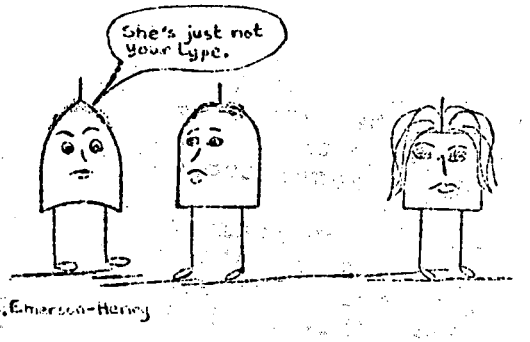
PEEK (65)

The Unofficial OSI Users Journal

★$1.50★

Editor: Al Peabody
Vol 2 No. 5, May 1981

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



Column One

A careful inspection of the stamps they are peddling at the Postal Disservice will reveal that it now costs 8 cents to mail a first class letter (actually, it costs $12 cents, not $0B cents). The USPD doesn't miss a trick. Although it was the first class increase from 15 to 18 cents which made the headlines, rest assured that the cost of bulk mail (as used by PEEK(65)) went up too. So did paper. And printing. And everything else around here except my salary. And so, as regular as the Spring which is making the fruit trees so beautiful outside my window, the price of PEEK(65) is going up. New rates are noted in the copyright notice on page 2. Now aren't you glad you just renewed for a long term?

My personal computing effort continues to be dominated by communications and electronic mail. Those of you who use the PEEK(65) CBBS know it has moved to a new location, (301) 363-4867. It has also been modified greatly. The old, rather clunky menu-driven system has been replaced by a command system. Instead of looking at lots of menus, all s-l-o-w-l-y filling your screen at 300 baud, you see the simple request "CMD:" to which you respond with such obvious commands as SCAN NOTICE, READ 3, COMPOSE, POST SALE and EXIT. Of course, if the appropriate command has fled your memory for the moment, you can type HELP and then your screen will be filled by an even larger menu, hopefully explaining your options and their implications. Comments are certainly invited.

I have found a document called the OSI Protocol for Data Communications, and am currently digesting it. Next month I promise a report, based on this document (OSI does not stand for Ohio Scientific in this case) and on the comments we are receiving concerning our call for an OSI data transfer protocol. Initial returns indicate that lots of you are very much interested in file transfer as well as messaging, and the standard we come up with should find a large and growing user community to support it.

Recent phone conversations with Tony Gold of Lifeboat Associates and Susan Padgett of Micropro International indicate that the future for OSI and CP/M is bright. There is a version of CP/M (2.23) which will run on CIII's with the 74MByte hard disk. There should soon be another which will run on the CD-7 machine. Micropro and Lifeboat are both anxious to cooperate in the effort to popularize CP/M for OSI, and the next few issues of PEEK(65) should contain reviews of WordStar and other CP/M offerings. My major objection to CP/M has long been the relatively slow disk I/O provided under OS-CP/M with floppy disks. I hope the new version, combined with Winchester Technology, will alleviate this problem and open the very broad doors of CP/M to OSI users.

It is not likely that MP/M will soon be offered, however. Mr. Gold reports that there are very difficult memory contention and hardware compatibility problems involved, and Lifeboat does not plan to write a version of MP/M for OSI at present.

Another problem is that, unlike floppy systems in which every time you boot up you are using a different disk, on a hard disk system the same disk stays in the machine all the time. This means that if you install CP/M on your CD-74, it is lost for use under any other system. You must back up all the old data on the disk, run a CP/M initializer program which takes about two hours, then you can use the disk ONLY for CP/M programs and files. So a very basic decision must be made by hard disk users. I predict that we will see increasing numbers of OSI-ers with "strictly CP/M" hard disk computers.

al

GRAPHICS UTILITY FOR OS-65D

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

I purchased my C2-4P as a BASIC-in-ROM machine and added a disk drive later. It was quite a disappointment to find that Startrek no longer ran. The ROM version of the program displayed the Enterprise, Klingons and Starbases as graphics characters by means of PRINT statements. Under OS-65D, those statements produced either blanks or alphanumeric characters.

Now, however, Startrek and all my other games that print graphics characters run fine, thanks to the GRAFIX utility, which allows almost all the graphics characters to be printed. The program is very short and simple and may be included at the beginning of a graphics-based game or stored on disk as a stand-alone utility.

One reason that graphics characters can't normally be printed is that the video display routine in OS-65D first takes each character to be displayed and masks off the most significant bit (bit 7). Since the graphics characters from 128 - 255 have a 1 in the bit 7 position, they will be printed as the corresponding ASCII characters with this bit removed. So, for example, if you enter PRINT CHR$(161), you will get an exclamation point, which is CHR$(33). 161 = $A1 = 10100001. 33 = $21 = 00100001.

You can't print the graphics characters from 0 - 31 either, because the video routine checks to see if the character to be printed is less than 32, and if it is, throws it away.

This may all sound ridiculous, after all, OSI went to the trouble of burning all those graphics characters into the character-generator ROM. There are a couple of reasons for it, however. One is that certain characters are codes that tell the computer to do something. You wouldn't want to print CHR$(13), for instance, because that is the ASCII code for a carriage return. You would get tired of seeing a tree (I think it is a tree) at the end of each line. Similarly, CHR$(10) is a line feed and CHR$(8) is a back space. There are two additional codes, CHR$(12) and CHR$(16), which the operating system recognizes for purposes of its own.

The other reason for not printing the set of graphics characters from 0 to 31 is that these are the ASCII control codes. While they have no function under 65D, they are recognized by some hardware devices such as terminals and printers and by some software such as word processors and communications programs. By not printing these characters, OSI computers maintain compatibility with other equipment and software.

Nevertheless, if your main purpose in life is to print graphics characters, the GRAFIX utility will allow you to do it. When you enable GRAFIX, it POKEs NO-OPs to the video display routine eliminating the code that masks the high bit and does the character checking. The only characters that remain unprintable are CR, LF and BS mentioned above, and the cursor symbol, CHR$(95). When you disable GRAFIX, it restores the video display routine to normal.

This utility works with both versions 3.0 and 3.2 of OS-65D. The video display routines differ in the two versions, but the masking and character checking code is the same.

The information on how the video display routine works comes from the "OS-65D V3.2 Disassembly Manual" published by Software Consultants in Memphis, TN. They usually have an ad in Peek (65). This is an excellent, well-annotated source listing and is worth the money just for the authors' irreverent comments on the inner workings of the operating system.

```
 10 REM  ********************
 20 REM  *                  *
 30 REM  *   G R A F I X    *
 40 REM  *                  *
 50 REM  ********************
 60 REM
 70 PRINT"DO YOU WANT TO
    ENABLE OR DISABLE THE
    GRAPHICS"
 80 INPUT"OPTION (E/D)";A$
 90 IF LEFT$(A$,1)="D" THEN
    200
100 REM
110 REM  THIS PROGRAM POKES
    NO-OPS TO THE VIDOUT
    ROUTINE
120 REM  TO ALLOW USE OF THE
    ROM GRAPHICS CHARACTERS.
130 REM
140 POKE 9633,234 : POKE
    9634,234
150 FOR I=0 TO 15 : POKE
    9649+I,234 : NEXT : END
160 REM CNTRL P AND CNTRL L
    ARE DISABLED.
170 REM
180 REM  RESTORE VIDOUT
    ROUTINE.
190 REM
200 POKE 9633,41:POKE
    9634,127:POKE
    9649,201:POKE
    9650,16
210 POKE 9651,240:POKE 9652,71
    :POKE 9653,201:POKE 9654,1
    2
220 POKE 9655,240:POKE 9656,67
    :POKE 9657,201:POKE 9658,
    32
230 POKE 9659,48:POKE 9660,26
    :POKE 9661,201:POKE 9662,
    123
240 POKE 9663,16:POKE 9664,22
250 END
```

★

THE BEGINNING ASSEMBLER (I)

Several letters to PEEK(65) have requested information on machine or assembly language programming suitable for beginners. Being a total beginner on the subject myself, I decided to give ML programming a try and share my experiences with all PEEKers.

In what I hope will be a series of brief columns, I will let you know what I learn. After a few short months, we should all be experts, right? Ri-i-ight.

The first step for any ML programmer is to get one of the several "excellent books on the subject." You need look no further than the box your computer came in to find one: "How to Program Microcomputers," by William Barden. This book is not badly written, and gives you an excellent idea of what goes on inside a typical microcomputer as it executes its programs. But it does have a couple of problems.

PEEK (65) is published monthly by DBMS, Inc., Owings Mills, MD 21117. Editor: Al Peabody.

