIVE ROUTINE JAN 2
65D BOOKS INTO BASIC VI.8 DEC 8 SEP 4 EXT. MON. DUMP PATCE JUN 21 FBASIC MICROSOFT - CORRECTION JUL 23 FBASIC VS MICROSOFT JUN 10 65D INDIRECT FILES 65D INDIRECT FILES, MORE 65D MACHINE LANGUAGE DIR MAY 8 FD STEPPING RATES SEP 22 SEP 22 MAR 18 FD TURN-OFF PROBS FIXED JUL 21 NOV 9 FORTRAN 65D SENI ADTO FILE CREAT FROG JUL 2: 65D V3.3 KEYBOARD ROUT, LOC. JUL 2: 65D V3.3 KEYBOARD SCAN JUN 2: FORTRAN, COBOL & BASIC ON 300 SEP 18 JUL 22 HAM NET JUL 22 HAYES CHRONOGRAPH - CLOCK USE APR 22 HAYES CLOCK PATCH TO WP6502 JUN 23 HEATEH 14 FRINTER FIX JAN 22 JUN 23 65D V3.3 TIDBITS APR 16 65U 1.3+ TERM OTHER THAN 1420 APR 19 65U 1.42 L3 FILE SIZE & ADDR. JAN 10 65U 1.42 L3 SEMAPHORE CHECK JAN 10 HEX LOADER PROGRAM MAR 6 HEXDOS CLPMF DISK BACK-UP HEXDOS PROGRAM COMPRESS PROG BEXDOS READS 65D FILES PROG. MAR 19 650 CD-28 CYL WRAP-AROUND FIX SEP 19 650 MEMORY LOCATION LIST MAR 9 SEP 22 65U SUBS AND UPS\$ JAN 6 65U V1.3+ CURSOR ADDRESSING MAR 12 65U V1.3+ EXTENDED INPUT APR 21 65U V1.43 230E TIMESHARE NOTE JUN 2 HEXDOS REVAME PROG. HI RES INSTALLATION MAR 22 MAR 12 HI RES-DO YOU REALLY WANT IT KEYCALC - PLANNER FEATURES MAR 18 SEP 18 65U VI.43 25UE TIMESHARE MU 65U VI.44 FEATURES 6809 ON OSI, SOME THOUGHTS APPLE II ON OSI ASCII CHARACTERS - PRINTING ASCII CHARACTERS - PRINTING LARGE SCALE BACK-UP ALTERNATE ADG 22 MA/COM TO KENDATA JAN 2 JUN 21 MF ADD A DISK DRIVE DEC 20
MF 'BREAK' & DISK CONTROLLER FEB 22 MAR 20 MAR 11 MF DISK RPM TIMER AUG 12
MF DISK RPM TIMER CORRECTION NOV 23 MAR 22 ASR 33 PRINTER HOOKUP CLP JAN 23 ASSEMBLY LANG WITH ROM BASIC AZIMUTH READING PROGRAM APR 5 MF DISK SWITCH TRURN-OFF MICROSOFT FBASIC CORRECTION MAY 19 JUL 23 MICROSOFT FBASIC CORRECTION
MICROSOFT VS FBASIC
MICROSOFT
MICROSOFT VS FBASIC
MICROSOFT
MICROSOF AZIOTE RESULTO PROGRAM
BASIC, INTERNAL STORE CHANGE
BASIC, INTERNAL STORE FORMAT
BASIC, OPTINIZING PART I
BASIC, OPTINIZING PART II
BASIC PROGS UNDER ROM BASIC
BAUD RATE MOD FOR CLP MAY SEP 17 OCT 17 MAR 18 AUG 6 BAUD RATE SWITCH 300/600 SEP 21
BEXEC\* IN 1 TRACK IMPROVED JUL 10
BEXEC\* IN 1 TRACK PROGRAM JUN 16
C19 PARALEL PRINT. INVER. EXPAN FEB 3
C1P 16 PORT PARA. I/O PROJECT AUG 15
C1P ASSEMBLER MERGE JUL 22 MODEM VS 12 % 48 VID PROBLEM NOV 22 MON ROM IMPROVEMENTS FOR SBII MAR 21 MONITOR CONVERSION TO TV INFO MONITOR UPGRADE 2716 EPROM JAN 12 CIP BAUD RATE MODIFICATION CIP CLOCK FIX FOR 2 MHZ CIP DATA SEPAR. CORRECTION CIP DATA SEPARATOR FOR SASI MORSE CODE PROGRAM MULTI GUESS TEST PROG D V3.3 ADG 6 OCT 19 SEP 2 AUG 18 OSI SIG COMPUSERVE INFO PER 23 PASCAL MAY 15 MAR 18 CIP DISK BOOT ROUTINE
CIP EPROM PROGRAMMER PROJECT PLANNER PLUS V4.X FIX PRINTED REPORT RESTART/65U PEB 23 NOV 2 CLP INVOICE FROGRAM CLP MAILING LIST PROGRAM PRINTER EXTRA LINEFEED PRINTER EXTRA LINEFEED FIX MAR 18 JUL 14 MAR 20 JUL 8 CIP MEMORY MAP EXPLAINED
CIP SA400 ADDITION PRINTER EXTRA LINEFEED FIX PRINTER EXTRA LINEFEED FIX MAY 22 JUL 20 FEB 22 CIP STRUCTURED EXPANSION AUG 11
CLPTI VIDEO SWAP/MODEM PROB. NOV 22
CLPMF 65D EXMON FIXES PLUS SEP 9
CLPMF WORD PROCESSOR IN BASIC DEC 2 AUG 11 NOV 22 PRINTER EXTRA LINEFEED FIXED PRINTER HOOK-UP ASR33 ON CLP MAR 19 JAN 23 PRINTER TRS FIX
PRINTING ACII CHARACTERS
PRINTING ASCII CHARACTERS
PROGRAMMING TIP FOR KEYBASIC
PROGRAMMING TIPS MAR 19

