# PEEK (65)

The Unofficial OSI Users Journal

★★ $1.75 ★★

November 1981
Vol. 2, No. 11

P.O. Box 347
Owings Mills, Md. 21117
(301) 363-3267

**INSIDE:**



## Column One

Who are you? What do you do for a living? What do you do with your OSI computer? What sort of software and hardware do you need to use it better? What sort of articles would you like to see in PEEK(65)?

These are important questions. Problem is, we have only a very vague idea concerning the classes of people who are using the hardware and software, and, more importantly (to us) reading PEEK(65). We know you are:

--Hobbyists who are very knowledgeable about programming;

--Programmers working for businessmen, also very knowledgeable, though probably about different aspects of programming;

--Businessmen who own OSI computers and really couldn't care less about what makes them work, so long as they earn their keep.

But is there also a large population of personal computer users out there, guys who don't know from programming, but use your computers for games, home control, etc.?

And just how many businessmen would really like to learn enough about your computers to at least know what you need, if not exactly how to get it?

Let me give you a frinstance of this last class. A.Z. (his real initials) runs a small business, and bought an OSI computer to help him. He quickly realized that to get the most out of his machine, he needed both a programmer and an experienced computer operator/bookkeeper. He hired both, and was in reasonable shape.

Then one day, for reasons having nothing to do with the computer or AZ, both guys quit. Wow. AZ now had a nice box, and a lot of disks with programs on them he didn't know how to use or modify.

Now AZ has signed up for a programming course at the community college, and has also hired himself another consultant programmer. The new programmer is doing most of the work on the computer, but AZ himself is learning how to operate it, and is asking a lot of very intelligent questions. No more is the computer considered to be another office machine which AZ never touches. No matter how long the consultant is around, AZ will know how to make that computer work.

How many more like him are out there? I have a feeling there are lots of you, and that we should be publishing more articles like Ken Holt's series on 65U which is continued in this issue.

Do drop us a line and let us know what you need. Don't assume that the contents of PEEK(65) are fixed. We will print what you want and need to read.

While we are on the subject of stuff to be printed in PEEK(65)... it seems that, in a way, we have been too successful. Our advertisers have learned that PEEK(65) is the best way to reach the OSI community, and as a result, we have the full capacity of the magazine booked up, ad-wise, for several issues in advance, with some scheduled through next September.

There are three possible answers to this problem. We could raise the prices of the ads. We could increase the number of ads per page of editorial copy. Or we could increase the editorial copy, and the size of the magazine, and the usefulness to all our readers, and fit in the extra ads with lots of extra copy.

We refuse to increase the number of ads per page of editorial copy. We are very reluctant to increase our ad rates. So it looks like the only answer is to put in more editorial copy and increase the size of the magazine.

That is where you come in. Last month I put out a Call for Articles. This month I am repeating and expanding it. I don't just want articles for novice users and business people. I want articles for everybody. I want letters to the editor. I want cartoons. I want whatever you want to send us. If it is useful to you, assume it will be useful to others and send it in!

I promised that the December issue will be a Christmas present to you, our friends. Just a couple of hints about what you will find in it:

The winner of the "logic in parens" boolean logic contest;

The index to this year's PEEK(65).

ae

OS65D3 #3 IN A SERIES
by D.R. "STRETCH" Manley
,890-Rogue River, OR 97537

## Increase Your 8" Disk Storage

There are ways to increase the amount of storage on the 8" disk drive. The 5 1/4" drive operating system seems to use its disk space pretty well, but the 8" system is very wasteful.

If you check in the OS65D3 manual, you will find that it states that a track can contain any mixture of various length sectors, as long as the total length in pages doesn't exceed 12 ($0C). But, it also says that if only one sector is written, it can be 13 ($0D) pages long. Here's where we can tighten up the storage.

If you boot up OS65D3 and exit to the operating system, you will find that the command "DIR TT" where TT is a track that has a data file on it will get you an answer of "01-0C", meaning 1 sector of 12 pages. A "DIR TT" where it is a track of a program file will get you a "01-0B" answer, showing that only 11 pages are stored on each track. What we will do is change BASIC and the operating system to always store 13 pages on each track. Thus your programs will store in approximately 85% of the normally required space, and your data files will store in approximately 92% of the normally required space.

Is 13 pages per track reliable, you ask? Sure! OS65U uses that format for its program and data file storage on 8" disks. Why not us?

To look at it another way, if you have a file of 150 records (random access), then it would normally take 7 tracks to store. With this improved storage technique, you can get the same 150 records on 6 tracks. This is because each track now will hold 26 records, instead of 24.

I have been using this technique for several months, and it has helped keep the cost of disks down. I can keep a lot more stuff on the same number of disks.

BASIC must be changed in 2 places, and the operating system in 4. The first change is to location $2133. This is luckily in an overlay, which is located at track 8, sector 3 in a normal system. This makes "EXIT" figure the new size in tracks. If you use the Extended Monitor, here's the dialog:

```
:!CA 40C4=08,3
:@4133
4133/0B (ENTER 0D, <CR>)
:!SA 08,3=40C4/1
```

If you are using more than 2 directory sectors, adjust the call and save sector parameter accordingly.

The next location that has to be changed is $2F0A. This is also in an overlay, the GET/PUT overlay. This changes the records per track for random access to 26. The Extended Monitor dialog is:

```
:!CA 2E79=08,4
:@2F0A
2F0A/18 (ENTER 1A, <CR>)
:!SA 08,4=2E79/1
```

BASIC is done. To change the operating system, we will have to use the track zero utility. These changes expand the file buffers, and move #7's buffer up 1 page, so that it doesn't interfere with #6's buffer. The last one changes the "PUT" to use 13 page segments, when it writes to disk. The Extended Monitor dialog for all the changes is as follows:

```
:EXIT
A*CA 0200=01,2
A*GO 0200
(CHOOSE 2, R4200, AND E)
A*RE EM
:@4329
4329/3D (ENTER 3E, <CR>)
:@432F
432F/3D(ENTER 3E, <CR>)
:@4331
4331/49 (ENTER 4B, <CR>)
:EXIT
A*GO 0200
(CHOOSE 2, W4200/2200,8, AND E)
A*RE EM
:!CA 4A00=01,1
:@4BE9
4BE9/0B (ENTER 0D, <CR>)
:!SA 01,1=4A00/5
```

Now the hard part begins! You will have to change the following programs to reflect the new buffer size:

1. CREATE
2. ZERO
3. CHANGE

Plus any others that are manipulating the buffer(s) directly.

You will have to change all your programs that use files to the new buffers. If they have 2 buffers, use the "LIST#6" idea in article #2 of this series. If they have 1, you will have to move buffer 6 to the top of RAM first, since the new file buffer size is 1 page longer, and opening a file will write over the first part of your BASIC program, hanging the system.

If you have programs that don't use files, the conversion is simple. "LOAD" the program, and then "PUT" it back to disk. The system is smart enough to load at 11 pages per track, and put at 13, with no coaching.

Conversion of data files is another story. You can't access a file with 12 pages per track correctly, when the system is expecting 13. You can either POKE new values before and after the access, or write a program to convert the files for you. This varies from simple to quite complex. Random files are easier than sequential.

The operating system figures the size of file buffer to write from the page parts of the buffer pointers. Those were the values we changed.

If you want a listing of a converter program that I use for random data files, send a stamped, addressed envelope to me and I'll send a printed listing along. It's for a 2 drive 8" system. Let me know what kind of system you have, what your favorite uses for it are, etc. I would like to get to know more of you.

**** ERRATTA ****

In article 1, a couple paragraphs got left out somewhere between me and the final product. In bug fix #5, the only fix printed was for programs that had no (0) file buffers. The fixes for one (1) and two (2) buffers are as follows:

For 1 buffer, type in POKE15101,32:POKE15102,30:POKE 15103,50(CR).
For 2 buffers, type in


PEEK (65) is published monthly by DBMS, Inc.
Owings Mills, MD 21117.
Editor - Al Peabody
Technical Editor - Dickinson H. McGuire
Asst. Technical Editor - Brian Hartson
Circulation & Advertising Mgr. - Karin Q. Gieske
Production Dept. - A. Fusselbaugh, Ginny Mays
Subscription Rates

| | |
|---|---|
| US (surface) | $15 |
| Canada & Mexico (1st class) | $23 |
| So. & Cen. America (Air) | $35 |
| Europe (Air) | $35 |
| Other Foreign (Air) | $40 |

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 Mills, MD 21117

POKE17149,32:POKE17150,30:POKE 17151,50(CR).

The rest is as printed.

**** COMMENTS ****

A word to Mr. Votaw, whose letter appeared in the September issue: letters like that are very encouraging. Thanks, and here's a way for you to make the initial 'RUN"BEXEC*' string into anything you want.

Mr. Votaw, you are right in your comment on my cautions. What I meant and what I said weren't the same, unfortunately. If you change the name of BEXEC* on disk, and not in the operating system buffer, it won't run. If you change them both, then no problem. In fact, you can make the 'RUN"BEXEC*(CR)' string in the operating system longer. You just have to relocate it a few bytes, and change the pointers to it to reflect the changes. The buffer is at $2E1E to $2E2F, a length of 18 characters. You can get 'RUN"13 CHARACTER FILENAME(CR)' in that much space. Unfortunately, the one quotation mark and the carriage return at the end are needed, or you could get 14 character filenames to fit in this buffer.

Anyhow, as it now stands, when the operating system is booted in, the input device flag is set to $10, decimal 16, or device #5. That's the memory input device, of course. The device #5 input pointer is at $238A (low) and $238B (high). It is set to $2E25, or the start of the string in the operating system buffer. Any of this can be changed to whatever you desire. It would be nice to have a long string in BASIC's buffer, but BASIC's buffer doesn't exist on disk. It is created in memory when BASIC is booted in. Nuts! You could use some other location, and change the boot routine and the operating system's initial state to reflect this new location.

**** COMMING ATTRACTIONS ****

I think that you will all agree that having only one random access file is not too useful. Next time I'll show you the code to convert OS65D3 BASIC to have two random files. Both #6 and #7 can be used either sequential or random. These will operate with no extraneous disk accesses, and a minimum of attention from the programmer. (No PEEKs and POKEs, normally).

I recommend that you get copies of the May 1981 BYTE, and September 1981 BYTE. The May issue tells how to speed up O.S.I.'s disk BASIC. The September issue has the corrections for the May article. This speed-up works (it increased the speed of a sort 10%), and opens up some memory to help hold the code for the two random files. It isn't essential for implementing the code for two random files, however.

STRETCH

★

## A "DOUBLE-BARRELED" DISK DIRECTORY FOR OS-65D

by Willis H. Cook
1298 Renee Drive
Lilburn, GA 30247

The standard DIR and DIRSRT directory utilities that come with the OS-65D operating system have one obvious shortcoming: they list the file names and track ranges in one column. Besides looking awkward, all scrunched over to the left, they are a nuisance when there are more files on a disk than there are display lines on the video screen. My monitor displays 26 lines, so when I have more files than that on a disk, the first ones have scrolled off the screen before the last ones are shown. There has to be a better way, and there is. With very little work, DIR and DIRSRT can be changed to display the directories in two columns, taking up only half the number of video display lines.

The listing below shows the modified version of DIR. The lines that have been changed or added are indicated by an asterisk (*). The modifications to DIRSRT are virtually the same, although the line numbers are different.

The modification works by setting up a counter, C, that is incremented every time a file name is read from the buffer and printed. Line 11095 checks to see if C is even or odd; if it is odd, the file name and track range are printed in the left column. If C is even, they are printed on the right. Adding another PRINT statement to print information starting in column 35, will break the directory listing into two columns, alternating from left to right. Don't forget to add the semicolon to the existing PRINT statement in line 11110 to allow for the second file name and track numbers on the same line.

The other changes are cosmetic: lines 10030 to 10055 center the directory title and print two headings for the two columns of information. I usually put the title of the disk in line 10030 to uniquely identify each disk. I also print one directory for each disk, cut it down to 7" by 7" and slip it into the disk folder. They are very handy when searching for a file without having to load each diskette and run a directory.

I have been told by the local OSI dealer that it is possible to print the directory in three columns, but have decided that it wasn't worth the trouble. I haven't put more than 52 files on a diskette yet, so my two-column version is adequate for me. A sample directory is included and shows how much more professional the two-column layout looks.

Regarding the problem mentioned by Mr. Arthur Goeres in the April issue of Peek (65), that allowed the creation of only 32 files instead of 64, the error is in line 580 of the CREATE program. In early version of OS-65D this line read:

```
580 DISK!"CA 2E79=08,2:
    GOSUB10000
```

It should be GOSUB 20000 and has been corrected by OSI in later issues of the operating system.

Listing on page 4.