Effective July 1, 1981
Subscription Rates

US (surface)	$15
Canada & Mexico (1st class)	$23
So. & Cen. Amer. (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

Software Packages

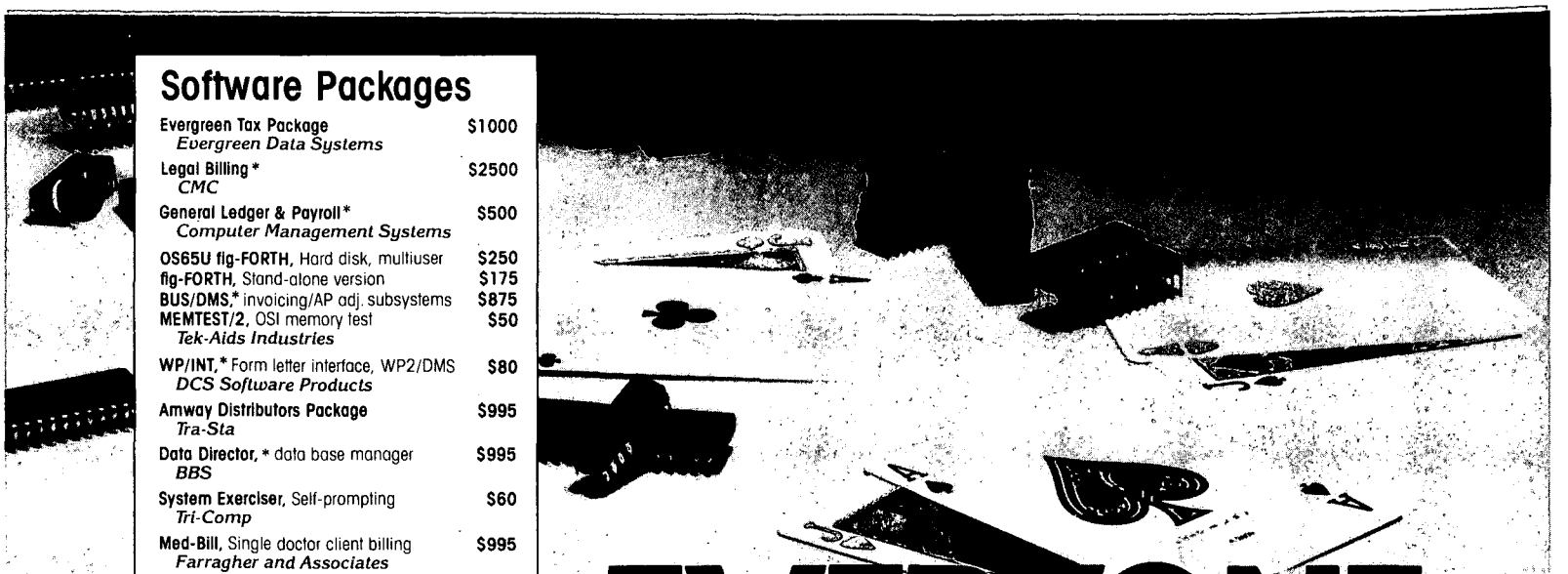
Evergreen Tax Package *Evergreen Data Systems*	$1000
Legal Billing * *CMC*	$2500
General Ledger & Payroll * *Computer Management Systems*	$500
OS65U fig-FORTH, Hard disk, multiuser	$250
fig-FORTH, Stand-alone version	$175
BUS/DMS,* invoicing/AP adj. subsystems	$875
MEMTEST/2, OSI memory test *Tek-Aids Industries*	$50
WP/INT,* Form letter interface, WP2/DMS *DCS Software Products*	$80
Amway Distributors Package *Tra-Sta*	$995
Data Director, * data base manager *BBS*	$995
System Exerciser, Self-prompting *Tri-Comp*	$60
Med-Bill, Single doctor client billing *Farragher and Associates*	$995
Manufacturer Control Systems *	$3500
Accounts Receivable	$1000
Accounts Payable	$800
General Ledger	$500
Payroll	$750
Inventory *Frisch Computer Systems*	$750
DQ Mail,* Version 1.2, Form letters	$75
DQ Justify, Version 1.2, copy and spacing	$50
WP6502, Version 1.2 **$100.** Version 1.3 *DQFLS*	$150
USUS Software Exchange Library *UCSD System Users Society*	$80

ask us about **BUS-II**

*DMS Compatible

Affiliated Dealers

Business Computer of Joliet	815/797-6500
Business Data Systems, Inc.	303/444-3652
Computer Management Systems	605/996-1300
Computer Systems Co.	713/972-1983
Data Buss	312/680-1133
Data Services Computer Corp.	303/371-4300
Computers Plus, Inc.	414/321-1770
Frisch Computer	612/770-5600
Custom Computer Application	309/342-0524
MAP Systems	309/685-1555
Practical Computer	217/523-9899
Specialized Computer Systems	517/764-6800
Total Data Systems	303/493-0915
Tra-Sta Computer Shoppe	303/542-8259
Caise Computer	815/939-4208
Delta Data Distributors	901/362-7065
International Automation	412/337-6080
Creative Office Systems	317/243-0566
Small Business Systems Inc.	308/436-2709
Kansas Computer	316/624-1888
Business Information Processing	217/224-3337
Custom Systems Development	316/262-1415
Cybercom Systems Inc.	215/432-5143
American Business Computers	203/445-5166
Crescent Computer Systems	212/926-7634
Opus Systems, Inc.	213/821-0722
Systematic Computer Services	516/791-5668
Mississippi Memory	601/875-2723
West Central Computer	612/839-2150
Advanced Computer Systems	315/265-5623
Computer Connection	313/477-4470
Newtronics	202/234-6646
Carolina Computer Bus. Systems	803/797-6500
Frank Thornberg Inc.	312/454-1471
Elgin Business Machines	312/741-4595
On-Line Technologies, Inc.	404/322-7334
The Computer Works	815/459-3476
Mikron Data Systems, Inc.	618/457-3576
Basic Business	804/874-0037
Creative Management Computers	414/228-9737
RTP and Associates	312/359-0923
Miami International Data, Inc.	305/443-3108
Cybertronics	713/680-2664
Tri-Comp Inc.	303/426-7743
Tek-Aids Industries, Inc.	312/870-7400

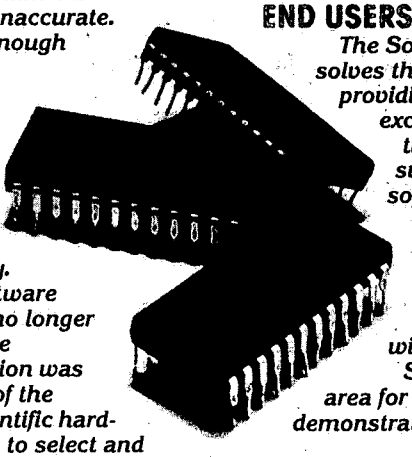


EVERYONE WINS

The Ohio Scientific Software Game

Selecting software for your Ohio Scientific computer is a chancy task at best. There are few trustworthy vendors with a national reputation. There are no consistent quality standards and the documentation is often cryptic and inaccurate. If you are lucky enough to find a good package, there's no guarantee of ongoing support. A wrong choice results in months of wasted time, effort, and money.

With the Software Federation, you no longer take that risk. The Software Federation was formed by three of the largest Ohio Scientific hardware distributors to select and market quality software through reputable dealers nationwide.



DEALERS

The Software Federation solves the dealer's problems by providing low cost access to high quality software with the sort of demonstration packages, documentation, and support that the dealer needs to successfully sell machines.

AUTHORS

The Software Federation solves the independent vendor's problems by providing a proprietary method of software protection, aggressive enforcement of software licenses, a strong dealer base, primary support, and national advertising.

END USERS

The Software Federation solves the user's problems by providing quality software, exceptional documentation, after-the-sale support, and optional software maintenance services.

Why risk making the wrong choice? With the Software Federation, everyone wins!

See the dealer in your area for a complete turnkey demonstration.

Software Federation Inc.
44 University Drive
Arlington Hts., IL 60004
Phone: 312/259-1355

TM

First of all, it is written for all three of the chips carried on the 510 (C-3 CPU) board, and so cannot go into too much detail on any one (and the information for the others is confusing to me); and secondly, while it tells you what the instructions for the various chips are, and what they do, and maybe even gives you all you need to know to write a simple program down on paper, after you get that program written, you might want to try it out.

Here, the book is no help at all. I wrote a little program to store some numbers in some memory locations, then turned to my C-3. The console stared at me invitingly, daring me to type anything in. I even knew what I wanted to type in, but just typing it onto the screen would do no good! I didn't know how to make it "go."

More research was needed. I tried the other manuals and books which came with the computer, and sure enough, in the "Challenger III Setup and Operations Manual," page C-18 (THAT took me a while to get around to!) is a sort of an explanation of what to do, complete with a very short program which causes whatever you type in to print twice on the screen.

By reading this explanation several times, and trying the program, I learned that you have to enter the monitor by responding "M" to the question "H/D/M" or "C/W/M" when you first turn your computer on, then you can change memory locations (an awkward way to type in programs) and run the programs you have typed in by simply typing "L012D280200R G" ... AFTER you "hand assemble" the programs, figuring out the proper operation code for each instruction from the chart in your programming text, calculating the length of each instruction, and writing it all down in two passes, one to figure out actual locations of routines, the second to code in their addresses when they are called as subroutines or in branch instructions.

I needed more help, and went to my neighborhood computer store for more literature.

The second book I tried was "Programming the 6502," by Rodnay Zaks. I don't want to put down Mr. Zaks; he has been around quite a while, and I gather he has written lots of neat books. In my opinion, this isn't one of them. It reads as if it had been written by someone who speaks very little English, and didn't help me to get any further along at all.

Back to the computer store. This time I bought "Programming and Interfacing the 6502, with Experiments," by Marvin L. De Jong, after looking it over carefully. It is a dandy book, but does have a couple of problems. It is designed solely for the 6502, but is intended to be run on a KIM, AIM or SYM computer, so some of the experiments don't fit. But it does start out very simple, and in combination with the C-III manual (sort of) explaining the monitor, I have been able to run several of the experiments. I wrote the following program:

```
LOOP    LDA  #$0   PUT A ZERO IN A
        STA  0300  STICK IT INTO LOCATION $300
        INC  0203  MODIFY THE PROGRAM -- A NONO
        JMP  LOOP  AND STICK IT INTO THE NEXT LOCATION
```

I will not critique the program (I judge only other authors!) but will comment only that the program is designed to set a bunch of locations to zero, and that it moves to the next location each time through the loop by changing itself, incrementing the location which contains the low-order byte of the location in the STA instruction. I did it this way because I didn't yet know how to store a 2-byte address and increment it. I have since learned that you can use STA 10,Y and the computer will take the address stored at locations 10 and 11, increment it by whatever is in the Y register, and store the contents of the Accumulator there. In combination with INY (increment Y) in a loop, this works fine. Then you can increment location 11 (the Hibyte of your address) and test it with INC and CMP instructions to see if you have got as far as you want to in your test. It is all quite time-consuming, especially if you are doing it by hand assembly, but it does force you to become familiar with the entire instruction set in all its variety and is an excellent way to learn.