REVIEW FIG FORTH FOR 65D ADG 2 REVIEW GANDER FINANCIAL PLANN MAY REVIEW GENERIC MEM+ BOARD AFR APR 10 REVIEW, KEYWORD REVIEW OSI GREATEST HIPS V2 JUN 10 REVIEW, EYSTEMS GENERATOR OCT 6
REVIEW TECO TEXT EDITOR TEC65 OCT 2
REVIEW, TIME & TASK PLANNER DEC 11 ROM BASIC ASSEMBLY LANG ROM BASIC DISK STORAGE APR 5 ROM BASIC DISK STORREGE JAN 22
ROM BASIC EXTENSION PROCESSOR MAY 9
ROM COLD START INFROVED APR 11
SBII MON ROM IMPROVEMENTS MAR 21
SINGLE SWITCH TURN-ON P MACH. JAN 16 SOFTWARE LISTINGS SOFTWARE LISTINGS OCT 11 SUB\$ AND UP\$ - 65U SUPERBOARD SECRETS JAN 6 TERMINAL PROG, SMART - 65U TERMINAL PROG, SMART - 65U SEP 11 TERMS OTHER THAN 1420 U VI.3+ APR 19 TV TO MONITOR CONVERSION INFO JAN 22 JUN 17 DSR FUNCT. THE ULTIMATE FEB 14 NOV 22 UTI, FUN WITH VID PROB 12 X 48 WITH MODEM VIDEO DISFLAY FIX MAR 16 VIDEO DISPLAY FIX MAR
WORD PROCESSOR IN BASIC MF DEC
WP-1 REMOVING LINEFEED JUL
WR3.3 V1.01 FIX FOR LEVEL 2 SEP
WF3.3 V1.01 FIX FRINT DV #5 SEP
WF6502 1.3A BUGS NOTED JAN
WF6502 C1/4P RIGHT JUST. MORE AUG DEC 2 JUL 20 SEP 18 JAN 22 WE6502 CL/4P RIGHT JUSTIFIED JUN 13 WE6502 MF WITH SELECTRIC JUN 20 WP6502 RELOCATING #2 (ROM) DEC 7 WP6502 RELOCATION \$1 (ROM) NOV 4 \* \* \* \* \* \* \* \* \* \* \* \* \* \*

AD\$ continued on page 22 Drive System OS-65D3.3 with extended monitor / assembler. Excellent condition. Full documentation, Sams Manual, best offer. AIS, 3517 Dunedin Dr. #204, Chesapeake, VA 23321. 804-484-8856.

\* \* \* \* \*

MAY 22 FEB 23

48K, C4P-MF, full documentation, v 3.3 DOS, heavy duty supply, some software, Assembler and Ext Monitor, mint condition, \$675, will ship. After 5 PM call 512-681-1983, San Antonio, TX.