"DOUBLE-BARRELED" DIRECTORY

```
 10 REM DIRECTORY UTILITY FOR OS-65D VERSION 3.2  NMHZ
 20 REM
*30 NF=0 : C=0
 40 PN=11897
 50 DEF FNA(X)=10*INT(X/16)+X-16*INT(X/16)
 80 DV=1 : Y=1 : X=PEEK(8994)
 90 IF X<=Y THEN 110
 100 DV=DV+1 : Y=Y+Y : GOTO 90
 110 PRINT "LIST ON LINEPRINTER INSTEAD OF DEVICE #";DV;
 120 INPUT A$
 130 IF MID$(A$,1,1)="Y" THEN DV=4
 10000 REM
 10010 REM PRINT A DIRECTORY OUT
 10020 REM
*10030 PRINT #DV : PRINT #DV,TAB(19)"OS-65D VERSION 3.2"
*10035 PRINT #DV,TAB(20)" -- DIRECTORY --" : PRINT #DV
*10040 PRINT #DV,"FILE NAME  TRACK RANGE";
*10045 PRINT #DV,TAB(35)"FILE NAME  TRACK RANGE"
*10050 PRINT #DV,"----------------------";
*10055 PRINT #DV,TAB(35)"----------------------"
 10060 DISK ! "CALL 2E79=08,1"
 10070 GOSUB 11000
 10080 DISK ! " CALL 2E79=08,2"
 10090 GOSUB 11000
 10130 PRINT #DV : PRINT #DVB,NF;"ENTRIES FREE OUT OF 64" :
       PRINT #DV
 10140 END
 11000 REM
 11010 REM READ DIRECTORY OUT OF BUFFER INTO ARRAYS
 11020 REM
 11040 FOR I=PN TO PN+248 STEP 8
 11050 IF PEEK(I)=35 THEN NF=NF+1 : GOTO 11130
 11060 N$=""
*11065 C=C+1
 11070 FOR J=1 TO I+5
 11080 N$=N$+CHR$(PEEK(J))
 11090 NEXT J
*11095 IF C/2=INT(C/2) THEN 11120
 11100 PRINT #DV,N$;TAB(12);FNA(PEEK(I+6));TAB(16);"-";
*11110 PRINT #DV,TAB(17);FNA(PEEK(I+7));
*11115 GOTO 11130
*11120 PRINT #DV,TAB(35);N$;TAB(47);FNA(PEEK(I+6));TAB(51);"-";
*11125 PRINT #DV,TAB(52);FNA(PEEK(I+7))
 11130 NEXT I
 11140 RETURN
```

```
                    OS 65-D Ver 3.2
                    -- DIRECTORY --
FILE NAME  TRACK RANGE            FILE NAME  TRACK RANGE
---------------------             ---------------------
OS65D3       0  - 8               BEXEC*       9  - 9
CHANGE       10 - 10              CREATE       13 - 14
DELETE       15 - 15              DIR          16 - 16
DIRSRT       17 - 17              RANLST       18 - 19
RENAME       20 - 20              SECDIR       21 - 21
SEQLST       22 - 23              TRACE        24 - 24
ZERO         25 - 26              ASAMPL       27 - 27
CLEARo       11 - 11              COLORS       28 - 28
C-ASM1       29 - 29              C-ASM2       30 - 30
EXEC         12 - 12              TEMP         70 - 76

  44 ENTRIES FREE OUT OF 64
```

Figure 1. A sample two-column directory.

★ ★ ★

```
10 REM   BOOLEAN CONTEST ENTRY
20 REM            BY
30 REM    W.G. LIBBY
40 REM    12 TRANMORE WAY
50 REM    CITY BEACH W.A. 6015
60 REM      AUSTRALIA
70 REM
500 INPUT"GUESS THE WORD";A$
510 X=A$="BOOLEAN"
520 IFXTHENPRINT"YOU'RE RIGHT"
525 PRINTX
530 GOTO500
```

submitted by:
Earl D. Morris
Midland, MI 48640

* * * * *

```
10 REM
11 REM  DEMONSTRATION OF SCREEN FORMATTING TECHNIQUES
12 REM
13 REM                    BY
14 REM                KEN HOLT
15 REM            H/B COMPUTERS, INC.
16 REM            217 E. MAIN STREET
17 REM        CHARLOTTESVILLE, VA. 22901
18 REM
19 REM                27 JULY 1981
20 REM
100 GOSUB 19000: IF CN THEN X=0: Y=0: FU=3: GOSUB 19900: GOTO
    63999
110 GOSUB 19110: IF CN THEN 100
120 X=0: Y=0: FU=3: GOSUB 19900
130 PRINT "INFORMATION RECEIVED:": PRINT
140 PRINT "  NAME: ";FV$(1): PRINT
150 PRINT "  ADDR: ";FV$(2)
160 PRINT "        ";FV$(3)
170 PRINT: PRINT: PRINT
180 PRINT "SEARCH CODE: ";FV$(4)
190 PRINT: PRINT: PRINT: GOTO 63999
19000 REM
19001 REM  SCREEN FORMATTING MANAGER
19002 REM
19003 REM
19004 REM  THE FOLLOWING ARRAYS ARE USED BY THE ROUTINE:
19005 REM
19006 REM  FS$(N,0) - FIELD TYPE
19007 REM  FS$(N,1) - TEXT OR SUBSCRIPT NUMBER (M)
19008 REM  FS(N,0)  - X COORDINATE
19009 REM  FS(N,1)  - Y COORDINATE
19011 REM  FS(N,2)  - FIELD LENGTH
19012 REM  FV$(M)   - FIELD VALUE M (SEE NOTE BELOW)
19013 REM
19014 REM  THE SPECIFICATIONS FOR INPUT AND OUTPUT FIELDS
      INCLUDE A
19015 REM  SUBSCRIPT VALUE.  THIS IS THE SUBSCRIPT USED TO
      DETERMINE
19016 REM  WHICH ELEMENT OF THE FV$ ARRAY WILL BE USED FOR THE
      VALUE.
19017 REM
19018 REM  A SLASH MAY BE ENTERED INSTEAD OF A FIELD VALUE.
      THIS WILL
19019 REM  BACK UP THE PROMPT TO THE PREVIOUS INPUT FIELD.
19021 REM
19022 REM  SEE SECTION STARTING AT 19500 FOR SCREEN FIELD
      DEFINITIONS
19023 REM
19024 REM  ENTRY POINTS:
19025 REM
19026 REM    19000 - FIRST USE (OR RETURN TO FIRST SCREEN)
19027 REM    19100 - NEXT SCREEN
19028 REM    19200 - REPEAT LAST SCREEN
19029 REM
19031 REM  RETURNED INFORMATION:
19032 REM
19033 REM      ARRAY FV$ CONTAINS FIELD VALUES, THE SPECIFIC
              SUBSCRIPTS
19034 REM      SUBSCRIPTS BEING SPECIFIED ON THE DATA STATEMENTS
               FOR
19035 REM      THE CORRESPONDING INPUT OR OUTPUT FIELD
               DEFINITIONS.
19036 REM
19037 REM      VARIABLE CN CONTAINS:  0  IF SCREEN WAS NOT
               CANCELLED
19038 REM                            -1  IF SCREEN WAS CANCELLED
19039 REM                                (A SLASH WAS ENTERED
                                         FOR
19041 REM                                THE FIRST INPUT
                                         FIELD)
19042 REM
19043 REM ------------------------------------------------------
19050 REM
19051 REM ** ENTRY POINT: FIRST USE (OR RETURN TO FIRST SCREEN)
19052 REM
19060 DATA $SCRFMT
19070 RESTORE
19080 READ FU$: IF FU$<>"$SCRFMT" THEN 19080
19100 REM
19101 REM ** ENTRY POINT: NEXT SCREEN
```

Last month, we saw a subroutine which helps to avoid the trap of hard-coded terminal-handling logic. This month, we'll expand on the idea to make a general-purpose screen "form" program.

The listing shown has three parts: the screen formatting manager at 19000, the screen formatting layout specs at 19500, and the terminal-specific screen handling code at 19900 (our old friend from last month.) If you took the hint last month, you've already re-written the 19900 subroutine for your flavor of terminal. If you didn't, you'll have to do it before you can make this month's program work.

The DATA statements at 19500 are how you specify the screen "form". There are four types of data patterns to be placed in the DATA statements. These are summarized in the REM's preceeding the DATA's. A heading specification defines a heading field on the screen; one which contains fixed information. The first item is the character "H" (for heading). The next two are row and column number to locate the beginning of the text on the screen. The last item is the text to be displayed on the screen. For example, 19520 DATA H,2,0, "NAME:" will display the heading NAME: at row 2, column 0.

An input specification defines an input field on the screen; one which is to receive operator input. The first item is the character "I" (for input). The next two are the row and column number to locate the beginning of the input field on the screen. The next item is a subscript value to be used for storing the input value into array FV$ (for field values). The last item is the maximum allowable field length. For example, 19530 DATA I,2,6,1,10 will provide for an input field of 10 characters at row 2, column 6. The input value will be placed in FV$(1).

An output specification defines an output field on the screen; one which is determined by the contents of a particular element of the FV$ array. The first item is the character "O" (for output). The next two are the row and column number to locate the beginning of the output field on the screen. The last item is a subscript value to be used for locating the desired

# OSI AARDVARK OSI
# NOW MEANS BUSINESS!

**WORD PROCESSING THE EASY WAY— WITH MAXI-PROS**

This is a line-oriented word processor designed for the office that doesn't want to send every new girl out for training in how to type a letter.

It has automatic right and left margin justification and lets you vary the width and margins during printing. It has automatic pagination and automatic page numbering. It will print any text single, double or triple spaced and has text centering commands. It will make any number of multiple copies or chain files together to print an entire disk of data at one time.

MAXI-PROS has both global and line edit capability and the polled keyboard versions contain a corrected keyboard routine that make the OSI keyboard decode as a standard typewriter keyboard.

MAXI-PROS also has sophisticated file capabilities. It can access a file for names and addresses, stop for inputs, and print form letters. It has file merging capabilities so that it can store and combine paragraphs and pages in any order.

Best of all, it is in BASIC (OS65D 51/4" or 8" disk) so that it can be easily adapted to any printer or printing job and so that it can be sold for a measly price.

MAXI-PROS — $39.95

**NEW-NEW—NEW TINY COMPILER**

The easy way to speed in your programs. The tiny compiler lets you write and debug your program in Basic and then automatically compiles a Machine Code version that runs from 50-150 times faster. The tiny compiler generates relocatable, native, transportable machine code that can be run on any 6502 system.

It does have some limitations. It is memory hungry — 8K is the minimum sized system that can run the Compiler. It also handles only a limited subset of Basic — about 20 keywords including FOR, NEXT, IF THEN, GOSUB, GOTO, RETURN, END, STOP, USR(X), PEEK, POKE, -,=,*,/, (,), <>, Variable names A-Z, and Integer Numbers from 0-64K.

TINY COMPILER is written in Basic. It can be modified and augmented by the user. It comes with a 20 page manual.

TINY COMPILER — $19.95 on tape or disk

**THE AARDVARK JOURNAL**

**FOR OSI USERS** — This is a bi-monthly tutorial journal running only articles about OSI systems. Every issue contains programs customized for OSI, tutorials on how to use and modify the system, and reviews of OSI related products. In the last two years we have run articles like these!

1) A tutorial on Machine Code for BASIC programmers.
2) Complete listings of two word processors for BASIC IN ROM machines.
3) Moving the Directory off track 12.
4) Listings for 20 game programs for the OSI.
5) How to write high speed BASIC — and lots more —

Vol. 1 (1980) 6 back issues - $9.00
Vol. 2 (1981) 2 back issues and subscription for 4 additional issues - $9.00.

**ACCOUNTS RECEIVABLE** — This program will handle up to 420 open accounts. It will age accounts, print invoices (including payment reminders) and give account totals. It can add automatic interest charges and warnings on late accounts, and can automatically provide and calculate volume discounts.

24K and OS65D required, dual disks recommended. Specify system.

Accounts Receivable. $99.95

*** * * SPECIAL DEAL — NO LESS! * * ***

A complete business package for OSI small systems — (C1, C2, C4 or C8). Includes MAXI-PROS, GENERAL LEDGER, INVENTORY, PAYROLL AND ACCOUNTS RECEIVABLE — ALL THE PROGRAMS THE SMALL BUSINESS MAN NEEDS. $299.95

P.S. We're so confident of the quality of these programs that the documentation contains the programmer's home phone number!

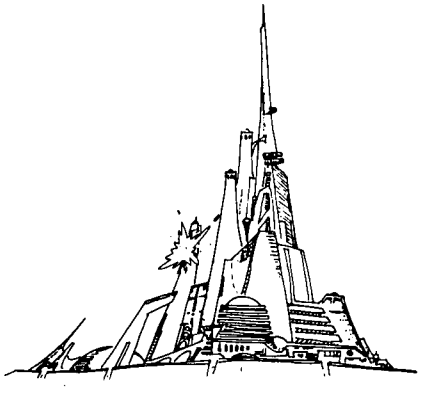
**SUPERDISK II**

This disk contains a new BEXEC* that boots up with a numbered directory and which allows creation, deletion and renaming of files without calling other programs. It also contains a slight modification to BASIC to allow 14 character file names.

The disk contains a disk manager that contains a disk packer, a hex/dec calculator and several other utilities.

It also has a full screen editor (in machine code on C2P/C4)) that makes corrections a snap. We'll also toss in renumbering and program search programs — and sell the whole thing for — SUPERDISK II $29.95 ( 5 1/4") $34.95 (8").

## AND FUN, TOO!



**BOOKKEEPING THE EASY WAY — WITH BUSINESS I**

Our business package 1 is a set of programs designed for the small businessman who does not have and does not need a full time accountant on his payroll.

This package is built around a **GENERAL LEDGER** program which records all transactions and which provides monthly, quarterly, annual, and year-to-date PROFIT AND LOSS statements. GENERAL LEDGER also provides for cash account balancing, provides a BALANCE SHEET and has modules for DEPRECIATION and LOAN ACCOUNT computation.

GENERAL LEDGER (and MODULES) $129.95.