By next month, if you are to follow along in this exercise of the blind leading the blind, assemble something which at least has a list of the instructions the 6502 will recognize and something else which lets you know how to load and run a program with your model of computer, and meet me in the next issue.

Al Peabody

★

GOODIES for OSI Users!

from

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 — $34.95 $ _____

( ) **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 BASIC program on any disk, floppy or hard. Most versatile disk sort yet. Will run under LEVEL I, II, or III. It should cost more but Sanders says, "...sell it for just..." — $89.00 $ _____

( ) **KYUTIL** - The ultimate OS-DMS keyfile utility package. This implementation of Sander's SORT/MERGE creates, loads and sorts multiple-field, conditionally loaded keyfiles. KYUTIL will load and sort a keyfile of over 15000 ZIP codes in under three hours. Never sort another Master File. — $100.00 _____

TOTAL	$_____
MD Residents add 5% Tax	$_____
C.O.D. orders add $1.40	$_____
Postage & Handling	$ 3.00
TOTAL DUE	$_____

( ) Cash enclosed ( ) Master Charge ( ) VISA
( ) C.O.D., add $1.40

Account No. _____ Expiration Date _____

Signature _____

Name _____

Street _____

City _____ State _____ Zip _____

THE WRONGS OF COPYRIGHTS

By Jeff Beamsley
The Software Federation

During the past several months I have had the opportunity to discuss the morality of the existing copyright laws with people who hold widely divergent views on the matter. The existing laws concerning computer software are vague and weak at best. It was only during the waning months of the Carter administration that the copyright laws were formally extended to cover computer software. They remain woefully inadequate to protect the rights of the software author.

Those laws at their foundation were designed to protect the printed word and rest very heavily on the fact that it is difficult to reproduce a bound volume. That is not the case with a program published on floppy diskette.

Because of the ambiguity and flexibility of the English language, the copyright laws also work with the assumption that it is fairly unlikely that two authors working independently will both express the same idea using the same words. Remember that the copyright laws do not protect the idea, only the method used to express it. Computer languages, though English-like, are much more restrictive in the range of possible expressions of the same idea. So it is entirely likely that two independent software authors could create programs with not only identical function, but nearly identical text.

These two failings in the present copyright laws leave a rather large loophole that many people with a healthy exercise of situational ethics are climbing through. The argument is a fairly easy one to make. You purchase a piece of software that does not perform up to your expectation. Maybe it doesn't function at all. In any case you decide to modify the program to fix/improve it. The argument then goes that at some point in the modification process, if it is sufficiently extensive, the program ceases to belong to the original author, and the modifier or improver can now claim it as his own. WRONG!

Contrary to the commonly held view, the current law does protect the rights of the original author. No modifier, regardless of the extent of the modifications, can claim rights to a previously copyrighted program. For the solipsists, though, that is a problem easily solved. As long as the new author ignores and denies the original author's claim of copyright protection, it becomes the responsibility of the original author to prove that the new author is infringing on his exclusive copyright. Given the current level of understanding in the courts of the nuances of computer programming, this is at best a risky undertaking.

That brings us to the current sorry state of affairs. The Software marketplace has become a dangerous place for independent software authors. They have had to resort to methods of protection above and beyond that afforded by law. These methods typically restrict the user's ability to copy or modify the program. Though certainly necessary to protect the rights of the author, these methods many times end up limiting the utility of the program as well.

Many well-intentioned users and dealers may rebel at the idea of limited utility as a result of protection schemes. It remains your responsibility, however, to respect the rights of the original author regardless of the form chosen to distribute the software. We must insure that there is an incentive for authors to invest time and money in the production of quality software. That incentive is threatened by unauthorized modification and duplication of software. If we have any hope for the existing marketplace for independent software to survive and grow, we must protect the rights of the original author.

Those rights extend to the poorly written and supported programs as well as those with merit. Poorly written and/or supported programs should be identified so that the software consumer can make an intelligent choice. the quality of a program, though, is no excuse to violate an author's rights. The sort of subjective judgement that would say otherwise is just as morally corrupt as the previous sloppy thinker who claimed a rewritten program as his own.

I don't mean to ignore the software author's contribution to the existing problem. The software author also has some practical responsiblities in order to protect the itegrity of his software. The specifics are the stuff of another article, but suffice it to say that as the software author restricts the ability of the user to back up and modify the program, his practical responsibility for quality code and support increases.

Software is the key to our growing business. The independent software author is much in demand. You can bet that the best software will be written for the machines and marketplace that bring the greatest return for the author. To some extent that is a function of the total number of units in the field. However, that return is also based on the integrity of the dealers and users in that marketplace. We must take advantage of the unique capabilities of Ohio Scientific hardware in order to remain competitive. That is possible only if we encourage software authors to write quality programs that exploit those capabilities.

Indirect Files for Program Transfer

By Al Peabody

A recent letter from a reader indicates that some people have been having problems using the Indirect File feature of 65U and WP-2. A half hour spent testing the system convinces me that it is no wonder people are having problems. I find the instructions in the manuals (..after I finally find the instructions in the manuals) to be somewhat confusing.

SO, for anyone who may be having troubles with indirect files, here's how it works on my system.

What Indirect Files Do

The indirect file feature allows you to transfer files between incompatible systems (say OS-65U and OS-65D). This is particularly useful to allow you to use the features of WP-2 to edit BASIC programs written under 65U, and that is the example I shall use to illustrate the process. Once a 65U BASIC program has been transferred to WP-2 under 65D, the Find and Change commands can be used to change devices, alter program flow, insert REMS, etc., etc.

How they Work

The Indirect File feature works quite simply by storing a second copy of your program (in ASCII form, not tokenized) in high RAM, then reading it byte by byte into the workspace of the other system. This same mechanism is used to transfer files in both directions. After you change systems by booting up with another disk, the copy of the program in high RAM is still there (stuff in RAM doesn't go away when you reboot, your computer is just smart enough not to use it).

So, to use the Indirect Files feature, we have to do a little skull work first. We must select a location high in RAM where we want to save our program while all the rebooting is going on. It must be high enough to be above the first copy of the program, but low enough that the end of the program will not run out of RAMspace. I used $8800 for the start of the indirect file space. Next we must figure out the "page number" of the high RAM address we have decided to use. As you know, RAM is divided into 256-byte "pages." A 48K system has 192 such pages. $8800 is at page 136. Now we have the numbers we need, let's look at...

The Procedure

1. Boot up under 65U and LOAD the program you want to edit.

2. POKE 14721, 24:POKE 14646, 91:POKE 11667, 136 <CR> These POKEs enable the indirect file feature and tell the system to start sticking the second copy of the program into RAM at page 136 ($8800).

3. LIST[<CR> The "[" will not appear on the screen, but must be typed. The program will appear on the screen as the system stuffs it into RAM starting at $8800.

4. When the listing is completed, type "]"<CR>. The "]" will appear on the screen TWICE. The <CR> will cause a ?SN ERROR message, but it still works.

5. Now you have two copies of your 65U program. One tokenized copy in the normal workspace at $6000, a second ASCII copy in high RAM at $8800. Remove the 65U disk from your drive, insert the WP-2 disk and boot up.

6. We have to tell WP-2 where the high-RAM copy of the file is. To do this, when you get the dot, type WH8800 <CR>.

7. To bring the high-RAM copy of the file down, just type control-X (hold the "control" button down and type an X). Now the system will go to the location you set in (6) and start reading the stuff there just as if you were typing it in from the keyboard. When I tried it, it looked a little funny, in that the OK at the end of the program caused a CMD ERROR message, then some lines repeated themselves, the "bell" sounded, then it finally saw the "]]" and stopped. It was OK though. I could PRINT the file, and it knew where the end was.

8. Now save your file to disk (!PU SCRTCH) and edit it.

After you have finished editing the file, you will have to take it back into 65U if it is to be of any use to you. Do be sure you have saved it on your WP-2 disk, in case anything goes wrong! Then...

9. If you have rebooted WP-2, type WH8800<CR> again.

10. Stuff it into high RAM by typing P control-\ <CR> The control-\ will appear as "\" and the file will list out on the screen.

11. When the listing is finished, type control-]<CR>. A "]" will appear on the screen.

12. Now we have the edited file in high RAM and we are ready to return to 65U. Remove the WP-2 disk and boot up the 65U disk.

13. Since we have just rebooted, the indirect file command is not enabled and the computer doesn't know where the high-RAM copy is. Make the same POKEs you did in (2) above to tell it.

14. Type a control-X. On my terminal I had to really punch the X to make it work, but no <CR> is needed. The file will list out on the screen as the system reads it down into the workspace from high RAM.

15. SAVE the file to disk (SAVE"SCRTCH). You now have your edited program back on your 65U disk, ready to run.

I used the filename SCRTCH in this example, not just to have something to type, but because I believe it is important to have a file with this name, equal in size to the largest program which will fit into RAM, on every disk. This is the file into which I stick programs which have grown too big to fit into their files, or new programs I decide to save after I have entered them. The rule is, never put anything into SCRTCH which you want to keep for more than a little while; run CREATE and make a permanent home for anything you will need tomorrow. Then you never have to worry about sticking something away there should the need arise.

I have just tried the procedure I outlined here, and on my 48K system with Hazeltine 1420 terminal it works. If your experience is different, let us know!