Please write or call for free Please write or call for free catalog listing of OSI compatible software products. This month's special MUSIC GENERATOR \$49.00, includes The Little Fugue by Bach and A Mighty Fortress. Aurora Software Associates, 37 South Mitchell, Arlington Heights, IL 60005, 312-259-3150.

MAR 22

NOV 20

SEP 21

OCT 21

FEB 23 JUN 23

OSI C3 48K Memory with additional 8K on I/O Board CA-18A w/dual 8" Shugart floppies in separate enclosure in Mint condition. Includes Lifeboat 2.2 CP/M, OS-65D/65U operating systems, Microsoft MBasic and Fortran Software with all manuals and hardware documenta-tion. \$1500. Brad Miller, 212-490-0535.



The Unofficial OSI Users Journa

P.O. Box 347 Owings Wills, Md. 21117 BULK RATE U.S. POSTAGE PAID Owings Mills, MD PERMIT NO. 18

# DELIVER TO:

ISCOCNEYEROUNG 3001\*0404:C DG WILBY, ER. 3001 LINDA EM. SIMKING SPRING, PA. 10668

# GOODIES for DSI Users!

PER CO

P.O. Box 347 · Owings Mills, Md. 21117 · (301) 363-3268

(	)	C1P Sams Photo-Facts Manual. Complete schematics, scope waveforms and board photos. All you studied to be a C1P or SII Wizard, just \$7.85 \$			
{	>	C4P Sams Photo-Facts Manual. Includes pinouts, photos, schematics for the 502, 505, 527, 540 and 542 boards. A bargain at	\$15.00 \$		
(	)	C2/C3 Sams Photo-Facts Manual. The facts you need to repair the larger OSI computers. Fat with useful information, but just	\$30.00 \$		
{	)	OSI's Small Systems Journals. The complete set, July 1977 through April 1978, bound and reproduced by PEEK (65). Full set only	\$15.00 \$		
{	}	Terminal Extensions Package - lets you program like the mini-users do, with direct cursor positioning, mnemonics and a number formatting function much more powerful than a mere "print using." Requires 65U.	\$50.00 \$		
(	}	RESEQ - BASIC program resequencer plus much more. Global changes, tables of bad references, GOSUB'S & GOTOs, variables by line number, resequences parts of programs or entire programs, handles line 50000 trap. Best debug tool I've seen. MACHINE LANGUAGE - VERY FASTI Requires 65U. Manual & samples only, \$5.00 Everything for	\$50.00 \$		
(	)	Sanders Machine Language Sort/Merge for 0S-65U. Complete disk sort and merge, documentation shows you how to call from any BASIC program on any disk and return it or any other BASIC program on any disk, floppy or hard. Most versatile disk sort yet. Will run under LEVEL I, II, or III. It should cost more but Sanders says, "sell it for just"	\$89.00 \$		
(	)	KYUTIL - The ultimate OS-DMS keyfile utility package. This implementation of Sander's SORT/MERGE creates, loads and sorts multiple-field, conditionally loaded keyfiles. KYUTIL will load and sort a keyfile of over 15000 ZIP codes in under three hours. Never sort another Master File.	\$100.00 \$		
,	}	BOOKS AND MANUALS (while quantities last) 65V Primer. Introduces machine language programming.	\$4.95 \$		
t	)	C4P Introductory Manual	\$5.96 \$		
1	;	Basic Reference Manual — (ROM, 65D and 65U)	\$5.95 \$		
!	)	C1P, C4P, C8P Users Manuals — (\$7.95 each, please specify)	\$7.95 \$		
(	)	How to program Microcomputers. The C-3 Series	\$7.95 \$		
l	)	Professional Computers Set Up & Operations Manual — C2-OEM/C2-D/C3-OEM/C3-D/C3-A/C3-B/C3-C/C3-C'	\$8.95 \$		
(	١	Cash enclosed ( ) Master Charge ( ) VISA		\$	
Account No Expiration Date MD Residents			id 5% Tax	\$	
	Signature		C.O.D. orders add \$1.65		
	_	Postage & Hand	ling	\$_3.50	
St	ree	TOTAL DUE		\$	
Ci	ty .	State Zip POSTAGE MAY	VARY FOR OVERS	EAS	

24