**PAYROLL** is designed to interface with the GENERAL LEDGER. It will handle annual records on 30 employees with as many as 6 deductions per employee.

PAYROLL - $49.95.

**INVENTORY** is also designed to interface with the general ledger. This one will provide instant information on suppliers, initial cost and current value of your inventory. It also keeps track of the order points and date of last shipment.

INVENTORY - $59.95.

**GAMES FOR ALL SYSTEMS**

**GALAXIAN** - 4K - One of the fastest and finest arcade games ever written for the OSI, this one features rows of hard-hitting evasive dogfighting aliens thirsty for your blood. For those who loved (and tired of) Alien Invaders. Specify system — A bargain at $9.95

**NEW — NEW — NEW**

**LABYRINTH** - 8K - This has a display background similar to MINOS as the action takes place in a realistic maze seen from ground level. This is, however, a real time monster hunt as you track down and shoot mobile monsters on foot. Checking out and testing this one was the most fun I've had in years! — $13.95.

**NIGHT RIDER** - You've seen similar games in the arcades. You see a winding twisting road ahead as you try to make time and stay on the road. NIGHT RIDER uses machine code to generate excellent high speed graphics - by the same author as MINOS.

NIGHT RIDER — $12.95 cassette only

**THIEF** - Another machine code goody for the C1P cassette only. You must use mobile cannon to protect the valuable jewels in the middle of the screen from increasingly nasty and trigger happy thiefs. Fast action and fun for one or two players. THIEF $13.95 on C1 cassette only!

**SUPPORT ROMS FOR BASIC IN ROM MACHINES** — C1S/C2S. This ROM adds line edit functions, software selectable scroll windows, bell support, choice of OSI or standard keyboard routines, two callable screen clears, and software support for 32-64 characters per line video. Has one character command to switch model 2 C1P from 24 to 48 character line. When installed in C2 or C4 (C2S) requires installation of additional chip. C1P requires only a jumper change. — $39.95

C1E/C2E similar to above but with extended machine code monitor. — $59.95

***Please specify system on all orders***

**This is only a partial listing of what we have to offer. We now offer over 100 programs, data sheets, ROMS, and boards for OSI systems. Our $1.00 catalog lists it all and contains free program listings and programming hints to boot.**

VISA

**AARDVARK TECHNICAL SERVICES, LTD.**
**2352 S. Commerce, Walled Lake, MI 48088**
**(313) 669-3110**

OSI OSI

```
19102 REM
19110 FC=0
19120 READ FU$: IF FU$="E" THEN 19210
19130 IF FU$<>"H" AND FU$<>"I" AND FU$<>"O" THEN STOP
19140 FC=FC+1: FS$(FC,0)=FU$: READ FS(FC,0),FS(FC,1),FS$(FC,1)
19150 IF FU$="I" THEN READ FS(FC,2)
19160 GOTO 19120
19200 REM
19201 REM ** ENTRY POINT: REPEAT LAST SCREEN
19202 REM
19210 FU=3: X=0: Y=0: GOSUB 19900
19220 FU=2: FOR FI=1 TO FC: IF FS$(FI,0)="I" THEN 19260
19230 X=FS(FI,1): Y=FS(FI,0): GOSUB 19900
19240 IF FS$(FI,0)="H" THEN PRINT FS$(FI,1): GOTO 19260
19250 PRINT FV$(VAL(FS$(FI,1)))
19260 NEXT FI
19270 FI=1: CF=0
19280 IF FS$(FI,0)="I" THEN 19320
19290 FI=FI+1: IF FI>FC THEN 19460
19310 GOTO 19280
19320 X=FS(FI,1): Y=FS(FI,0): FU=2: GOSUB 19900
19330 IF CF THEN PRINT SPC(FS(FI,2)): GOSUB 19900: CF=4
19340 P1=PEEK(2794): P2=PEEK(2797): POKE 2794,0: POKE 2797,0
19350 POKE 2972,255: POKE 2976,255: POKE 2968,255: POKE 2888,0
19360 POKE 1398,FS(FI,2): INPUT FV$: POKE 1398,71
19370 POKE 2972,58: POKE 2976,44: POKE 2968,34: POKE 2888,27
19380 POKE 2794,P1: POKE 2797,P2
19390 IF FV$<>"/" THEN FV$(VAL(FS$(FI,1)))=FV$: GOTO 19290
19410 GOSUB 19900: PRINT SPC(FS(FI,2))
19420 FI=FI-1: IF FI=0 THEN 19450
19430 IF FS$(FI,0)="I" THEN CF=-1: GOTO 19320
19440 GOTO 19420
19450 CN=-1: RETURN
19460 CN=0:  RETURN
19470 REM
19480 REM ------------------------------------------------------
19500 REM
19501 REM  FIELD DEFINITION DATA
19502 REM
19503 REM  FORMAT:
19504 REM
19505 REM    HEADING:   NNNNN DATA "H",ROW,COL,"TEXT"
19506 REM    INPUT:     NNNNN DATA "I",ROW,COL,SUBSCRIPT,LENGTH
19507 REM    OUTPUT:    NNNNN DATA "O",ROW,COL,SUBSCRIPT
19508 REM    END LIST:  NNNNN DATA "E"
19509 REM
19510 DATA H,0,23,"TEST SCREEN"
19520 DATA H,2,0,"NAME:"
19530 DATA I,2,6,1,10
19540 DATA H,4,0,"ADDRESS:"
19550 DATA I,4,9,2,25
19560 DATA I,5,9,3,25
19570 DATA E
19580 DATA H,0,18,"TEST SCREEN #2"
19590 DATA H,4,10,"NAME:"
19610 DATA O,4,16,1
19620 DATA H,8,30,"ENTER SEARCH CODE:"
19630 DATA I,8,49,4,10
19640 DATA E
19898 REM
19899 REM ------------------------------------------------------
19900 REM
19901 REM  SCREEN FORMATTING ROUTINES MICRO-TERM ACT-5A
19902 REM
19903 REM  CALL WITH FU, X, AND Y SET
19904 REM
19905 REM  FU=1: GO TO X,Y AND CLEAR TO EOL
19906 REM  FU=2: GO TO X,Y ONLY
19907 REM  FU=3: CLEAR SCREEN AND GO TO X,Y
19908 REM  FU=4: GO TO X,Y AND CLEAR TO EOS
19909 REM
19910 ON FU GOTO 19920,19930,19940,19950: STOP
19920 GOSUB 19930: PRINT CHR$(30);: RETURN
19930 IF X<0 OR X>79 OR Y<0 OR Y>23 THEN STOP
19935 PRINT CHR$(20);CHR$(Y);CHR$(X);CHR$(0);: RETURN
19940 PRINT CHR$(12);CHR$(0);: GOTO 19930
19950 GOSUB 19930: PRINT CHR$(31);CHR$(0);: RETURN
63997 REM
63998 REM
63999 END
```

element of FV$ containing the output value. For example, 19610 DATA O,4,16,1 will display the string stored in FV$(1) at row 4, column 16.

The last specification marks the end of a screen "form". It consists of the character "E" as the only item. This may be followed by more screen specifications for other "forms".

Using this program is easy. Just describe your screen layout in data statements starting at 19510, ending each screen description with an "E" specification. Then, just GOSUB 19000 to display the first screen. When the routine returns, the FV$ array will contain the input values in the elements specified in the screen layout (DATA statements). To "replay" the same screen, GOSUB 19200. To go on to the next screen, GOSUB 19100.

The operator may specify a "/" in place of data for an input field. If this is done, the screen formatting manager backs up the prompt to the previous input field. If a "/" is given for the first input field (there is no previous one), the screen is considered to be "cancelled" by the operator. In this case, the GOSUB will return with variable CN set to -1 to signal a cancelled screen. CN is set to 0 if the screen was not cancelled.

The code from 100 to 190 is a short demonstration program which uses the sample DATA statements from 19500 to 19640. It displays the first screen and waits for it to be filled in. If it is cancelled, the screen is cleared and the program ends. If filled out, the second screen is displayed. If the second screen is cancelled, the first screen is repeated. If the second screen is filled out, the screen is cleared and a summary of the input data is displayed. Note that the summary is displayed "the hard way" instead of using the screen formatting manager.

Type in everything listed and give it a try. Then, build a few screens of your own. For fun, try your hand at figuring out how the code works. Next month I'll tell all, and you can see how close you were. Also, I'll suggest some things you could do to make the screen manager more useful.

## EXTENDED USR(X) revisited

by Yasuo Morishita
405 Lively Blvd.
Elk Grove Village, IL 60007

How nice it would be to have extra commands in OSI's ROM BASIC, such as CALL KY (=get an ASCII character from the keyboard), CALL SYSTEM (=jump to monitor warm start), CALL CLS (=clear screen), etc!

In PEEK (65), Vol. 2, #7, I showed you how to have such a convenience using an EXTENDED USR(X).

If you dare to modify the OSI BASIC ROM, you might be able to have such a nice feature, for example, by changing "NULL" to "CALL".

But OSI-ers who do not have an EPROM programmer, like me, have to do something else, right?

The result is: ?USR(0)KY, ?USR(0)SYSTEM. You have to type in ?USR(0) instead of CALL. The general format is:

?USR(0)XX,A,B,$xxxx,C,D,E,F

where XX =CALL address in Hex, a decimal value, a BASIC expression or a CALL name such as "SY", "KY", "HX" in the example program listing. (If in Hex, put a "$" sign in front of the Hex address such as ?USR(0)$FD00);

A,B,...,F and $xxxx are data to be passed to the CALL routine. 7 data items are the maximum in the program shown. These may be a BASIC expression, Hex or decimal number.

EXAMPLES

1. ?USR(0)SY - This transfers control from BASIC to the system monitor. ($FE0C)

2. K=USR(0)KY - This waits for a key input and returns with the ASCII value in BASIC variable "K".

3. U=USR(0)HX,$1F00 - This converts HEX value $1F00 into decimal value 7936 and sets BASIC variable "U" to 7936.

4. ?USR(0)$FE0C - This is the same as example 1.

5. U=USR(0)AD+10*I,B - This executes the subroutine whose start address is AD+10*I and stores a binary value of "B" at $E2,$E3 (L/H), which the subroutine may use for calculation.

6. ?USR(0)DSG,$1F00,$1FFF,100,2 - This is the command form

Program listing 1

```
 10 REM   === EXTENDED USR(X)  VER 3 ===
 20 REM
 30 REM   BY YASUO MORISHITA (SEP 1Ø,1981)
 40 REM
 50 M=576 :REM EXTUSR start address=$Ø24Ø
 60 N=146 :POKE11,64  :POKE12,2
 70 FOR X=M TO M+N-1:READ J:POKEX,J:NEXT: NEW
 80 REM
 90 DATA 162,Ø,134,9Ø,134,147,32,194,Ø,32,135,2,16Ø,254,2ØØ,2ØØ
100 DATA 185,186,2,48,23,197,147,2Ø8,245,2ØØ,185,186,2,197,148
110 DATA 2Ø8,238,185,193,2,133,225,136,185,193,2,133,224,165
120 DATA 224,5,225,24Ø,15,32,194,Ø,24Ø,13,32,1,172,32,135,2,224
130 DATA 17,48,241,76,12,172,1Ø8,224,Ø
140 DATA 2Ø1,36,2Ø8,3Ø,162,156,169,5,133,89,32,188,Ø,198,89,24Ø
150 DATA 1Ø,32,147,254,48,228,32,218,254,24Ø,239,164,149,165
160 DATA 15Ø,24,144,6,32,173,17Ø,32,8,180,166,9Ø,148,224,232
170 DATA 149,224,232,134,9Ø,96
180 DATA 83,89,75,89,72,88,255,12,254,199,2,2Ø5,2
190 DATA 32,Ø,253,76,37,18Ø,165,227,76,193,175
```

program listing 2 (1/2)

```
 10                    ; === EXTENDED USR(X) Ver 3 ===
 20                    ;
 30                    ;     by YASUO MORISHITA (SEP.10,1981)
 40                    ;
 50                    ; ==FORMAT==  ?USR(Ø)XX,A,$xxxx, B,C+1ØØ
 60                    ;
 70                    CALPTR=$EØ  ;CALL pointer
 80                    ;
 90                     *=$Ø24Ø    ;EXTUSR entry point=576
100                    ;
110 A2 ØØ      EXTUSR LDX #Ø
120 86 5A              STX $5A     ;Reset data counter
130 86 93              STX $93     ;Reset Variable name
140 2Ø C2 ØØ           JSR $ØØC2   ;Get current chr.
150 2Ø 87 Ø2           JSR CHKDTA  ;Get CALL destination
160 AØ FE              LDY #$FE
170 C8         JE2     INY
180 C8         JE3     INY
190 B9 BA Ø2           LDA CALNAM,Y ;Get 1-st chr of reserved name
200 3Ø 17              BMI JE1     ;Not found
210 C5 93              CMP $93     ;1-st chr of variable name
220 DØ F5              BNE JE2
230 C8                 INY
240 B9 BA Ø2           LDA CALNAM,Y ;Get 2-nd chr of reserved name
250 C5 94              CMP $94     ;2-nd chr of variable name
260 DØ EE              BNE JE3     ;Check more
270 B9 C1 Ø2           LDA CALTBL,Y ;Get JMP address from table
280 85 E1              STA CALPTR+1
290 88                 DEY
300 B9 C1 Ø2           LDA CALTBL,Y
310 85 EØ              STA CALPTR
320 A5 EØ      JE1     LDA CALPTR
330 Ø5 E1              ORA CALPTR+1 ;Detect JMP ($ØØØØ) to avoid
340 FØ ØF              BEQ SEROR1   ; system crash
350 2Ø C2 ØØ   JE4     JSR $ØØC2   ;Check if input line end
360 FØ ØD              BEQ JEOUT   ;Yes, it is
370 2Ø Ø1 AC           JSR $ACØ1   ;Check ",", else syntax error
380 2Ø 87 Ø2           JSR CHKDTA  ;Get input data from line
390 EØ 11              CPX #17     ;Max input data # = 7
400 3Ø F1              BMI JE4
410 4C ØC AC   SEROR1  JMP $ACØC   ;Syntax error
420 6C EØ ØØ   JEOUT   JMP (CALPTR) ;Execute CALL routine
430                    ;
440 C9 24      CHKDTA  CMP #$24    ;"$", Is this HEX data?
450 DØ 1E              BNE JCHD1
```

of my DATA STATEMENT GENERATOR, which generates DATA statements in the memory block ($1F00 - $1FFF) with BASIC program line numbers starting at 100 with an increment of 2.