★

MYSTERIOUS DISK ERROR

By Al Peabody

I recently ran a modified version of the CBBS program, and got a DISK ERROR 0 message. Never having heard of a disk error 0, I put on

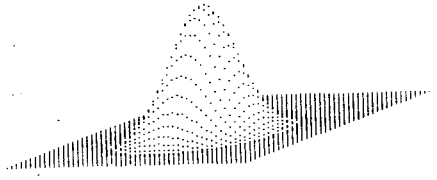
the statement trace and reran the program. Sure enough, the code which appeared looked something like:

```
* 41010 IF QA$="N"THEN250
* 41020 IF QA$<>"Y"THEN 41000
* 50000 REM DISK ERROR
```

I still couldn't figure out how an IF-THEN statement could send me to the line 50000 error trapping routine. I won't keep you in suspense any longer. Turns out, in modifying the routine to INPUT QA$ and test it for various possible errors, I had covered up the last statement, RETURN. The next line in the program was line 50000. I have now added the line 49999 STOP to my standard "starter" program which forms the basis for all new programs I write. God takes care of fools and children, it is said, but I guess programmers have to take care of themselves.

3-D HIDDEN LINE PLOTTING

HIDDEN LINE is a new software package for the plotting of three dimensional functions on any standard CRT. FULL control is provided for positioning of data on the base "Net", Z value amplification, selectable resolution, and output device width assignment.



The program is available in Microsoft Basic as a listing and will run on all popular systems in 8k. HIDDEN LINE is $25 postage paid from: Cognitive Electronics Laboratory, 9797 Meadowglen - 1608, Houston, Texas 77042.

ANNOUNCEMENTS

Program exchange timesharing service available free, 202-282-0950, LOGIN BBC. Micro BASIC, any questions, contact Allan Truelove at: 703-750-0500.

STAND-ALONE fig-FORTH FOR MINIFLOPPY MACHINES ANNOUNCED

FORTH Tools has announced a stand-alone fig-FORTH for all OSI mini-floppy computers. This system combines the well known FORTH Interest Group FORTH with powerful "stand-alone" machine drivers by FORTH Tools. OS-65D is superfluous - FORTH boots up directly, yet the disk is OS-65D compatible.

Since FORTH does not need the support of a separate operating system, FORTH Tools has developed disk, display and keyboard drivers for the OSI hardware. These are completely new and said to be more powerful than those provided by OSI.

A structured 6502 macro-assembler and disk utilities designed by FORTH Tools are included, as well as the FIG portable line editor. FORTH, the assembler, and the editor can all be in memory at once.

Stand-alone fig-FORTH for OSI is available on one 5 1/4" disk for C1 (Superboard), C2 and C4 machines with 24K. The complete price is $49.95.

Contact FORTH Tools, P.O. Box 12054, Seattle, WA 98102, for more details.

AD$

SUPERBOARD C1-P Software, hardware, and accessories for your OSI personal computer. Handsome wood cabinet for the superboard-- complete kit $20. C-10 data cassettes with plastic case 0.75 each. Write for free catalog to: Dee Products, 150 Birchwood, Lake Marion, IL 60110

BIG BAG $25 (BIGGEST BAG (5 1/4" DISK) of software you can buy for the bucks) 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 I/O. 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. New PERSWP personal word processor for OS65D C1P/4P. PERSWP is $15 with BIGBAG, $20 sold separately. For OS65D C1P or C4P. Detailed documentation is available for $5, refundable when software is purchased.
Computer Power
3223 Suffolk Lane
Fallston, MD 21047
Phone 301-692-6538

MEMORY BOARD for sale (OSI-527). Completely socketed with all support chips and 4K of 2114L-2 (expandable to 24K). $145 or will trade. Gary Kaufman, 19 Freeman Street, Edison, NJ 08820.

LETTERS

ED:

In response to Andrew Weiss' question on BASIC in ROM to disk conversion for the C2-4P, I have some information he and others might find useful.

I What You Need:
- A) 470B Disk Controller Board
- B) Enough memory (527 Board)
- C) An 8" or 5 1/4" floppy with FM Data Separator
- D) Power Supplies
- E) Connecting Harness
- F) A copy of the Disk Operating System OS65D

II What You Need To Do:

A) First decide on an 8" or 5 1/4" floppy. (Power supply requirements and Disk Controller Board configurations differ.) (5 1/4" regular +5V,+12V - 8" regular +5V,+24V. This varies depending on the disk).

B) Change the ROM Monitor strapping on the 502 CPU Board to get H-D-M on the screen after Break-Reset. H- Hard Disk, D- Floppy Disk, M- Monitor.

I have just finished adding an 8" floppy to my C2-4P. My choice of an 8" drive was based on the availability of Siemans FD-100-8Ds, The increased storage (±250K) and the relative cost factor.

The Siemans FD-100 8Ds can be purchased for about $375.00 with manuals from Floppy Disk Services. Also the 470 B Disk Controller Board is normally configured for 8" drives and needs modification to be used with 5 1/4" drives.

The FM Data Separator Components need to be installed on the FD-100 8D disk P.C. Board. The strapping on the disk has to be checked for the head load sequence. This strapping should be Step to Track 00 then load the head. The clock signals and data signals have to be checked and set as needed on both the FM Data Separator circuits and the controller board.

CAUTION! "You are on your own!" The information above was hard to come by. Help in the form of assistance, advice and vital information is almost nill.

After I get familiar with my new system, I'll write a detailed step by step procedure on how to add the 8" floppy.

Ron Biedenbach
Buffalo, NY

Ron:

Please do write a detailed report. I wouldn't want to try this complex mod without it!

AL

* * * * *

ED:

Your CBBS was a nice idea. But the implementation evokes no compliments whatsoever. I had several choice words for the system as I was using it, which I'll keep to myself.

My guess is that you have tried the impossible; to write the system in Microsoft BASIC. That language is just not fast enough!

As for your CBBS, it is too simple-minded. With too many operator errors not checked for. Your idea of simply listing a BASIC program to the system will not work! Again because of speed limitations.

I would very much like to use the system. But in it's current form I can't afford to!

Richard Foulk
Honolulu, HI 96816

Richard:

The CBBS is indeed written in BASIC. However:-
1) Please tell me which operator errors need to be checked for. Since the CBBS is in BASIC, adding additional checks is easy.
2) I have listed a 10K program (the CBBS source itself) onto the board with no problem, what problem did you have?
3) 260 users since February 1st haven't had many problems (including at least one from Alaska). I wonder if you might have had a transmission line problem. Do please document exactly what problems you are having so we can fix them!
4)The problem isn't the speed of BASIC - we input and output stuff at 1200 and 9600 baud all the time so 300 baud is not too fast. Let's find out what bugs are still hiding in my program!

AL

* * * * *

ED:

Nice to get to know you over the computer! Here is my letter about using MICRONET with a Superboard II. Actually, most of this is machine code. Dumping and loading BASIC to Micronet is done, however, using BASIC. Let me hear from you on what you think about these methods.

FILING OSI BASIC PROGRAMS ON MICRONET
For those C1 and Superboard II users who subscribe to MICRONET, here are some comments you may find useful.

1. My system is a Superboard II, 8K running at 2 MHZ (a simple mod I recommend to all C1 users). I have made the RS-232 mod, and can switch input from the ACIA to be from tape, or from a CAT type modem (set to originate, full duplex).

2. My "TERMINAL" program is all machine code and resides in 1E2D to 1FFF. To start it up, I enter the monitor (BREAK M) and type 1 F 0 0 G. A booter for "TERMINAL" is attached. You may key it in and make your own autoload tape if you like from the resulting machine code.

3. With this program running, my system behaves like a full duplex, 300 baud, ASCII terminal. I can do all normal log-ins, run MICRONET commands, etc.

4. To get on MICRONET (assuming you've signed up, paid the fees, etc.) a) Turn on your system, set the ACIA switch to input from tape. b) Load the "TERMINAL" program, then switch the ACIA switch to input from modem. (Alternately, key in and run the BASIC booter, then switch ACIA.) c) On your keyboard, type in BREAK M 1 F 0 0 G . d) Dial your local MICRONET telephone number. e) Place the handset in the modem cuffs. f) Respond as necessary to the prompts which appear from COMPUSERVE to log in, etc. g) Enter MICRONET (option 9 in COMPUSERVE's menu).

5. To save an OSI BASIC program on MICRONET a) Start system with ACIA switch in tape position. b) Load the desired BASIC program from tape, or key it in. c) Go to the monitor (BREAK M). d) Type in L to load the "TERMINAL" program from tape. e) Go to the monitor (BREAK M). f) Type in 1 F 0 0 G to start the "TERMINAL". g)

Switch the ACIA switch to input from modem. h) Log on to MICRONET. i) Type in NEW XXXX.DAT (CR) XXXX = any characters. This sets up a MICRONET file to receive line numbered input. Be sure to put a space between NEW and the file name. j) Warm start BASIC (BREAK W). k) Type in SAVE:LIST (CR). This will cause the program to list out on the screen, as if you were saving it on tape. At the same time, the SAVE caused the listing to be output to the modem, where MICRONET is sitting and waiting for line numbered input. All of the other details of communication take care of themselves, except that during the listing of the program, the C1 system ignores the full duplex responses of MICRONET and behaves for the time being like a half duplex terminal. l) When the listing is done, you may return to the "TERMINAL" program to interact directly with MICRONET. Simply type in BREAK M 1 F 0 0 G . It's a good idea to get back to MICRONET and do a SAV command, to place the file in your permanent disc space.

6. To load an OSI BASIC program that has been stored in a MICRONET file, a) Start up system with ACIA switched to input from tape. b) Cold start BASIC, and use 7500 as the memory size. c) Go to Monitor (BREAK M). d) Type in L to load the "TERMINAL" program from tape. e) Go to monitor (BREAK M). f) Type in 1 F 0 0 G to start up the "TERMINAL" program. g) Log onto MICRONET. h) Type in OLD XXXX.DAT to load in the desired file to your MICRONET workspace. i) Type in T Y P (NO CR). j) Warm start BASIC (BREAK W). k) Type in SAVE:PRINT LOAD (CR). l) Watch your program load. All other details of communication are taken care of automatically. After the program loads, your C1 will be hanging in the LOAD state. m) When done, you may return to the "TERMINAL" program to interact directly with MICRONET. Simply type in BREAK M 1 F 0 0 G

While it may not seem elegant to do the manual switching between OSI BASIC and MICRONET, this technique involves minimum overhead (TERMINAL is only about 500 bytes) and is concrete. It may seem a little odd at first to be operating on two levels, but it is easy to get used to. This method has almost as much power as "smart terminals" and the MICRONET EXECUTIVE program which is not available for C1P.

You must note that OSI BASIC and MICRONET BASIC are vastly different. That is why I recommend .DAT as the file-name extension rather than .XBA . While you can't run many of your programs on MICRONET, these methods allow you to use MICRONET's disk as program storage. Just eliminating the handling of numerous cassette tapes is a blessing! Manual chaining of very long programs is possible. You can organize a long program into subsections that are individually stored as MICRONET files, then call up the appropriate sections into your C1 as needed; all the while using MICRONET as a sort of floppy disk. You can also copy software to other users by allowing them to copy your MICRONET files.

After you load a program, you may disconnect from MICRONET and do a BASIC warm start to run the program on your C1. Because 7500 is used as the memory size, there is no danger of over-writing "TERMINAL" . "TERMINAL" may be re-activated at any time by BREAK M 1 F 0 0 G . Use normal caution to avoid over-writing it.

I am now working on a version of KNAPSACK which will read and write encrypted text files using MICRONET's FILGE package. In this version, all of the menu functions of KNAPSACK will be stored as separate files which can be loaded into the C1 and executed. A program in MICRONET BASIC, running on MICRONET, will be the user interface. With these techniques, I hope to bring the security of public key cryptography to bear on the electronic mail features of MICRONET!

Only one other thing to watch out for; when you dump a program into a MICRONET file, MICRONET automatically places several extra spaces between the line number and the first character of the line. If your OSI BASIC program has some very long lines, you'll lose a few characters at the end when you reload the program from the MICRONET file. I recommend just avoiding very long program lines. This also causes a program to occupy more memory when recalled from MICRONET than it did when written on your C1.

BASIC BOOTSTRAP FOR "TERMINAL"
The following BASIC program will generate "TERMINAL" in the RAM locations 1E2D to 1FFF. The BASIC program will also enter the code at 1F00 itself, and begin to run automatically. This program essentially copies some of the monitor into RAM and makes a couple of changes. Key locations and their contents are:
7775 ($1E5F) TERMINAL LINE LENGTH (Preset at 20 characters) (LINE 260).
7936 ($1F00) ENTRY POINT (LINE 280).

```
  1 REM  BASIC BOOTSTRAP FOR
    "TERMINAL BY D. WOLF,
    PH.D.
  2 REM  FOR OSI C1P ONLY
100 RESTORE
110 A=48941:B=7725:GOSUB 300
120 A=64768:B=7936:GOSUB 300
200 DATA 76,216,31,173,0,240,
        74,144,11,173,1,240
210 DATA 234,234,234,41,127,
        32,45,30,76,4,31,32,
        177,252
220 DATA 169,0,234,234,234,
        76,4,31
230 FOR X=0 TO 33:READ A:
    POKE 8129 + X,A:NEXT
240 POKE 8052,79
250 POKE 7774,169
260 POKE 7775,20
270 POKE 7776,234
280 POKE 11,0:POKE 12,31:
    X=USR(X)
300 FOR X=0 TO 255
310 Y=PEEK(A+X)
320 POKE B+X,Y : NEXT : RETURN
```

"TERMINAL" is an endless loop, dumb terminal. It doesn't do a perfect ASCII decode, but functions admirably. Its functional flow is:

```
READ KEYBOARD          (1F00-1FC3)
 if key depressed goto OUTPUT
READ ACIA              (1FC4-1FD1)
 if no char. goto READ KEYBRD.
DISPLAY CHAR. ACIA (1E2D-1EFF)
GOTO READ KEYBOARD (1FD5-1FD7)
OUTPUT CHAR. ACIA  (1FD8-1FDA)
GOTO READ KEYBOARD (1FDB-1FE2)
```

To exit from "TERMINAL" just hit the BREAK key.

DANIEL WOLF, PH.D.

DANIEL:

Very interesting! However, for those of us who don't have C1P's, it would be still more interesting to see a commented source listing, so perhaps we could modify your routine to work in our computers. What are the 6 NOP's ? Saving space for 2 3-byte instructions, perhaps? Thanks.

AL

* * * * *

CA-9D PIA Layout

PIA Data Register A Layout: (Address $F500)

RSTR	CSTB	FRDY	CHEK	DRDY	LSTB	PRDY	CRDY
(Out)	(Out)	(In)	(In)	(In)	(Out)	(In)	(In)

PIA Control Register A Layout: (Address $F501)

N/U	N/U	1	1	FSTB	1	0	0
		(Set)	(Set)	(Out)	(Set)	(Set)	(Set)

PIA Data Register B Layout: (Address $F502)

DB8	DB7	DB6	DB5	DB4	DB3	DB2	DB1
(Out)	(Out)	(Out)	(Out)	(Out)	(Out)	(Out)	(Out)

PIA Control Register B Layout: (Address $F503)

N/U	N/U	1	1	PSTB	1	0	0
		(Set)	(Set)	(Out)	(Set)	(Set)	(Set)

RSTR - Restore (Clear)
CSTB - Carriage Strobe
FRDY - Feed Ready
CHEK - Device Check (Error Condition)
DRDY - Device Ready
LSTB - Latch (DB9-DB12) Strobe
PRDY - Print Wheel Ready
CRDY - Carriage Ready
FSTB - Feed Strobe
DBn - Data Bit n
PSTB - Print Wheel Strobe

CA-9D Board Addressing

Address	Read	Write
F500	PIA: PA0 thru PA7	PIA: PA0 thru PA7 or DDA0 thru DDA7
F501	PIA: Port A Ctrl	PIA: Port A Control
F502	PIA: PB0 thru PB7	PIA: PB0 thru PB7 or DDB0 thru DDB7
F503	PIA: Port B Ctrl	PIA: Port B Control

CA-9D Interface Pinout

Cable Pin	Interfaced to	Description
1	15 DB10 Latch	Data Bit 10
2	12 GROUND	GROUND
3	NC	Paper Out
4	NC	Ribbon Out
5	NC	Cover Open
6	NC	GROUND
7	NC	NOT USED
8	12 GROUND	GROUND
9	16 DB11 Latch	Data Bit 11
10	22 DB12 Latch	Data Bit 12
11	12 GROUND	GROUND
12	20 PA4	Check
13	18 PA7	Restore
14	12 GROUND	GROUND
15	10 CA2	Feed Strobe
16	12 GROUND	GROUND
17	09 PA6	Carr. Strobe
18	12 GROUND	GROUND
19	NC	NOT USED
20	12 GROUND	GROUND
21	17 CB2	Prt Wheel Strobe
22	12 GROUND	GROUND
23	12 GROUND	Ribbon Lift
24	12 GROUND	Select Printer
25	12 GROUND	GROUND
26	24 PA0	Carr. Ready
27	23 PA1	Print Wheel Ready
28	21 PA3	Device Ready
29	NC	GROUND
30	NC	NOT USED
31	NC	NOT USED
32	12 GROUND	GROUND
33	05 PB3	Data Bit 4
34	19 PA5	Feed Ready
35	12 GROUND	GROUND
36	07 PB1	Data Bit 2
37	08 PB0	Data Bit 1
38	12 GROUND	GROUND
39	06 PB2	Data Bit 3
40	04 PB4	Data Bit 5
41	12 GROUND	GROUND
42	03 PB5	Data Bit 6
43	02 PB6	Data Bit 7
44	12 GROUND	GROUND
45	01 PB7	Data Bit 8
46	14 DB9 Latch	Data Bit 9
47	12 GROUND	GROUND
48	NC	NOT USED
49	NC	NOT USED
50	NC	GROUND

Note: The DB9-DB12 latch is loaded via the DB0-DB3 lines and strobed by PA2 (Active low).

THE END

OSI-FORTH 2.0 / FIG-FORTH

OSI-FORTH 2.0 is a full implementation of the FORTH Interest Group (FIG) version of the FORTH language. It conforms to the FORTH-79 Standards Specification, and includes a resident text editor and 6502 assembler as part of the basic package. The OSI-FORTH system runs under the OS-65D3 operating system on any disk-based Ohio Scientific system, and has access to all DOS commands and resources. The package price is $79.95 for either a 8" or mini disk and manual. The manual is available separately for $9.95, deductible later from package purchase price.

Mr. Clean

Let Mr. Clean pretty up those BASIC programs! This very fast machine language program will resequence (renumber) and/or pack (remove spaces and REMarks), from any OS-65D3 BASIC program. Mini or 8" disk and instructions is $17.95.

VIDEO BANNER

Video Banner puts up to three lines of nine letters and/or punctuation marks each on the OSI 540 video screen. The letters are '3-D', built from graphics elements (resembling letters above). Mini or 8" disk contains the data files for the characters, utility programs to display them on the screen and save screens to disk, and demo program. Documentation provided will enable you to incorporate the data base and display into application programs (educational programs, games, clock programs, display 'signs'). An excellent program for dealers to catch customer eyes! Mini or 8" disk is $24.95 .

ADVENTURE-65

Explore a deep and mysterious cave, but watch out for the dragon and don't spill the lamp oil! Will you find all of the treasure before the lamp burns out? Will your leap across the chasm be successful? How do you deal with the dragon? Can you find all of the rooms in the cave? This micro version of the famous 370 Adventure retains all of the excitement and frustration of the original. No two trips into the cave will be exactly alike, so keep exploring! Requires 8K of memory. Cassette is $9.95, mini or 8" disk is $14.95 .