Examples 5 and 6 require subroutines and name registration.

HOW TO REGISTER YOUR OWN NAME

If you want to have your own subroutines registered, replace "SY", "KY", "HX" with your own names or add them and adjust the CALNAM and CALTBL addresses. Only the first 2 characters of the name are used, the same as BASIC variables. You also have to add your own JMP tables in CALTBL.

In my case, I register only frequently used routines such as "CLS", "CCL", "SY","HX", "DSG", "RNMBR" etc. Others are called with ?USR(0)$xxxx.

HOW TO SET UP "EXTENDED USR(X)"

LOAD the BASIC program (listing 1) right after a BASIC cold start and RUN it. That's it!! The BASIC program sets up USR(X) pointers and EXTUSR routine. Due to line 70 "NEW", it will delete the program after it is RUN. If you do not want to delete it, you should change "NEW" to "END".

HOW TO RELOCATE

It is necessary to change the following subroutines and label addresses accordingly.

CHKDTA, CALNAM, CALTBL
(INKEY, HEXDML)

=== NOTES ===

1. This EXTUSR is valid only with SUPERBOARD, C1P, C2-4P and C4P BASIC in ROM ver. 1.0 Rev. 3.2 of Ohio Scientific. For other versions, it is necessary to find out the equivalent subroutines and locations and rewrite them.

2. This routine uses page zero locations $E0-$EF, $59 and $5A. If your system uses them, it is necessary to exchange them with unused ones to avoid strange happenings.

3. If you have a USR(X) statement in your own program, you have to change it to USR(0)$xxxx, otherwise you will have a syntax error.

4. If you have modified NULL into CALL, this EXTUSR may be the best location to be pointed to by CALL.

```
460 A2 99           LDX #$99     ;Offset from $FC to $95(SYNMON)
470                              ; If CEGMON, do LDX #$9C
480 A9 05           LDA #5
490 85 59           STA $59      ;Preset HEX digit counter
500 20 BC 00  JCHD3 JSR $00BC    ;Get next chr from line
510 C6 59           DEC $59
520 F0 0A           BEQ JCHD2    ;4 digits are converted
530 20 93 FE        JSR $FE93    ;HEX-Binary conversion
540 30 E4           BMI SEROR1   ;Illegal HEX found
550 20 DA FE        JSR $FEDA    ;Roll nibble into $95,96
560 F0 EF           BEQ JCHD3    ;=JMP
570 A4 95     JCHD2 LDY $95      ;CALL Address L.
580 A5 96           LDA $96      ;       "      H.
590 18              CLC
600 90 06           BCC SAVDTX   ;=JMP
610 20 AD AA JCHD1  JSR $AAAD    ;Evaluate expression from line
620 20 08 B4        JSR $B408    ;Get value into (Y/A=L/H)
630 A6 5A     SAVDTX LDX $5A     ;Get data counter
640 94 E0           STY CALPTR,X ;Save L. byte
650 E8              INX
660 95 E0           STA CALPTR,X ;Save H. byte
670 E8              INX
680 86 5A           STX $5A
690 60              RTS
700                 ;
710                 ;Registered CALL name & JMP table
720                 ;
730 53        CALNAM .BYTE 'SY','KY','HX',$FF
    59
    48
    58
    FF
740 0C FE     CALTBL .WORD $FE0C,INKEY,HEXDML
    C7 02
    CD 02
750                 ;
760                 ;Example routines (Can be omitted)
770                 ;
780 20 00 FD INKEY  JSR $FD00    ;Get chr from keyboard
790 4C 25 B4        JMP $B425
800                 ;
810 A5 E3    HEXDML LDA CALPTR+3 ;HEX-Decimal conversion
820 4C C1 AF        JMP $AFC1
830                 ;
```

---

5. Give this a try: suppose you have several machine language subroutines, which you want to switch around depending on the value of condition I. Just write:

```
10 FOR I=0TO 8:READ A(I):NEXT
20 some code here to define I.
30 U=USR(0)A(I)
40 ------------------
50 DATA 7000,7050,7090,7100,
   7120,7150,7180,7190,7200
```

A(I) should be the starting addresses of the subroutines. It works like "ON I GOTO ..." but with machine language subroutines. ★ ★ ★

---

# ADS

PRINTER SPOOLING FOR OSI
Output directed to device #1 is automatically spooled to disc tracks. Outspool to printer at your leisure. Outspool features multiple copies and auto-paging. 5-1/4" dual floppy systems only. Specify RAM size and 65D rev (3.2 or 3.3) $20. Peter Koski, 144-F Delaware Ave., Troy, NY 12180

* * * * *

C1P Mini Floppy owners add computer power to your machine with "BIG BAG". The biggest bag (5 1/4" disk) of software you can buy for the bucks) "BIG BAG" contains:

1. Single disk copier
2. Terminal emulator - makes a terminal out of your computer
3. High memory disk buffers
4. Data management system (not OSI)
5. Disk head load unload for all disk IO
6. Spool to disk (better than indirect memory)
7. VTOC fixer
8. Menu of files at boot time
9. File transmission program via phone lines with error checking and recovery, and a personal word processor.

BIGBAG is only $25. and WP is $15. with BIGBAG or $20. sold seperately. Detailed documentation is available for $5. deductable when software is purchased. Computer Power, 3223 Suffolk Lane, Fallston, MD 21047, PH. 301-692-6538.

A C4PMF ASSEMBLER EDITOR

By Al Peabody

We have received a screen editor and a couple of other programs for review (Screen Editor, RENUMB and PACKER, Western Colorado Software, 1319 N. 16th, Grand Junction, CO 81501). I think it is a great idea for vendors to send us review copies of software; we will be glad to review them in PEEK(65), within the limitations of time and space. Vendors should note that we have a limited range of computers available to us; if your package won't run on a C2 or C3 serial terminal machine or a C4PMF, you would be better off sending it to another user to review, then sending us a copy of the review.

The editor in question is incorporated into the assembler on the disk. You don't have to do anything to load it; it is there all the time. To invoke the editor, just type ^O ("^" means control-- hold down the "control" key while typing the O). When you do this, a block cursor appears at the bottom left of the C4P's screen. Typing a ^U moves the cursor up, ^L moves it to the left, ^R to the right. All other keys function normally. Rubout will erase the character to the LEFT of the location of the block cursor, typing another key will insert a new character in the same position, so it is easy to move the block cursor just to the right of a letter you want to change, rub it out and type in the replacement, without retyping the whole line.

A carriage return at any time will take you out of the edit mode and insert the revised line into the assembler's memory. It sure beats retyping lines to make changes!

One word of caution. Be sure you P (print) your source code listing out and edit the printed version. Don't do like I did. I assembled a short program, and the assembler, as usual, pointed out the more obvious errors. Then I invoked the editor and edited several lines produced by typing A (assemble). Sure enough, they were changed, but the stuff the assembler puts on the left, like the machine location of the line and the object code, was now included in the corrected line! When I reassembled, of course, it produced more errors.

Along with the assembler editor on the same disk were two machine language programs which resequence and delete REM's from Basic programs. Both work smoothly and rapidly, taking out all REMs in the one case (even making multiple-statement lines out of several single statement lines) and renumbering the entire program instantly. The RENUMB program allows you to specify a line number range to be renumbered, with starting line number, ending line number, new first line number and interval between new line numbers. Pretty slick.

As is noted in the very brief instructions provided, you must not try to add new lines from the keyboard between the time you LOAD a program to be renumbered and the time you renumber it. This will lock up the computer. Also, the packer program does not keep track of the lines which have GOTOs and GOSUBs addressed to them, so if you GOTO a REM line then repack the program the REM line number may be all gone, and the GOTO will create an error.

These three programs work well and quickly, and are quite simple to use. They are machine language fast and easy to understand. The instructions are sparse, but the programs are so simple to use that this creates no problems.

**Z-FORTH IN ROM** by Tom Zimmer
5 to 10 times faster than Basic. Once you use it, you'll never go back to BASIC! $ 75.00
source listing add $ 20.00

**OSI FIG-FORTH** True fig FORTH model for 0S65D with fig editor named files, string package & much more $ 45.00

**TINY PASCAL Operates** in fig-FORTH, an exceptional value when purchased with forth.
TINY PASCAL & documentation $ 45.00
FORTH & TINY PASCAL $ 65.00

**SPACE INVADERS** 100% machine code for all systems with 64 chr. video. Full color & sound on C2, 4P & 8P systems. The fastest arcade program available. $ 14.95

**PROGRAMMABLE CHARACTER GENERATOR** $ 99.95
Use OSI's graphics or make a complete set of your own! Easy to use, comes assembled & tested.
2 Mhz. boards $109.95

**PROGRAMMABLE SOUND BOARD** $ 74.95
Complete sound system featuring the AY-3-8910 sound chip.
Bare boards available. $ 29.95

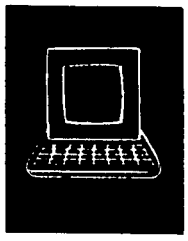
**32/64 CHARACTER VIDEO MODIFICATION** $ 39.95
Oldest and most popular video mod. True 32 chr. C1P, or 32/64 chr. C4P video display. Also adds many other options.

**ROMS!!!**
Augment Video Mod with our Roms. Full screen editing, print at,selectable scroll, disk support and many more features.
Basic 4 & Monitor $ 49.95
Basic 3 $ 18.95
All 3 for $ 65.00

**65D DISASSEMBLY MANUAL.** by Software Consultants
First class throughout. A must for any 65D user. $ 24.95

*NUMEROUS BASIC PROGRAMS, UTILITY PROGRAMS AND GAMES ALONG WITH HARDWARE PROJECTS. ALL PRICES ARE U S FUNDS.*
*Send for our $1.50 catalogue with free program (hardcopy) Memory Map and Auto Load Routine.*

CHARGEX VISA master charge



## OSI Software & Hardware

3336 Avondale Court
Windsor, Ontario, Canada N9E 1X6
(519) 969-2500

3281 Countryside Circle
Pontiac Township, Michigan 48057
(313) 373-0468

progressive computing

## THE PORT 8 DRIVER IN 65U

By Al Peabody

Recently, somebody asked how he could change the address to which his computer sent characters in response to a PRINT#8 statement. He had some sort of board which was addressed to CE00 instead of the standard CF00 of the CA-10 board. Curiosity bit. I started PEEKing around in ram with the program

```
10 FOR X=513TO24575
20 IFPEEK(X)<>0THEN50
30 IFPEEK(X+1)=207THENPRINTX
50 NEXTX
60 PRINT CHR$(7)
```

Which, of course, looks through the system, from just above the stack to just under Basic's workspace (under 65U), to see where 00CF (lobyte, hibyte form, as used in machine language instructions) would be found. I figured one of these places would be where the system was checking the status port of the CA-10 board, which is at CF00, to print characters out to it.

To make a long story a little shorter, it was. Around 19780-19800 it came up several times. "Self," I told myself, "I betcha that's why you POKE 19798 with an index to the CA-10 port number. That must be where the driver is!

A bit of disassembly led me to the routines below (I made up the mnemonics, natch). Since the address for the terminal port is FC00, not CF00, if you simply change the two CF's in this part of RAM to FC's, for instance by the Basic statement

```
POKE
19816,252:POKE19824,252
```

this will make all PRINT#8 statements come to the terminal screen.

However, I am not sure just what the code at $4D66 is doing. Obviously waiting till the status port says the device is ready for the next character, but just how is not clear. Perhaps some reader can explain.

I may even attempt more such efforts. One more question for our clever readers: what is really the best way to figure out where these little buggers live? Surely a basic program full of PEEKs and IF..THEN PRINT X statements isn't the most efficient way???

```
WHICH=$4D56   ;CA-10 PORT INDEX (19798)
CAOUT=$CF00   ;CA-10 STATUS PORT
 CHAR=$38B6   ;CHARACTER STORED HERE

   *=$4D5B

4D5B          TYA              ;SAVE Y REG
4D5C          PHA
4D5D          LDY  WHICH       ;CA-10 PORT > 0?
4D60          BPL  NEXT        ;IF SO, SKIP TO NEXT
4D62          LDY  #0          ;IF NOT, SET INDEX 0
4D64  NEXT    LDA  #02         ;READY FOR CHR?
4D66          AND  CAOUT,Y
4D69          BEQ  NEXT        ;WAIT IF NOT
4D6B          LDA  CHAR        ;GET CHARACTER
4D6E          STA  CAOUT+1,Y   ;STUFF IT
4D71          PLA              ;RESTORE Y REG
4D72          TAY              ;AND GO ON...
```

---

## BEYOND GAMES

### A Good Programming Book

A review by Al Peabody

I have been struggling with assembly language programming for some months now. As most readers know, one problem any OSI assembly language programmer has is finding decent instructions. OSI provides virtually none, and most of the books on the subject are PCIPU and Apple, too (Perfectly Clear If Previously Understood and written for the Apple computer). I have finally found one which is neither. The name of the book is BEYOND GAMES -- Systems Software for your 6502 Personal Computer, by Ken Skier [BYTE books, 70 Main Street, Peterborough NH 03458, soft cover].

Several things set this book apart from the masses:

it does not cover all of assembly-language programming the 6502, but rather concentrates on the development of useful system utilities for your computer;

within its area of application, it starts at the beginning and proceeds through an introduction to machine-language programming the 6502 to the development of such complex system software as a monitor, a table-driven disassembler and a simple text editor;

although it does not cover all possible instructions and addressing modes in complete detail, it does give excellent coverage to the most useful ones,and of course ALL instructions and addressing modes are covered in its tables and appendices;

the text, though densely written (don't plan to scan it!), is quite understandable, and is supplemented with 52 useful appendices, including listings of all software discussed. The listings occupy more than half the total length of the book, and represent a very valuable source of ready-made software;

perhaps most importantly for OSI users, the author assembled and ran all the programs in the text using his C1P (24K, minifloppy, though all programs will run easily in an 8K machine with cassette mass storage).

The book assumes that the reader has a machine with a ROM monitor and memory-mapped video board, although it would be no great challenge to adapt the programs to work with a serial terminal.

What I like best about the text is the author's professional, top-down approach to the whole process of programming. All routines are written as subroutines. In each case, the problem is defined, and the major steps in its solution coded as nested subroutine calls. Then we are led down the tree of embedded calls to the lowest level, where the real action happens.

The programming technique proposed here is fully compatible with the way most of us work, sitting at the console with only the most rudimentary flow charts scratched onto notebook paper. Skier doesn't waste time telling us all our work must be thoroughly flow-charted before we touch a keycap; he shows us when we really need flow-charts and how to make them an integral part of the programming process.

I got my copy of BEYOND GAMES through a book club and paid $11.95 for it. The list price is $14.95. Whatever price you

pay, it is certainly worth it as a reference work and text on 6502 Assembly-language programming.

★

OSI Hardware Tutorial
by Brian Hartson

The following is a start on a series of articles on OSI hardware.

The OSI 48 pin buss - pins 39,40,41 and 42 are used for system clocking and interfacing. Pin 39 Phase Two is used to time all data transfers. Pin 40 R/W (read/write) is used to control buss data direction. Pin 41 VMA, valid memory address, is used only when using the 6800 microprocessor. Pin 42 Phase Two/VMA, this line is used as the enable line for chip interfacing. The following pins are used as data and address lines. Pins 5-12 are bi-directional data lines. Pins 29-38,43-48 are the 16 address lines. Power lines - pins 25 & 26 are the +5volts. Pin 23 +12volts. Pin 24 -9volts. Pins 27 & 28 are ground. The following pins have special timing or interfacing considerations. Pin 4 Data Direction is used to control the CPU's data buss buffers. Pin 2 NMI, nonmaskable interrupt and interrupt line from an external device not used by OSI. Pin 3 IRQ, interrupt request, used to interrupt the processor to process a data transfer typically. Pin 1 Wait used on C3 systems to change the processor clock; not used on C2 systems. Pin 17 Reset: this is not system reset; this pin is not typically used. Pin 18 user defined. Pins 19-22 are called address 16-19. These extra address pins are used by level3 to do what is commonly called bank select. They select the memory partition that is currently being executed in. Pins 13-16 are the extra data lines when using the 12 bit processor (intersil 6100). Not used otherwise.

Next time, an overview of the superboard II. If you have any questions concerning OSI hardware or about interfacing to OSI hardware, please drop us a line here at PEEK (65) and we will try to answer them.

★

# LETTERS

ED:

I want to respond to your Column One in the August issue. My letter is, I suppose, an open one, addressed to the Cassette Users of our PEEK (65) community. The message is, "Count your blessings."

You compared the cars we drove five years ago with those we drive today, and suggested that the computers we have today would, and should, be replaced within five years with units which include 16 bit processors, hard disks boasting 30 megabytes on line, CP/M, MP/M, Oasis, as well as 65U and D.

Let's reshape that automobile analogy a little. Twenty years ago most of us, if we were old enough, had four cylinder, stick shifted, fabric seated Chevies. Most of us submitted to the Detroit hype across the years and "upgraded" to Parlor Cars -- V8, automatic, factory air, power windows, and vinyl seating. So what do we own today? Four cylinder, stick shifted, fabric seated Chevies. It took the Europeans and Japanese to teach us (and Detroit) that those 2 ton, 400 horse, Dream Boats were down right ludicrous for taking the kids to ballet or going to the store for a pound of butter.

A case in point. A couple of months ago I succumbed to the Silicon Valley hype, and decided to upgrade my three year old C2. My dealer worked out what I needed for dual 8 inch floppies, 48K RAM, and the rest. He then told me of a customer who was moving to a multi-user environment and had a fully upgraded C2 for sale. I investigated. The machine has everything -- dual floppies in separate blue boxes, parallel out, serial in and out, modem capability, and even four interrupt switches on the front panel. Beautiful!!! I bought it and took it home. After changing my programs to run under 65D, I began to assess gains and losses.

What I have gained is the ability to load a program in about four seconds that used to take twenty-one minutes, to print at a slightly faster rate using the parallel interface instead of the ACIA port, and to archive important data more conveniently. What I have lost is user memory, speed of execution, my personal comfort, and money.

I actually have less useful memory with 48K in my new unit, than I had with 36K in my old unit. 65D occupies 12672 bytes out of the 48K. This leaves 36480 bytes free--385 more than what I have in my C2-8P. However, 9 digit numeric arrays occupy 25% more memory than 6 digit, and disk BASIC IO conventions require much more space than ROM BASIC. As a consequence of these increased demands for memory, my largest professional program must be partitioned to disk under 65D, and, of course, I can't come close to getting it into memory under 65U.

Execution time is greatly increased, due in part to the program partitioning, and in part to the speed of 9 digit arithmetic as opposed to 6 digit.

My room, which used to be as quiet as a monastery library, now sounds like a flipping computer center. There is a fan for Drive A, another for Drive B, and a third for the computer itself. (OSI will deny it, but a C2 filled with 8 cards gets hot, and begins to throw bits after a couple of hours of operation).

Finally, I am typing this letter on my old C2 using DQ's WP6502. I paid $50. for the cassette version; the disk version costs $100. There is a lesson here.

In sum, Cassette Users, don't be intimidated by the disk and fancy systems people. If you are accomplishing what you want to with your computer, efficiently, comfortably, and inexpensively, count your blessings. If you must "upgrade", be forewarned, there is a price to pay.

Ian A. Morton
St. Paul, MN 55105

* * * * *

ED:

Someone mentioned that if you know the right PEEKs and POKEs, you can use Apple color software on Super II with grafix! Have you heard this? If so, obviously, I would love to know about it.

Gerald King
Canada

* * * * *

# SPEED POWER EFFICIENCY for OSI

## 65D3 SYSTEMS

### FBASIC: BASIC Compiler $155/$10

- **FAST** efficient machine code written with the ease of BASIC.
- **SPEED**-optimized, native-code compiler.
- **INTEGER** subset of OSI-supplied BASIC.
- **DISK** based: No problem with size of source or object files.
- **EXTENSIONS** to BASIC:
  - Simple interface to system hardware and software.
  - Direct access to 6502 registers.
  - Array initialization.
  - Optional absolute array locations.
  - WHILE and other structures.
  - Simple technique for combining compiler and interpreter advantages.
- **FULL** system: utilities (plus source), manual, and many useful examples.

### R-EDIT: Edit any program or text with ease! $40

- **FULL CURSOR** control. Edit anywhere on the screen.
- **INSERT**, replace, add, delete.
- **ONE KEY** stroke and you're editing.
- **BASIC**, assembler, etc. can all be edited without reloading editor.
- **RAM**-resident: Always ready!
- **SYSGEN** relocates R-EDIT and customizes.

### SPUL-65: Printer Spooler $95/$10 Virtual Indirect File

- **STOP WAITING** for your printer!
- **PROCESS** words, write programs...all while printing!
- **QUEUE** lets you pile on print jobs.
- **MULTIPLE COPIES** printed with top and bottom page margins.
- **SYSGEN** relocates SPUL65 and gives extensive customization.
- **INDIRECT FILE** commands produce disk files giving you:
  - A virtually unlimited temporary file.
  - A link between incompatible files; for example, use WP-2 for extensive BASIC editing.
  - Ability to merge multiple program segments.

### XREF: BASIC Cross Referencer $25

- **TABULATES:**
  - Referenced line numbers.
  - Variable names (numeric, string, array).
  - Defined functions.
- **FAST** machine language program.
- **DISK** based: Handles large BASIC source files on any drive.

### CP/M to OSI Disk Translation

Frustrated by all those good CP/M disks that won't run on your OSI CP/M system? It's that special OSI disk format! And we can fix that. Just send us your disk, $15, and you'll soon have an OSI compatible disk.

drc software

Data Resource Corporation
Suite 205
1040 Lunaai Street
Kailua, HI 96734 (808) 261-2012

Manual orders applied to software purchases. Programs supplied on 8-in, single-density, single-sided disks. Hawaii residents add 4% tax.

ED:

Is there life after death and can Shugart SA400 minifloppy drives be interfaced to an OSI C4P? The first question is quite short and near impossible to answer. The second question is much longer but quite easy to answer. The answer is yes. Ah! But you are not going to get off the hook quite so easily. There is more to mating the OSI with an SA400 than can be seen at first blush.

There are some very important differences to be considered before tying an SA400 to an OSI system. One difference is the small matter of the number of tracks that each drive can access. The standard OSI drive, the MPI-51, can access 40 tracks. The SA400 can only access 35 tracks. Up to track 35 the two drives can exchange diskettes.

Another difference is that the SA400 steps from track to track about eight times slower than the MPI-51. This is a very important difference because OS-65D takes advantage of the MPI-51's faster step rate.

Although not strictly a difference, the OSI supplied MPI-51 comes with a read data/clock separator. (the MPI-51 drives offered for sale in the back pages of many magazines do not come supplied with data/clock separators.) For your information I have supplied a circuit diagram of a useable data/clock separator. I'll have more to say about this subject later on.

The final difference between the two mini-drives is admittedly difficult to measure. The SA400 is a more reliable drive. That's right - the SA400 will probably last longer than the MPI-51 given the same operating conditions. However, both drives are very popular and well represented in the microcomputer marketplace.

How well does an SA400 work with an OSI C4P system? Very well, thank you! Each stroke of data that I enter into my OSI C4P is destined to pass through the pearly gates of one of my SA400 drives. I upgraded my system from ROM to diskette in the spring of 1980. I chose to use SA400's because of my experience with both the MPI-51 and Shugart SA400 drives. To put it gently, I have yet to see a broken SA400.

Installing the SA400 on my system was not simple - probably because I was the only one I know who had ever tried, and I had everything to learn about disk drives. But it can be done! First you must build a data/clock separator. I recommend that you do not attempt to build a separator for each and every drive that you install. Rather, build one separator and wire it between the disk drives and whatever OSI compatible floppy interface that you intend to use. There won't be any electrical conflict between drives in the same system. This is due to the fact that all drive outputs are either three-state or open-collector outputs.

The next step assumes that you are upgrading a BASIC-in-ROM system. You need to re-wire the addressing of the 65V monitor ROM to expose the disk boot routines and hide the BASIC-in-ROM I/O routines. Whatever you do, don't butcher the board! Take a line from the hippocratic oath - do no harm!!

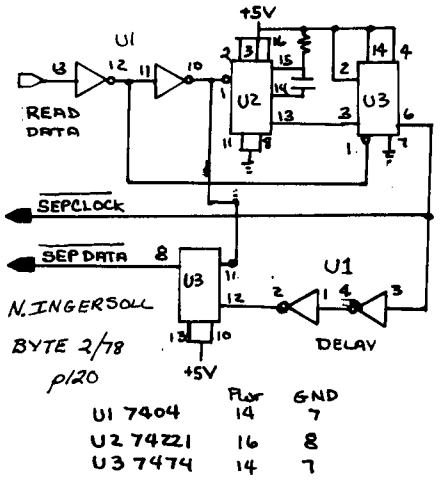
Caution - Insanity Warning!!

Catch-22:

```
10 FOR I=1 TO N+1 : REM N =
   INFINITY
20 PRINT"TO MAKE SA400'S WORK"
25 PRINT"YOU MUST CHANGE $26A3
   TO $20."
30 PRINT"TO CHANGE $26A3 FROM
   $08"
35 PRINT"THE SA400'S MUST
   WORK."
40 NEXT I
```

Everything was wired. I had checked and re-checked my work. A NOSI (non-OSI) friend checked and re-checked my work. Still it did not compute. My booting drives stumbled. Then, as these things go, I suffered an insight after bedding down for the night. The answer lay in the forgotten fact the $SA400 drives step the head at a slower rate than MPI-51 drives. The next day I took less than 15 minutes to find the software loop in OS-65D that regulates the head stepping rate. I weaseled my way into the local dealer's den whereupon I used the OSI track0 read/write program to modify track 0. Once home I discovered that there is hope in this world. My system worked great, and still does.