DISK-PAK 1

A collection of very useful OS-65D3 utilities. DSKPRP prepares a new diskette for use, by initializing any track range and putting in a blank directory. DIR540 uses the full 540 video screen (or terminal) to display the complete sorted directory in one view, allowing file creation with the full directory in view. SAVE automatically creates a file of given length by finding available space for you! It will ALSO optionally figure out the space needed to save whatever program is in a scratch file ("SCR"), create the file and transfer the program from SCR to the new file! All of the convenience of an Apple with the simple sequential files of OSI! COPIER copys the OS-65D3 System onto a new disk with a minimum of disk swaps (uses 24K). ONERR is a program (subroutine) that modifies OS-65D3 to allow your BASIC program to automatically execute any given line number upon detection of an error. BACKUP maintains a backup directory on disk and will restore a directory lost due to a disk error or mistake. Mini or 8" disk is $24.95 .

MANUALS

C4P/C4P-MF Users Manual: This is a nicely typeset manual recently released by Ohio Scientific. $8.95

OSI-SAMS Service Manuals: These excellent manuals contain schematics, logic diagrams, parts lists, scope traces and other valuable service data. C4P/C4P-MF manual is $14.95 (C4 manual for C8, too). C1P/SB11 is $7.49 C3/C20EM manual is $38.95 .

CONNECTORS

A set of four male and four female connectors of the type used by OSI for board connection to the backplane and other I/O. $4.95

INFO

Our Floppy Disk Information pack provides the information you need to upgrade your OSI ROM BASIC system to a disk-based system. $3.49

** Allow 30 days for delivery. Florida residents add 4% Sales Tax. Specify system model. **

Send for our new FREE catalog containing many programs for ROM or disk systems, information packs, hardware, and other useful goodies for Ohio Scientific systems!

TECHNICAL PRODUCTS COMPANY
P. O. BOX 12983
GAINESVILLE, FLORIDA 32604

ED:

Among the many advantages to owning OSI equipment is the opportunity to use Dwo Quong Fok Lok Sow's word processor along with it. WP6502 is remarkably easy to use, loads quickly, and occupies less than 4K RAM. At $50.00 for the cassette version it is a genuine bargain.

Recently I tried using it as a program text editor and discovered that loading and unloading programs is a bit tricky. WP6502 recognizes the end of a text as the @ sign. Since this symbol does not appear at the end of a SAVED BASIC listing, WP6502 simply hangs waiting for the @ which never comes. There are two ways to overcome the problem.

The easier way is to re-SAVE the program and with the recorder still in SAVE mode, type in @ after the usual OK. With this addition WP6502 will know the text is completed, do its housekeeping and come up with the MENU.

The second way is to READ the program just as it is into WP6502 and allow it to hang. Then do the following: 1) Stop the tape, press BREAK and Warm start. 2) Go into Global Edit and respond to the FROM prompt with about 10 spaces and to the TO prompt with a Carriage Return. 3) To the Y/N choices which show no text at all, press Y. The MENU will now appear, the text will be consolidated and the @ sign will be found at the end of the program.

When saving a program to tape after editing it, press V for VIEW and Y for PRINTER. For #Lm10 and #P01, press 00 and be sure to wait until you have your tape running before pressing RETURN for Ho?. Your program will be saved in runnable form.

If you want to save your program in WP6502 editing form (not a bad idea when you anticipate the need to make additional changes), just WRITE the text to a different tape using W from the MENU.

I have found WP6502 useful in removing spaces and consolidating lines in existing programs. For me, writing new programs in the TYPE format is somewhat awkward. There is only one unorthodox key stroke required -- LINE FEED instead of RETURN at the ends of lines.

Equipment: C2-8P, cassette, 36K RAM.

Ian A. Morton
St. Paul, MN

Ian:

Now figure out how to get rid of the dots and page numbers at the tops of the pages.

AL

* * * * *

ED:

I have a C3-OEM with a Soroc IQ120 and an Okidata SL 160 line printer. I have the extra 8K of memory to run CP/M but I have a slight problem. I have obtained software from CREATIVE SOFTWARE and CPUG and cannot get either disk to copy. I don't dare experiment with the original disks and no one else around has CP/M. I can load "OBASIC" and write my own programs but no luck with outside stuff. Any ideas? Any help?

John Fijalkowski
Neffs, OH

* * * * *

ED:

In volume 2, #1, page 11, C.D. Lombard wanted to find out how to program the character generator chip. Other OSI owners may have wondered about this mysterious code. At least that's how I thought of it until I plugged the ROM into an EPROM programmer and read the contents using the programmer's read mode. Assuming that the C.G. ROM in the C2 is similar to the one in the Superboard/C1P, the code is quite simple. The ROM could be replaced with a 2716 EPROM that has been programmed with any character set which your imagination can dream up. The Superboard/C1P C.G. ROM code is as follows:

HEX ADDRESS	HEX CONTENTS	BINARY 01234567	VIDEO IMAGE
00 00	5A	01011010	* ** *
00 01	7E	01111110	******
00 02	5A	01011010	* ** *
00 03	18	00011000	**
00 04	18	00011000	**
00 05	5A	01011010	* ** *
00 06	7E	01111110	******
00 07	5A	01011010	* ** *
02 30	3E	01111100	*****
02 31	02	01000000	*
02 32	02	01000000	*
02 33	1E	01111000	****
02 34	02	01000000	*
02 35	02	01000000	*
02 36	02	01000000	*
02 37	00	00000000	

Of course this not a complete listing. The first character shown is #0 in the character set... a race car. The second character is #70 in the character set... letter "F". Imagine that the ones are white dots on the video screen and the zeros are un-dots. Armed with this knowledge, you can program a 2716 (single 5V.) EPROM with your own 256 character set.

You will probably find it necessary to change the chip enabling logic on pins 18, 19 and 20. The read mode of the 2716 requires that pins 18 and 20 be tied to ground (or a logic low) and pin 21 be tied to +5V.

Pete Hitt
La Luz, NM

Pete:

Maybe the smart guys will know just what you mean by "change the chip enabling logic", but for us dumb guys, could you be more specific? I don't even know if you mean I have to change some logic in a program, or maybe just ground pins 18-20, or what? Any way, thanks a million for the information... it is just what a lot of guys (smart and dumb) have been looking for!

AL

* * * * *

ED:

RE: LEVEL 3 HORROR STORIES

Operating three different systems using Level 3 with multiple users, I too have had my share of problems with writing data to wrong files. And, it hasn't been a "rare" occasion, either.

We put in the fixes suggested in OSI's Bulletin 28... and happy to say that we haven't had a problem since... with about 5 weeks of operation.

So, with our fingers crossed, we go forth with a great deal of confidence!!! Seriously, the problem seems fixed!

Douglas J. Mckilligan
Minneapolis, MN 55420

READERS:

Contact your dealer for a copy of the fix.

AL

* * * * *

OSI SOFTWARE FOR OSI OSI

A JOURNAL FOR OSI USERS!!

The Aardvark Journal is a bimonthly tutorial for OSI users. It features programs customized for OSI and has run articles like these:

1) Using String Variables.

2) High Speed Basic On An OSI.

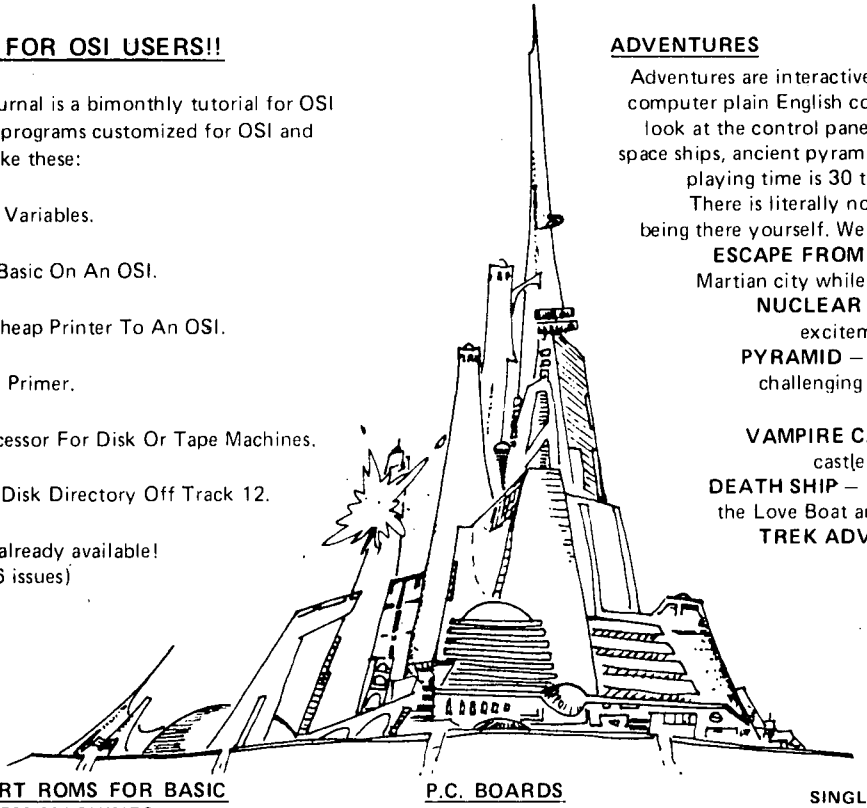
3) Hooking a Cheap Printer To An OSI.

4) An OSI Disk Primer.

5) A Word Processor For Disk Or Tape Machines.

6) Moving The Disk Directory Off Track 12.

First year issues already available!
$9.00 per year (6 issues)



ADVENTURES

Adventures are interactive fantasies where you give the computer plain English commands (i.e. take the sword, look at the control panel.) as you explore alien cities, space ships, ancient pyramids and sunken subs. Average playing time is 30 to 40 hours in several sessions. There is literally nothing else like them – except being there yourself. We have six adventures available.