I have not detailed how to do some of the necessary modifications. I didn't on purpose. If you intend to step away from the beaten path, you need to think. Besides, I am in a sadistical mood. Good luck!




Nelson E. Ingersoll
Littleton, CO

Nelson:

Should your sadism be overcome (by avarice, perhaps), please recall that Peek (65) pays cash for detailed articles on such subjects.

AL

* * * * *

ED:

Let me assure you that it's no fun having a Hazeltine 1520 when you know what it can do, but you don't have the software to do it. I am constantly amazed at these companies that make this sort of equipment, when in their documentation they state something to the effect that with the proper software you can do this and that, but for all practical matters, it takes dozens of phone calls, hundreds of letters, hours of frustration just to FIND the proper software that will support this particular unit. I don't know if your Terminal Extensions Package does, but I'm crossing my fingers that it'll help.

The same adage goes for Centronics 737 (parallel) for example, where software is supposed to allow one to justify the right margin; unfortunately, all this software is written for ATARI, and ATARI BASIC and OSI BASIC aren't the same by a long shot! I am still trying to get source files of the software; to date no luck, dealers only want to sell it with the disk, and with an Atari controller for which I have no use whatever.

Then I heard about this outfit by the name of 'Microdome' (in New Jersey) which allegedly made a gadget which you'd hook up between the 470 board (Centronics parallel interface) and the printer, and with a few simple commands, it would work. Well, it didn't. It went back to the dealer. Another unit came, this time allegedly for TRS80 or an Apple; it also didn't work. The designer of the gadget wanted to make the unit useable for a broad range of machines, and after talking with him, I had the distinct impression that this included TRS80 and Apple, and not OSI. This story hasn't ended yet, but I really don't foresee an easy solution. I could almost trade up to a NEC, but that's driving the point a little too far. I want right margin justification, but I don't want it that bad!

I am not even going to elaborate what does or doesn't, did or didn't work on the 'standard' OSI equipment when I got it 8 months ago, except to say that, after concerted effort on the part of my dealer and myself, it now works as well as it ever will. Can't say that it wasn't fun, in part, but in principle when one buys an $8000 system, one would expect it to work from the onset. Well, the grass is always greener on the other side of the fence, and I suppose that other manufacturers of computer hardware and software have their problems too.

One more tiny complaint (and I hope you DO print this), when OSI sells a 550 board (multi serial port) why in the world don't they populate the board for full handshaking lines for a MODEM?? And since they don't, then why don't they make it clear in their documentation HOW to do it properly. Not everyone is an electronic specialist (I do some fixing but my mouth is bigger than my knowledge). A modem would work so much better when the DCD, CTS and other lines are properly implemented and for the price of the board, that's the LEAST they could have done!

Frederick S. Schaeffer
Jamaica, NY

Mr. Schaeffer:

You are apparently using WP-6502 V1.1 as a word processor. With no new hardware, by using another word processor, such as WP6502 V1.3, Wordstar, OSI's WP-3, etc., you too, can have justification.

It is tough to know what features to add to a board. Adding the extra lines to the 550 board, for instance, would make it tough to allow enough pins around the edges for 16 ports.

I gather you have added DCD and CTS. Why not share with us in detail how you did it?

AL

* * * * *

ED:

Thought some of your readers might be interested in the following demo program.

This is the way to change the names of arrays during execution. It can be very useful in programs using many arrays.

```
100 DEF FNV(Q)=PEEK(Q)+256*
    PEEK(Q+1) : REM Evaluate
    address vector at Q
110 DEF FNO(Q)=FNV(Q+2)-FNV(Q)
    : REM Offset between
    consec. vectors
120 :
130 REM   Declare some arrays
    for demo
140 :
150 A=FNO(124):DIM ARRAY(12) :
    REM A is relative RAM
    address of ARRAY
160 B=FNO(124):DIM B$(3) : REM
    B is relative RAM address
    of B$
170 C=FNO(124):DIM C%(7) : REM
    C is relative RAM address
    of C%
180 :
190 REM Addresses are relative
    to end of simple variable
    table
200 REM which is at FNV(124).
    Newly declared arrays are
    placed
210 REM at FNV(126).
220 :
230 REM   Put some stuff in
    'em for demo
240 :
250 FORI=1TO12:ARRAY(I)=I:NEXT
260 B$(1)="Hello":B$(2)="
    out":B$(3)=" there"
270 FORJ=1TO7:C%(J)=J*J:NEXT
280 :
290 REM   Change their names,
    to protect the...
291 :
300 S=FNV(124)
310 POKE A+S, ASC("B") : REM
    ARRAY BECOMES BRRAY
320 POKE B+S,ASC("A") : REM B$
    BECOMES A$
330 POKE C+S, ASC("X")+128 :
    REM C% BECOMES X% (Why
    +128 ??)
340 :
350 REM To distinguish array
    types BASIC stores the
    two characters of
360 REM of array names as
    follows:
370 REM Real.....the ACSII
    value of each character
380 REM Integer..decimal 128
    (hex80) added to ASCII
    value of each chr
381 REM String...decimal 128
    added to ASCII value of
    2nd chr
390 :
400 REM Now show 'em we ain't
    foolin'
410 :
420 FORI=1TO12:PRINT BRRAY(I);
    :NEXT:PRINT
430 FORI=1TO3 :PRINT A$(I);
    :NEXT:PRINT
440 FORI+1TO7 :PRINT X%(I);
    :NEXT:PRINT
450 :
460 END:  PS
```

For those who know how BASIC stores arrays this is of course no big deal. I knew nothing about it until Dick McGuire of DBMS was kind enough to show me. The demonstration program above is worth, I believe, many words of explanation. It is not however, the whole truth and nothing but... if it whets your appetite maybe you can get Dick to tell you what he told me.

Harry Suber
Salisbury, MD

* * * * *

ED:

Ever need the address of a variable? This need arises often in USR routines. For example, you may want to sort an ARRAY M% (J) of integers, as I did. The plan then is:

a) pass the address of M%(0) to the USR routine
b) manipulate M%(.) at pleasure in the USR routine
c) return to BASIC with M%(.) ready to use

We recommend one of the following approaches:

1. If you do a lot of this, you need to read OSI Technical Newsletter #21, page 14. (or)
2. Consider the following demo. Caveat! Line 100 is necessary.

Thanks to Al Heath for pointing out the Newsletter in the first place.

```
  5 REM OS-65U ADDRESS OF A
    VARIABLE DEMO BY DALE H.
    KING
 10 DIM M%(20)
 11 REM WE WILL PASS THE
```

# At last!
# Software Development TOOLS
# for Professional OS-65U Programmers:

## FIND:

If you program in OS-65U BASIC, you need FIND, a machine code overlay which resides permanently in the operating system, extending the FIND command to allow searches for variables, literals, statements, commands, functions, and constants such as line numbers.

FIND is an invaluable tool for writing code and debugging programs — especially someone else's! May be used in the immediate mode with any BASIC program in the user's workspace.

## COPY & DELETE:

These utilities save you from spending hours manually copying and moving BASIC program code. Both reside in the operating system, allowing use in the immediate mode.

COPY copies program lines character-for-character to a new line number location. Tests to make sure no existing lines will be overwritten.

DELETE removes program lines. Any linerange may be specified, although the DELETE command without a linerange is not accepted (to prevent accidental erasure of a whole program).

Using a single COPY-DELETE command with a linerange performs a MOVE of the block of lines to a new location.

COPY & DELETE are available without EDITV3 for video-based systems.

## EDITV3:

Has the usual OSI EDIT features, including Control R, F, P, and Tab, Delete, and Backspace. New features: Control D (erase from cursor to EOL), Auto Upper Case, Bell on All Illegal Characters, Auto -CR- at First-space-closest-to-EOL Flag, Masked Output Flag (prints X's instead of characters for password protection). Underscore and @ symbol are legal characters, replaced with DEL and Control X respectively. Backspace and Delete/Insert work normally. Control T now toggles Insert/Overstrike Character mode, allowing the user to overstrike characters in the middle of a line (without first deleting the old characters and then typing the new). Edit Line command deletes both first space and space between line number and statement, adding one character to editable lines.

Above flags may be set using the calling routine. The Input Editor may also be preloaded with a string to be edited, placing the cursor on the appropriate character within the line (for use in BASIC programs). EDITV3 with COPY & DELETE requires no reserved words.

## MONITR:

Find out what is going on in your Level 3 system! This simple-to-use program allows you to monitor activity while Level 3 is running. MONITR shows which users are 'up,' which are active, what program they're using, the line number (if running), and their input status. Also enables the user to debug multi-user partitions and programs. Runs from any port, and allows any user to reboot any other user. Designed for use by programmers as well as system operators.

## PRICES:

Each program package supplied on an 8-inch flexible disk.

Package 1: FIND $75.00
Package 2: COPY & DELETE (for video-based systems) $75.00
Package 3: FIND, EDITV3 incorporating COPY & DELETE (not for video-based systems) $235.00
Package 4: MONITR w/Talkie $175.00

Brochures available — write or call.


Brown/Collinson Associates
619 "E" Avenue
Lake Oswego, Oregon 97034
(503) 635-5055

```
    LOCATION OF THE ABOVE TO
    A 'USR ROUTINE'
 20 INPUT"ENTER AN INTEGER";
    M%(0) :REM TEST NUMBER
100 TEMP=M%(0) :REM SAVE IT
120 M%(0)=PEEK(148)+PEEK(149)
    *256: REM WHEN BASIC
    INTERPRETS
121 REM LINE 120 IT WILL STORE
122 REM WHAT WE SEEK IN 148
    AND 149
130 ADDR=M%(0): PRINTADDR;"IS
    ADDR" :REM GOT IT!
140 M%(0)=TEMP: REM RESTORE
    VALUE
150 PRINTM%(0) : REM
160 PRINTPEEK(ADDR)*256+PEEK
    (ADDR+1): REM JUST TO MAKE
    SURE !
200 REM ADDR=USR(ADDR) REM WE
    COULD PASS ADDR AT THIS
    POINT
```

Dale H. King
BKM Micro Systems Corp.
Suite 516,
2719 Ave. E. East
Arlington, TX 76011

* * * * *

ED:

PEEK (65) has proven its worth again. I just bought Word Star V. 3.0 and found it disappointing until you published the POKEs necessary to patching it. Unfortunately, those POKEs are not valid for the V3.0 of Word Star and CP/M V. 2.21 which is the one I have. I called Al Black and he was able to piece together the correct information and I want to share that with you. Patching to INISUB: should read:

C3 F7 02 C3 02 FD

The patches to 02F7 should read:

3E 00 32 10 BF C9 3E 01 32 17 BF C9

The first five bytes from the original line you published are not needed if 80 is already in location 0003. Hopefully it is, since in the area after 02F7 there is no longer enough room for the entire patch. But this should work for standard printer interfaces.

I am a bit disappointed by the lack of cooperation from Micropro with regard to this patch. I thought that just OSI gave poor support. Al Black was a GREAT help.

Michael Anderson
Arlington, VA

* * * * *

ED:

UPDATE ON; Indirect Jumps for the C2/C4, (RE: PEEK(65) August 1981).

Mr. Elliot Spiro of Wantash, NY, has been working on a similar indirect jump mod. His mod also works for cassette based C2 systems using SYNMON V1.0, but in addition, preserves the initialization routine for the old 430 board.

He has sent me the following patch. This patch makes my mod work with C2 systems having the 430 board. Thank you Mr. Spiro for sharing your info.

MY ORIGINAL:

```
(FF1B) 441B 200EFE JSR $FE0E
       441E BD2BFF LDA $FF2B,X
       4421 9D1802 STA $0218,X
       4424 E8     INX
       4425 E00A   CPX #$0A
       4427 D0F5   BNE $441E
(FF29) 4429 F00D   BEQ $4438
       442B B8     CLV
       442C FF     ???
       442D 67     ???
       442E FF     ???
       442F 99FF89 STA $89FF,Y
       4432 FF     ???
       4433 94FF   STY $FF,X
(FF35) 4435 FF     ???
       4436 FF     ???
       4437 FF     ???
       4438 B95FFF LDA $FF5F,Y
(FF3B) 443B F006   BEQ $4443
```

THE PATCHED VERSION:

```
(FF1B) 441B 200EFE JSR $FE0E
       441E BD2BFF LDA $FF2B,X
       4421 9D1802 STA $0218,X
       4424 E8     INX
       4425 E00A   CPX #$0A
       4427 D0F5   BNE $441E
(FF29) 4429 F00A   BEQ $4435
       442B B8     CLV
       442C FF     ???
       442D 67     ???
       442E FF     ???
       442F 99FF89 STA $89FF,Y
       4432 FF     ???
       4433 94FF   STY $FF,X
(FF35) 4435 20FEBE JSR $BEFE
       4438 B95FFF LDA $FF5F,Y
(FF3B) 443B F006   BEQ $4443
```

The subroutine at $BEFE has the same code as the routine wiped out at $FE04 to $FE09 (UART initialization on the 430 board) of the SYNMON chip.

A. Penaloza
Morton Grove, IL

* * * * *

ED:

Eureka! Thanks to the Software Consultants OS-65D Disassembly Manual, I found the bug in my 65D which prevented my CA-10 board from working for input of RS-232 data (PEEK (65), Dec. '80). The OS-65D supplied with our C3 has a $4F at location $24BA; this should be a $4D. Since this is part of a branch instruction, it causes the system to jump to a wrong location and made nonsense of the code. A simple POKE 9402,77 repairs the code and permits INPUT #8 to function properly.

What burns me is that Ohio Scientific must have known of this error long ago; we're certainly not the first customers in two years to attempt using the CA-10 for data input. But they feel no responsibility to advise customers of their blunders. Perhaps this bug was noted in one of their technical notices, but it escaped our local OSI dealer. I wasted probably two weeks of work time, first in pursuing the problem, then in devising methods to bypass it.

On another note, Tim Boege's fix to make WP-2 work as a BASIC editor (PEEK (65), Sept. '81) works perfectly. I'm grateful to PEEK (65) for providing this invaluable communication among users; I would never have found this

fix by myself.

Jack McKay
Washington, DC 20010

Jack:

Turns out, Tech Newsletter #4, May 1979, corrected the problem. This newsletter was sent to all OSI dealers at the time. One problem OSI has had is the ease with which people could become dealers. Things seem to be changing for the better now, with more stringent technical requirements for, and better support of, OSI dealers. Now if they will come up with a disassembly of 65U, we can get to work on it too!

AL

* * * * *

ED:

After reading your article on CP/M for OSI and glowing report on LIFEBOAT ASSOCIATES, I hunted up their catalog to see their software offerings. Much to my pleasure, I found described a "Visicalc-like" program which I have wanted, in addition to the CP/M.

With Master Charge ready, I picked up the phone to order both. Upon advising the Lifeboat person that I had a C3 with 430 board and a NEC 5520 serial printer, the air was let out of my excitement balloon. Neither had included a driver for the NEC.

The person said that I would have to do the programming in assembly language to get the NEC to go. Well, I haven't yet whipped BASIC and am not willing to begin on assembly. My question - does anyone have easy to follow instructions that I might beg, borrow or buy that would enable me to get the programming done?

With regard to good software backup, I purchased an accounting program package from Digital Technology, San Diego, CA. It is the only software that I have bought that I have been able to immediately run. They have a program called DEVSET that all I had to do was enter the type of equipment I was running and it took care of the habitual NEC serial problem. In addition, their constant updates and revisions have mainly been without charge. Questions have always been answered on the first phone call.

Where can a set (or a copy of a set) of OSI Technical Newsletters be purchased?

C. Alan Skoog
Chadwick, IL 61014

Alan:

Any OSI dealer should have a complete set of Tech Newsletters. CP/M will certainly support your serial NEC. CP/M V2.2 from Lifeboat will not support the 12-bit parallel NEC driven by "NECDRV" under 65U, but will support a serial NEC driven through the CA-10 (550) or CA-6 (430) boards.

AL

* * * * *

ED:

I would like to offer the following random pattern generating program for those programmers too wasted to do anything but stare at the wallpaper.

```
10 GOTO 30
20 FOR L =LL TO ULSTEPST :
   POKEL,C :NEXTL
30 C= INT (256*RND(1))
40 ST = INT (5*RND(1))+1
50 LL = 53248 + INT(3*RND(1))
60 UL = 54271
70 GOTO20
```

Who knows? Maybe one day the latest art forms will be in changing displays controlled by microcomputers.

David Roha
Brecksville, OH 44141

David:

I ran this program on my C3D and nothing happened. What system does it work for?

AL

* * * * *

ED:

In PEEK (65), August 1981, Vol. 2, page 14, regarding Mr. Frankforther's question: 527 board to C4P cassette.

When you add on a 527 board, you are required to add on an extra DC +5V power supply, which may be purchased or built by yourself. An extra 8K bytes of RAM with 2114L may need approximately 1 amp. If you have 24K of RAM installed on your 527, then you may need 3.0 - 3.5 amp of current capacity. The original C4P supports 5.5A maximum and does not have any room for the extra circuit. It sounds wiser to add an extra power supply than to use your C4P as a space heater!

# digital technology

**BUS-II LEVEL I BOOKKEEPING & ACCOUNTING SYSTEM**
The BUS-II turn-key multi-client accounting package is the leading OSI business software package. BUS-II Version 3.1 includes five principle modules:

| | Inst. Price | Ref. Price |
|---|---|---|
| GENERAL LEDGER | $1200 | $599 |
| ACCOUNTS RECEIVABLE (a) | 1000 | 599 |
| ACCOUNTS PAYABLE (a) | 1000 | 599 |
| ORDER ENTRY W/ INVENTORY (a)(b) | 1000 | 599 |
| PAYROLL | 1200 | 799 |

The Accounts Receivable, Accounts Payable, and Order Entry W/Inventory are completely interactive with the BUS-II General Ledger. Two optional specialized packages (completely interactive) are also available.

CPA EXTENSIONS (see below)
POINT-OF-SALE TERMINAL W/INVENTORY (see below)

The BUS-II CPA EXTENSIONS Package provides special features for accountants and bookkeepers. The POS-1 Point-of-Sale Terminal package enables the operator to use the computer system's video terminal as an on-line "electronic cash register".

Note: BUS-II V 3.1 operates on floppy-disk or hard disk-based systems running the OS-65U V 1.2 operating system (LEVEL I, II, or III). Multi-client use can accommodate any number of client companies on floppy disk systems or hard disk systems with H/D/E (required for hard disk use). BUS-II LEVEL I files are limited in size for floppy disk back-up; floppy disk operation continues in case of hard disk failure.

**BUS-II LEVEL II (EXPANSION TO BUS-II LEVEL I)**
BUS-II LEVEL II is designed for much larger businesses. Expanded file size and special operations allow virtually unlimited numbers of accounts and transactions. BUS-II LEVEL II requires BUS-II LEVEL I. Minimal back-up is data cassette (tape) or floppies—although multiple Winchester disk operation is recommended (provides ability to continue computerized bookkeeping functions in case of hard disk failure). H/D/E Hard Disk Executive is required.

| | Inst. Price | Ref. Price |
|---|---|---|
| GENERAL LEDGER | $ 600 | $ 399 |
| ACCOUNTS RECEIVABLE | 600 | 399 |
| ACCOUNTS PAYABLE | 600 | 399 |
| ORDER ENTRY W/ INVENTORY | 600 | 399 |

**POS-1 POINT-OF-SALE TERMINAL** (a)(b)
POS-1 is an on-line multi-store point-of-sale terminal program with integrated inventory designed for cash register emulation. POS-1 controls cash drawer and ticket printer (or system printer). Automates taxable or nontaxable sales, cash transactions, and credit sales (with verification operations). POS-1 is interactive with the BUS-II V 3.1 BOOKKEEPING & ACCOUNTING SYSTEM.

POS-1 Inst. Price $2400 Ref. Price $1199

**CPA EXTENSIONS PACKAGE** (a)
CPA EXTENSIONS is designed for public accounting firms. A number of special operations are provided: a "bankers" Balance Sheet and Profit and Loss statement with summarization and consolidation options, Statement of Changes in Financial Position, Statement of Changes in Components of Working Capital, Cash Flow Analysis, Departmentalized Sales Analysis, Comparative Income Statement, Budgetary Analysis, Asset Depreciation Schedule (compatible with TAXMAN-1040), and Loan Amortization Schedule. CPA EXTENSIONS is interactive with BUS-II V 3.1 BOOKKEEPING & ACCOUNTING SYSTEM.

CPA Extensions Inst. Price $ 3600 Ref. Price $1599

**TAXMAN-1040 PERSONAL INCOME TAX PREPARATION**
TAXMAN-1040 is designed for tax practitioners and public accountants. TAXMAN-1040 is the leading tax package for OSI microcomputers—the package has been installed on OSI, Hewlett-Packard, DEC and IBM systems. Designed and supported by CPA tax experts. This package automatically prepares FORM 1040 and 28 schedules. Individual state tax option available. Support includes annual forms and tax table revisions. Purchasers of 1980 TAXMAN will automatically receive 1981 revisions at no extra charge.

TAXMAN-1040 Inst. Price $3600 Ref. Price $2399

**TAXMAN-1120 CORPORATE INCOME TAX PREPARATION** (a)
TAXMAN-1120 (under development) is a corporate tax preparation package designed to work in conjunction with TAXMAN-1040 to provide full-service tax accounting functions. TAXMAN-1120 requires BUS-II G/L. Individual state tax option available; support includes annual forms and tax table revisions. Purchasers of 1980 TAXMAN will automatically receive 1981 revisions at no extra charge.

TAXMAN-1120 Inst. Price $3600 Ref. Price $2399

**OS-DMX DATABASE MANAGEMENT SYSTEM**
Command-oriented OS-DMS compatible database management system. OS-DMX operates under the OS-65U V 1.2 operating system (LEVEL I, II, or III). Features such as control files, extensive operating commands, and the innovative HELP feature, in addition to Digital Technology's exclusive on-line documentation, make this one of the most usable—as well as powerful—systems available for microcomputers. OS-DMX may be used instead of, or in addition to, OS-DMS Nucleus, Query, Sort; OS-DMX will replace virtually all of the specialized OS-DMS modules—and in most applications will provide greatly improved performance.

OS-DMX Inst. Price $2000 Ref. Price $1199

**ECR-1(P) ELECTRONIC CASH REGISTER POLLING** (c)
ECR-1(P) provides cash register polling and control (including cash register reprogramming) in conjunction with OSI microcomputers. Cash register polling is an alternative to on-line operation which allows the use of regular preset-total style electronic cash registers (with RS-232 communications). Versions are currently available for MKD BANTAM II and certain NCR cash register systems.

ECR-1(P) Inst. Price $1600 Ref. Price $799

**ECR-1(C) DATA CASSETTE POLLING** (c)
ECR-1(C) provides data cassette polling, allowing multi-store cash register polling. ECR-1(C) is recommended when diverse store locations make telephone line communications prohibitively expensive.

ECR-1(C) Inst. Price $1600 Ref. Price $799

**SALES-1 SALES ANALYSIS** (c)
SALES-1 is an OS-DMX-based sales analysis package for use in conjunction with OS-DMX, ECR-1(P), or ECR-1(C). Breakdown is provided by key-hit, family group, etc., indicating totals and percentages of sales. OS-DMX is required; ECR-1 is recommended; manual stand-alone operation is optional.

SALES-1 Inst. Price $1600 Ref. Price $799

**INV-1 RESTAURANT INVENTORY & MENU EXPLOSION** (c)(d)
INV-1, used in conjunction with OS-DMX, ECR-1 and SALES-1, provides a detailed breakdown of sales by family group and menu components. Provides managerial information detailing waste, pilferage, menu costs, stock levels, reorder levels, percentage-of-sales and percentage-of-cost from menu explosion. OS-DMX required; ECR-1 and SALES-1 recommended; manual stand-alone operation optional.

INV-1 Inst. Price $1600 Ref. Price $799

**H/D/E HARD DISK EXECUTIVE**
Digital Technology's implementation of H/D/E is the answer to AMCAP's HDM. Digital Technology's H/D/E provides user functions not found on HDM or similar products: ability to copy from any user "system" to another; automatic recovery in case of "back-up to floppy" or "restore from floppy" utility failures, allowing the user 3 options: (1) ignore error, (2) abort to menu, (3) try again; use of both "A" and "B" floppy drives to back-up hard disk files; and automatic back-up diskette initialization. H/D/E operates on any OSI Winchester disk system from 7 - 80 megabytes. Re-use of hard disk space is provided. Superior to AMCAP's hard disk manager in every respect (and Digital Technology software does not self-destruct). NOTE: H/D/E is required when installing any Digital Technology business applications packages on OSI hard disk systems.

H/D/E Inst. Price $800 Ref. Price $499

**H/D/M MULTI-USER MANAGER** (g)
H/D/M (under development) is Digital Technology's multi-user extensions to OS-65U. Replaces T-MUM by AMCAP. Need we say more?

H/D/M Inst. Price $1200 Ref. Price $499

**DIGITAL TECHNOLOGY, INC.** software is sold through OSI Dealers worldwide. For detailed product information call (714)270-2000. For the name of your nearest OSI Dealer call (toll free) OSI's "hot line" 1-800-321-6850.

**digital technology**
inc.
**P.O. BOX 178580**
**SAN DIEGO, CA 92117**
**(714) 270-2000**

**REQUIREMENTS**
(a) BUS-II LEVEL I G/L req'd
(b) BUS-II LEVEL I A/R req'd
(c) OS-DMX req'd
(d) ECR-1 recommended
(e) C3 CPU W/56K RAM & OS-CP/M or Lifeboat Associates CP/M req'd
(f) SYNCHRONOUS INTERFACE ASSY req'd
(g) H/D/E Required

I used a 527 board with 8K of RAM for one year and then I bought D&N's MEM CM9-FR board instead. The new board has a floppy controller with real time clock and space for 24K of RAM. Right now I'm not using a floppy, but using 16K+3K of RAM on this board.

I showed you one way to extend your BASIC in ROM via USR(X) in the July issue. There is a way to extend your BASIC without using USR(X).

You must modify the GETCHR routine ($00C2-$00D3) to detect your own CTRL key words. But this method slows down the computer's execution speed, since this routine is called at least once per text character.

If you can stand this inconvenience, it may be better to modify this routine to do what you want, because it will simplify command input. You can type in "!GET" instead of "U=USR(0)GET". Which way do you prefer? It's up to you!!