ESCAPE FROM MARS – Explore an ancient Martian city while you prepare for your escape.

NUCLEAR SUBMARINE – Fast moving excitement at the bottom of the sea.

PYRAMID – Our most advanced and most challenging adventure. Takes place in our own special ancient pyramid.

VAMPIRE CASTLE – A day in old Drac's castle. But it's getting dark outside.

DEATH SHIP – It's a cruise ship – but it ain't the Love Boat and survival is far from certain.

TREK ADVENTURE – Takes place on a familiar starship. Almost as good as being there.

NEW SUPPORT ROMS FOR BASIC IN ROM MACHINES

C1S – for the C1P only, this ROM adds full screen edit functions (insert, delete, change characters in a basic line). Softwave selectable scroll windows, two instant screen clears (scroll window only and full screen), software chose of OSI or standard keyboard format, Bell support, 600 Baud cassette support, and a few other features. It plugs in in place of the OSI ROM. NOTE: this ROM also supports video conversions for 24, 32, 48 or 64 characters per line. Replaces video swap tape on C1P model 2. All that and it sells for a measly $39.95.

C1E/C2E for C1/C2/C4/C8 Basic in ROM machines. This ROM adds full screen editing, software selectable scroll windows, keyboard correction (software selectable), and contains an extended machine code monitor. It has breakpoint utilities, machine code load and save, block memory move and hex dump utilities. A must for the machine code programmer replaces OSI support ROM. Requires installation of additional chip when installed in a C2 or C4. C1 installation requires only a jumper move. Specify system $59.95.

DISK UTILITIES

SUPER COPY – Single Disk Copier
This copy program makes multiple copies, copies track zero, and copies all the tracks that your memory can hold at one time – up to 12 tracks at a pass. It's almost as fast as dual disk copying. – $15.95

MAXIPROSS (WORD PROCESSOR) – 65D polled keyboard only - has global and line edit, right and left margin justification, imbedded margin commands, choice of single, double or triple spacing, file access capabilities and all the features of a major word processor – and it's only $39.95.

P.C. BOARDS

MEMORY BOARDS!! – for the C1P. – and they contain parallel ports!
Aardvarks new memory board supports 8K of 2114's and has provision for a PIA to give a parallel ports! It sells as a bare board for $29.95. When assembled, the board plugs into the expansion connector on the 600 board. Available now!

PROM BURNER FOR THE C1P – Burns single supply 2716's. Bare board – $24.95.

MOTHER BOARD – Expand your expansion connector from one to five connectors or use it to adapt our C1P boards to your C4/8P. - $14.95.

ARCADE AND VIDEO GAMES

GALAXIA one of the fastest and finest arcade games ever written for the OSI, this one features rows of evasive, hardhitting, dogfighting aliens thirsty for your blood. For those who loved (and tired of) Alien Invaders. – P.S. The price is a giveaway. SPECIFY SYSTEM!
Cassette $9.95 – Disk $12.95

TIME TREK (8K) – real time Startrek action. See your torpedoes move across the screen! Real graphics – no more scrolling displays. $9.95

INTERCEPTOR C1P ONLY! An all machine code program as fast and smooth as the arcades. You use your interceptor to protect your cities from hordes of enemy invaders. A pair of automatic cannons help out, but the action speeds up with each wave of incoming ships. The fastest and most exciting C1P game yet.
C1P Cassette $19.95

MINOS – A game with amazing 3D graphics. You see a maze from the top, the screen blanks, and then you are in the maze at ground level, finding your way through on foot. Realistic enough to cause claustrophobia. – $12.95

SINGLE STEPPER / MONITOR

This is probably the finest debugging tool for machine code ever offered for OSI systems. Its' trace function allows you to single step through a machine code program while it continuously displays the A, X, Y and status registers and the program and stack pointers. You can change any of the registers or pointers or any memory loca- at any time under program control. It takes well under 1k and can be relocated anywhere in free memory. It is a fine tool for all systems – and the best news of all is the extremely low price we put on it. – Tape $19.95 – Disk $24.95

FOR DISK SYSTEMS – (65D, polled keyboard and standard video only.)
SUPERDISK. Contains a basic text editor with functions similar to the above programs and also contains a renumberer, variable table maker, search and new BEXEC* programs. The BEXEC* provides a directory, create, delete, and change utilities on one track and is worth having by itself. – $24.95 on 5" disk - $26.95 on 8".

AARDVARK IS NOW AN OSI DEALER!

Now you can buy from people who can support your machine.

– THIS MONTH'S SPECIALS –

Superboard II	$279
C1P Model II	429
C4P	749
8K 610 board for C1P	269
Epson MX-80 printer with RS232 installed	595

. . . and we'll include a free Text Editor Tape with each machine!

True 32X32 Video Mod Plans for C1P
(4 Chips $3.00 Crystal Required)
$7.95

VISA

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.

master charge

Aardvark Technical Services • 1690 Bolton • Walled Lake, MI 48088
(313) 669-3110

OSI OSI

ED:

Brian Winkel's question (October 1980) concerning operation of a line printer utilizing the OS-CP/M was similar to our first encounters with this system on our C3-OEM. The BASIC Input Output System (BIOS) as delivered in the CP/M is configured to support a parallel line printer at $F400. It turned out to be a simple matter to modify the BIOS to support Port 1 of a CA-10X Board, located at $CF00 & $CF01, as the list device.

A copy of the BIOS Assembly source should have been included on your master disk under the file name ZBIOS ASM. If you examine this source listing, you will find that item 6 in the vector table is the list device. Any program such as BASIC, word processor, etc., which operates in the CP/M environment must use this vector entry point to access the listing device (usually the printer). In the case of OBASIC, it is delivered with (JMP LIST). Following down the BIOS source listing, you will encounter the label LIST. Using the editor, a rewrite of this driver, as follows, will direct output to Port 1 of a CA-10X card:

```
LIST    PUSH PSW     ;    Save the processor status
        LDA 0CF00H   ;    Get ACIA status
        RAR          ;    To see if it's clear
        RAR          ;
        JNC LIST+1   ;    Loop if not clear
        MOV A, C     ;    Put character in accumulator
        STA 0CF01H   ;    Output it
        POP PSW      ;    Retrieve Status &
        RET          ;    Return
```

Assemble the modified BIOS source into ZBIOS.HEX, then, overlay the system as follows:

```
Step 1  DDT CPMXX.COM      (Load the previously saved
                            copy of the CP/M system
                            NOTE XX=48 for 48K System)
Step 2  IZBIOS.HEX         (Opens a file control block
                            for new BIOS)
Step 3  R6080              (Overlays new BIOS into system
                            IMAGE.
                            NOTE: 6080 is the IMAGE bias
                            for 48K Memory)
```

The overlay has now been placed in the copy of CP/M which is in the Transient Processor Area (TPA). Use SYSGEN as follows to place this revised copy into the system area:

```
Step 1  Type SYSGEN
Step 2  "Source Drive Name" - Type CR (the image is already
                              in memory)
Step 3  "Destination Drive" - Insert an initialized disk in
                              Drive B then type B
Step 4  "Destination Drive B then type Return" - Type CR
Step 5  Move the disk from Drive B to Drive A & attempt a
        reboot to ensure the SYSGEN has been successful.
Step 6  Move the new disk back down to Drive B and place the
        old disk in Drive A.
Step 7  Use the peripheral interchange program (PIP) command
        to move the files from A to B. This will result in
        a new master copy in Drive B.
```

The procedure is a bit tedious but once accomplished eliminates the necessity for doing several pokes every time CP/M is booted. Please note that this is not the only way one can modify the list portion of BIOS. A jump to CA1-100U (CA-10X Output) would probably accomplish the same task. This would also allow port selection since the label CA10X is a port selection word similar to memory location 19798 in the OS-65U.

A significant problem with OS-CP/M is that the cold start does not initialize the ACIA's on the CA-10X Board. This can be accomplished for Port 1 by using the pokes provided by Brian's letter. If you are using multiple ports, as we are on our system, the following program will initialize all sixteen ports:

```
; INIT
;
ORG 0100H
;
  LOOP
  LXI H,0CF20H; Point to
                port 16+2
  DCR L       ; back up two
  DCR L       ;
  MVI M,3     ; initialize the
                port
  MVZ M,17    ;
  JNZ LOOP+3  ; do another
                till done
  RET
  END
```

After assembling and loading the program, it can be run, after bootup, as a transient command.

In the area of word processing we are using Dwo Quong's WP-6502 with the Heath H-14 as an edit printer and a serial NEC Spinwriter as the final copy printer. Simple words of praise would not do justice to the WP-6502 word processing software. Our local dealer (Radecon) allowed us to examine both the WP-6502 and the WP-2 packages prior to purchase. Needless to say the WP-6502 was the unquestionable favorite of everyone who used the two packages during this trial period. At $125 the package is simply amazing. If the add on package to provide proportional spacing and right justification ($50.) is as well structured, the combination will be the most powerful word processor, dollar for dollar, available for OSI equipment.

Tim Davis
Tucson, AZ

* * * * *

ED:

Enclosed is a copy of a real time clock program that will run on a C1P-SBII. The original article was in 5/80 Kilobaud. It was written in machine language and in addition to a real time clock it has an alarm function. I rewrote the program and took the alarm out. In order to make the program work you need to jump NMI to C15 (U61 pin 11) thru a SPST switch. Since the program is located at $0130 you can do a cold start and the program will still run. The time is displayed below the bottom printing line so it doesn't effect the running of other programs.