Yasuo Morishita
Elk Grove Village, IL 60007

* * * * *

ED:

Missouri Indexing has undertaken an important and large project in the Personal Computer field. We are preparing a cumulative index of all past and present articles, editorials, and columns contained in over 22 Personal Computer magazines. This publication will not be a duplication of MICROCOMPUTER INDEX(tm) by Joe Ward. Missouri Indexing's will index by multiple subjects all past articles, etc. combined with all other magazine's articles, etc. A cumulative combined index of major, 600+ pages, proportions! These indexes will be resident on our large scale computer for further updating and the preparation of special indexes.

In order to prepare this index, which will exceed 600 pages, we need copies of all past issues of Peek (65).

Please ship these magazines to us as soon as possible, as we have committed to a September 1981 printing schedule.

William H. Wallace
St. Louis, MO

William:

Glad to cooperate. Let us know when the index is ready and how PEEKers can access your computer file directly or generate special search requests.

AL

* * * * *

ED:

You did a great job of printing my X Mon. article in the Sept. PEEK(65)!! The only error I found was one of mine. When relocating the X Mon., one should type 0800, 1000 instead of 0800,0FFF.

Here are some new tricks I've learned. To kill the '$' in disassemblies and (in my opinion) make the display more readable:

change $08BD (A4) to A0 or 80
change $08C1 (A4) to A0 or 80
change $08C2 (A4) to A0 or 80

To kill the 'Q' counter when printing long disassemblies:

change $09AC (30) to 4C
$09AD (37) to A0
$09AE (D0) to 09
$09AF (F0) to EA

See Listing for a mod to make the 'N' command print ALL data strings found in the search area. The 'deluxe' version doesn't fit in the space that the old code occupied.

TOTAL SEARCH FOR 'N' COMMAND

```
 10 ;TOTAL SEARCH FOR
 20 ;X-MON. 'N' COMMAND
 30 ;
 40 ;MY X-MON @ $1800
 50 ;
 60 *=$1D71
 70       LDX  $DC
 80       LDA  $DD
 90       JSR  $1A78 PRINT
          ADDRESS
100       JSR  $1B07 LF/CR
110       BNE  $1D64 LOOP
120       BRK
130 ;
140 ;DELUXE VERSION
150 *=$1D71
160       JMP  $2000
170 *=$2000
180       JSR  $1C51  PRINT
          'SPACE'
190       LDX  $DC
200       LDA  $DD
210       JSR  $1A78
220       JSR  $1B07
230       JMP  $1D64
```

Kerry Lourash
Decatur, IL

* * * * *

ED:

I was never quite clear on the exact procedure for getting the OS-65U disassembly listing. Please publish or send the needed information.

Carl Eidbo
Fargo, ND 58108

Carl:

Nobody is clear about getting a 65U disassembly listing. Supposedly, Four State Computer has made a nearly complete one; it is rumored the factory is producing an "official" one; lots of guys (me included) have disassembled little bits of 65U. I hope we will all be able to buy a complete listing soon!

AL

* * * * *

ED:

The Basic compiler sold by Aardvark also screams.
The program:

```
10 FOR I=1 TO 60000
20 A=A+1
30 NEXT I
40 END
```

Runs in 7 seconds and requires 110 bytes. The compiler costs only $19.95!!

E. Morris
Midland, MI

* * * * *

ED:

I need help with a screen formatting problem on a C8P using OS65D. I want to input characters (from keyboard) and have them output to screen locations other than $D740.

I tried changing the output flag to memory (DISK!"IO ,10") and then specifying video screen memory locations (e.g. DISK!"MEM D1F0,D1F0").

When an INPUT statement is encountered, the "?" prompt and character inputs appear as desired. However, the character delete (SHIFT-O) function is disabled (prints out some characters) as well as the carriage return.

I have also attempted a row by row keyboard polling routine, but I can't seem to do this efficiently.

Need help fast!

Steve Stratton
Clemson, SC

* * * * *

NEW PRODUCTS
*****ANNOUNCEMENT*****

Five new products for increasing OSI computer areas of usage.

The first two add IEEE-488 General Purpose Instrumentation Bus (GPIB) controller functions. These interfaces provide programmable control for all GPIB functions of a standard 'C26' type IEEE-488 instrumentation controller, giving OSI machines a superior capability for controlling any combination (up to 15 at a time) of standard IEEE-488 instruments. Machine code GPIB drivers are linked to BASIC to provide easy control of the GPIB.

Disk GPIB controller software for OS-65D and for OS-65U is available. Software for ROM systems is a set of three 2716 EPROMS which include GPIB drivers, all features of ROM-TERM, Super Trace, Real Time Clock and parallel printer driver.

The second two new products are C1-P monitor ROMs. ROM-TERM II eliminates the need to mess with OSI tapes for 48 column display and adds a superior machine code terminal emulator program.

These ROMs provide extremely capable terminals for use on Time Share Systems including use as remote terminals in a dial-up net on OSI Multiuser Systems.

Another NEW ROM called SYNKEY is for C4P-MF, C4P-DF and C8P-DF computers and makes them either a Polled Keyboard Video System or a Serial System at the flip of a switch.

Micro Computer and Video/Data systems, LasVegas, NV, 702/871-3263

**********
*TIDBITS FROM OSI*
**********

*** DMS PLANNER PLUS ***

The DMS Planner software package has been upgraded to DMS Planner Plus. This new package has enhanced execution speeds. The price is unaffected by this change. This new software is now in production and replaces the older DMS Planner.

**********

*** COLOR CASSETTES ***

Several dealers have inquired as to which cassettes have color and sound. The following is a list of all cassettes which are available in color or with sound. These cassettes are for the C4P.

| NAME | SOUND | COLOR |
|---|---|---|
| Baseball I | | X |
| Bowling | | X |
| Concentration | | X |
| Golf | | X |
| Hearts | | X |
| Othello | | X |
| OSI Invaders | | X |
| Slot Machine | X | X |
| Spades | | X |
| Zulu 9 | X | X |

**********

* CD-7 BOOT-UP PROBLEM FIXED *

A problem with the SYNMON ROM was found in C3-D and C2-D computers that caused a long delay between reset and booting the operating system. A new ROM, called "DMON", is now available from your sales coordinator free of charge.

**********

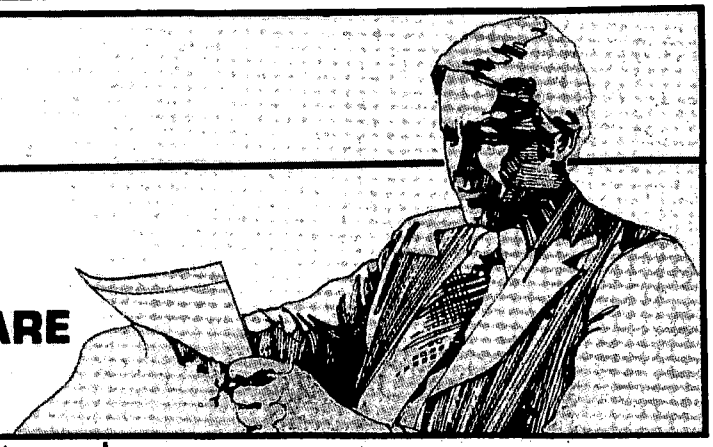
*** TECDRV REPLACES NECDRV ***

A new program is available on OS-65U disks to run the NEC, TEC and Diablo word processing printers. TECDRV incorporates all TECH NEWSLETTER fixes through TNL #28.

**********

# READ

## ABOUT OUR HI-QUALITY SOFTWARE



**That's right. The kind you get from SOFTWARE CONSULTANTS... it's efficient, cost-effective, and flexible. Next time you dump some old programs, remember to load up with the hardworking software you get from SOFTWARE CONSULTANTS.**

We're an independent vendor dedicated to producing top quality software just for you , the OSI users, at the most reasonable prices around. As we've told you before, most of our products are modifications and/or extensions of existing OSI packages. Some of our items completely replace OSI's, and all of our products are easily extensible for your custom applications.

The many responses we've recently received have shown us just how badly many of you needed a quality source for OSI software. So, to keep up with your requests for more "good OSI stuff," we've been working overtime on lots of new products. We'll be telling you about each one of them as soon as they're ready, so be sure to keep us in mind for all of your upcoming projects.

We just don't have enough room in these ads to fully describe our products to you, so please call or write us for the latest copy of the free SOFTWARE CONSULTANTS product catalog. It'll give you all the facts on our current line and full details as each new product is introduced.

Remember, for fine OSI software at the most reasonable prices around, it's SOFTWARE CONSULTANTS. Take a quick look at this list and ask yourself...can you really afford to keep using any one else's OSI software?

1. **OS-65D V3.2 DISASSEMBLY MANUAL**
   A super-complete manual that has it all -- 50 pgs. of dissassembly listings, complete and clear comments on most every line, 10 pgs. of computer generated cross reference listings, and more! Praised by many OSIers who couldn't believe it 'til they bought one. A deal at $25.95.

2. **REF COMMAND UNDER BASIC**
   A complete, cross reference utility that'll find and list any BASIC line,number, variable or numeric constant. It's available under 65D or 65U and comes on 5¼" or 8" floppies. This one will save your sanity, and cut out hours of wasted time. Yours for $31.95.

3. **SPOOLER/DESPOOLER UTILITY**
   A useful utility that feeds backed-up data to your printer for normal output, and leaves your screen free for other work by intercepting data bound for your printer and temporarily storing it on hard disk. Written in super fast machine language. Interfaces with serial and parallel printers.

4. **FIG FORTH UNDER OS-65U**
   So far, the only one that's running under 65U at a price within reach. You get all the pluses, like terminal oriented editor, lots of printer and terminal tools, and more. Under multi-user, runs BASIC simultaneously, too. Unbelievable at $89.95.

5. **VIDEO ROUTINE**
   This convenient program really spices up your video system with little niceties like 24 separate control codes, horizontal and vertical plotting, and many variable screen parameters. Software extensions are available to connect this with the graphics resolution booster. The routine alone is $25.95, with extensions, $29.95.

6. **GRAPHICS RESOLUTION BOOSTER**
   An ingenous piece of hardware that'll increase the graphic resolution per character by 8 times. With this device, your circles will be rounder and your curves smoother. A slick addition to your video-based system. Priced at $49.95, or, with the video routine and extensions, $79.95.

Unlike the majority of other software vendors, we offer our customers copies of source code (on floppies) for any of our products they've purchased. For a nominal fee of $12, covering our material, postage, and handling, we'll send you the source code you choose. Simply write it in as you fill out the order coupon. Of course, dealer inquiries are invited for all of our products.

SOFTWARE CONSULTANTS
HARDWORKING *software*

Enclose your check with coupon and mail to: SOFTWARE CONSULTANTS, 6435 Summer Ave., Memphis,TN 38134 U.S.A. Phone us at 901/377-3503 for rush orders. TN. residents add 6% sales tax. Dear Software Consultants:

Sounds great. Here's my order. I want:

_____DISASSEMBLY MANUAL(S) @ $25.95 ea.
_____CROSS REFERENCE UTILITY(S) @ $31.95 ea.
Disk size 5¼"_____or 8"_____
OS-65D_____or OS-65U_____
_____SPOOLER/DESPOOLER UTILITY(S) @ $69.95.
Std. parallel interface_____
Serial interface w/430 board or equivalent_____
Serial interface w/CA10X board_____
_____FIG FORTH(S) @ $89.95 ea.
_____VIDEO ROUTINE(S) @ $25.95 ea.
With software extensions @ $29.95 ea.
_____SOURCE CODE @ $12.00 ea.
For_____
_____GRAPHIC RESOLUTION BOOSTER @ $49.95 ea.
With Video Routine and extensions @ $79.95 ea.
_____SOFTWARE CONSULTANTS Product Catalog Free

**NAME** _____

**ADDRESS** _____

**FIRM** _____

**CITY/STATE** _____ **POSTAL CODE** _____

**PEEK (65)**
The Unofficial OSI Users Journal

P.O. Box 347
Owings Mills, Md. 21117

BULK RATE
U.S. POSTAGE
**PAID**
Owings Mills, MD
PERMIT NO. 18

**DELIVER TO:**

# GOODIES for OSI Users!

## DBMS, INC.

**P.O. Box 347**
**Owings Mills, Md. 21117**
**(301) 363-3267**

( ) **C1P Sams Photo-Facts Manual.** Complete schematics, scope waveforms and board photos. All you need to be a C1P or SII Wizard, just — $7.95 $ _____

( ) **C4P Sams Photo-Facts Manual.** Includes pinouts, photos, schematics for the 502, 505, 527, 540 and 542 boards. A bargain at — $15.95 $ _____

( ) **C2/C3 Sams Photo-Facts Manual.** The facts you need to repair the larger OSI computers. Fat with useful information, but just — $39.00 $ _____

( ) **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, **GOSUBs** & 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 FAST! Requires 65U. Manual & samples only, $5.00 Everything for — $50.00 $ _____

( ) **Sanders Machine Language Sort/Merge** for OS-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 ASIC 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 _____

( ) Cash enclosed ( ) Master Charge ( ) VISA
( ) C.O.D., add $1.50

Account No. _____ Expiration Date _____

Signature _____

Name _____

Street _____

City _____ State _____ Zip _____

| | |
|---|---|
| TOTAL | $_____ |
| MD Residents add 5% Tax | $_____ |
| C.O.D. orders add $1.50 | $_____ |
| Postage & Handling | $ 3.00 |
| TOTAL DUE | $_____ |

POSTAGE MAY VARY FOR OVERSEAS