Keep up the great work on Peek (65)!

```
900 REM MOD FROM KILOBAUD
910 REM 5/80 BY B HOYT
920 REM
930 REM
940 REM MOD BY T SHINGARA
950 REM NMI MUST BE WIRED
960 REM TO C15 WITH A SPST
970 REM SWITCH.
980 REM FOR C1P ONLY
990 PRINT"REAL TIME CLOCK"
1000 DIMCL(86)
1010 FORX=304TO389
1020 READCL
1030 POKEX,CL
1040 NEXTX
1050 PRINT"12 OR 24 HR";
1060 INPUTQ
1070 IFQ=12THENPOKE341,19
1080 IFQ=12THENPOKE345,1
1090 INPUT"SET HRS.";X
1100 GOSUB1500
1110 POKE240,X
1120 INPUT"SET MIN.";X
1130 GOSUB1500
1140 POKE241,X
1150 INPUT"SET SEC.";X
1160 GOSUB1500
1170 POKE242,X
1180 POKE239,0
1190 POKE243,96
1200 POKE244,0
1210 PRINT"TURN SWITCH ON"
1220 PRINT"AND CLEAR PROGRAM"
1230 END
1300 DATA72,198,243,208,61
1310 DATA138,72,152,72,248
1320 DATA169,60,133,243,162
1330 DATA2,24,181,240,105,1
1340 DATA149,240,202,48,10
1350 DATA201,96,208,14,169
1360 DATA0,149,241,240,236
1370 DATA201,36,208,4,169
1380 DATA0,133,240,162,2,160
1390 DATA7,181,240,32,116,1
1400 DATA169,58,153,177,211
1410 DATA136,202,16,242,104
1420 DATA 168,104,170,104,64
1430 DATA32,123,1,74,74,74
1440 DATA74,72,41,15,9,48
1450 DATA153,177,211,136
1460 DATA104,96
1500 IFX>60THEN1000
1510 IFX>59THENX=X+36:RETURN
1520 IFX>49THENX=X+30:RETURN
1530 IFX>39THENX=X+24:RETURN
1540 IFX>29THENX=X+18:RETURN
1550 IFX>19THENX=X+12:RETURN
1560 IFX>9THENX=X+6:RETURN
1570 RETURN
```

T.M. Shingara
Harrisburg, PA

* * * * *

ED:

We are using a Superbaord II, with hardware modifications, to drive a Model 33 Teletype. Since this is rather slow (110 baud), we would like to be able to reduce the number of nulls after the CR, which result from the PRINT statement in Basic, from 10 to 2 or 3. Neither the Basic command NULL nor POKE 13,x do this. It seems these only add additional nulls. I think 2 or 3 nulls would be sufficient to give the Teletype a chance to perform a carriage return. Does anybody out there know how to do this?

Also, has anyone done a review of the substitute ROMs offered by Aardvark or others for the S II? We can live with the video, backspace, etc., but would really like to get rid of the garbage collector problem.

If anyone is interested in how I interfaced the S II and the teletype. I would be happy to write in with the details.

Jay Jackson
Falls Church, VA

Jay:

Of course we are interested!!

AL

* * * * *

ED:

I disagree strongly with U2 columnist Greg Stevenson that FORTH is a "high level" language. If we can agree that COBOL, PASCAL and some versions of BASIC are high level, we could see that we can all make some sense out of these programs even if we are proficient in only one. (You'll note that I said "some" versions of BASIC! Those with only two character variable names such as used by OSI can no longer be called "high level.") This fairly easy level of understanding is not true for FORTH. I whole-heartedly agree with his comment that FORTH was not designed to be friendly to beginners. In fact, I suspect it may only be friendly to those who enjoy machine language programming - because there is no similarity between FORTH and "conventional" programming.

With the declining cost of computer power, I believe that languages of the near future will be easier to use - not more difficult. This is especially true for business languages where users should be concerned about the long term capability of support of their programs. While an individual programmer with the talents of Greg Stevenson or Larry Hinsley may very well be able to write concise, fast business programs in FORTH - as a dealer, I would not want to use these programs unless I could count on the availability of programming support locally. Most business BASIC programmers could "switch" to COBOL or PASCAL. I doubt that most would want to switch to FORTH.

The only strength I see at the moment is that there may be some "utilities" developed in FORTH which will be 65U compatible. While all of us would like to see more support given to 65U, I, for one, would far rather see 65U compatible PASCAL - or even more powerful BASIC such as offered in the Z80 OASIS operating system. The file handling capabilities combined with the FIND feature of OSI's 65U make it potentially outstanding as a truly high level system. FORTH, even 65U compatible, should not be considered a "high level" replacement for BASIC.

Tom Stover
Gering, NB

* * * * *

ED:

I've just installed the BASIC 4 and MONITOR ROMs from Progressive Computing in my C1. In conjunction with the 64 character video hardware mod, they work perfectly!

Progressive Computing took my advice and are only offering the BASIC 4 ROM with the MONITOR ROM. They are symbiotic and are a waste of money if not used together. Also, they reduced the price by $10.00 (not my advice, but who's complaining?) for the pair.

In addition to a cursor you can direct to any location (in Edit mode), the MONITOR allows one to VIEW data at the serial port without LOADing it. There's even an instant screen clear routine that can be done with just one PRINT statement, or machine code JSR.

Finally, there's a PRINT AT feature and a Selectable Scroll Window. These are most useful in graphics applications.

Originally I bought a C1 (superboard). I've added 4K more RAM, 64 character video and the two ROMs, giving me a quasi - C4. The beauty of it, financially, was that I built this machine a chunk at a time as my budget permitted. Plus, there was the joy of doing it myself! It's like living between the worlds of "homebrew" and "turnkey", with the pleasures of both.

Bruce Showalter
Abilene, TX

* * * * *

ED:

This is in reference to the letter I sent you (published in the Peek (65), March 1981 issue) about my 8P-DF.

What was the obvious solution to a problem, wasn't. Somehow, when I moved the fluorescent light the problem went away, for two weeks. Then it came back, without the light. Mr. Morris Harvey of DATA BUSS in Grayslake, IL, who has advised me on other occasions, had the solution. The CA-9 board that was installed recently was too close to upper memory and memory was picking up stray signals from it. Moving the boards apart solved the problem permanently.

Raymond G. Bruns
Chicago, IL

* * * * *

ED:

In this month's letter I'd like to write about the C1P/SBII expansion sockets, followed by a discussion of interfacing a Southwest Technical Products PR-40 printer, and end up asking a question or two of the C1P/SBII users.

The basic Superboard and C1 have an expansion socket located on the component side of the board, just behind the keyboard. It's labeled in the schematics as J1 and is a 40 pin DIP socket. After pouring over the free schematics OSI supplies, I finally decoded what signal is available at each pin. Refering to figure 1 we see that the 16 address lines, 8 data lines, clock line, read/write line, 2 interrupt lines, and a data direction line are supplied. Notably missing are the reset, ready, and sync lines, thus making some projects (such as a single stepper or direct memory access) difficult; however, there is an unconnected pin that could be used for one of those signals if desired. With the exception of data direction (DD) all the signals are standard microcomputer stuff. DD is required because of the way the 6502 and the C1/SB are designed.

With S100 bus systems there are actually 16 data lines, 8 inbound to the CPU and 8 outbound to the peripherals; in most other micros only 8 lines are supplied (thus reducing the size and cost of the bus) and so data must flow both directions on the same lines. To further complicate matters, each chip hooked up to a bus line takes a bite out of the current available on that line; this is called fan-in and the number of bites that a chip can supply is called fan-out. The upshot of all this is the fact that only so many devices can be hung on a bus line; putting too many on causes erratic operation or no operation at all. To allow a micro chip to drive many peripherals we use a thing called a buffer. However, since the C1 has only 8 data lines, the buffer has to be able to talk in either direction. Furthermore, these buffers have to be told which direction to speak since to talk both ways at once would wreak havoc. This is where DD comes in. DD is a signal, GENERATED BY A PERIPHERAL, that tells the data-line buffers on the C1/SB board

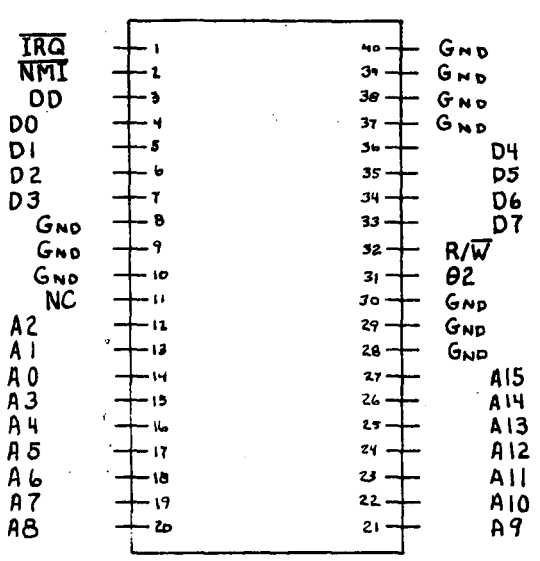
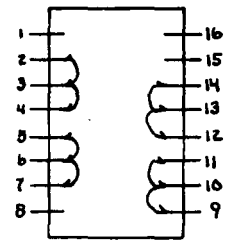


FIGURE 1

J1 EXPANSION SOCKET

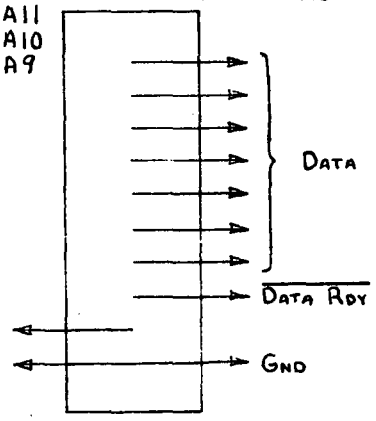
FIGURE 2



8T28 SUBSTITUTES DIP HEADER WIRING

FIGURE 3

PR-40 SIGNAL REQUIREMENTS



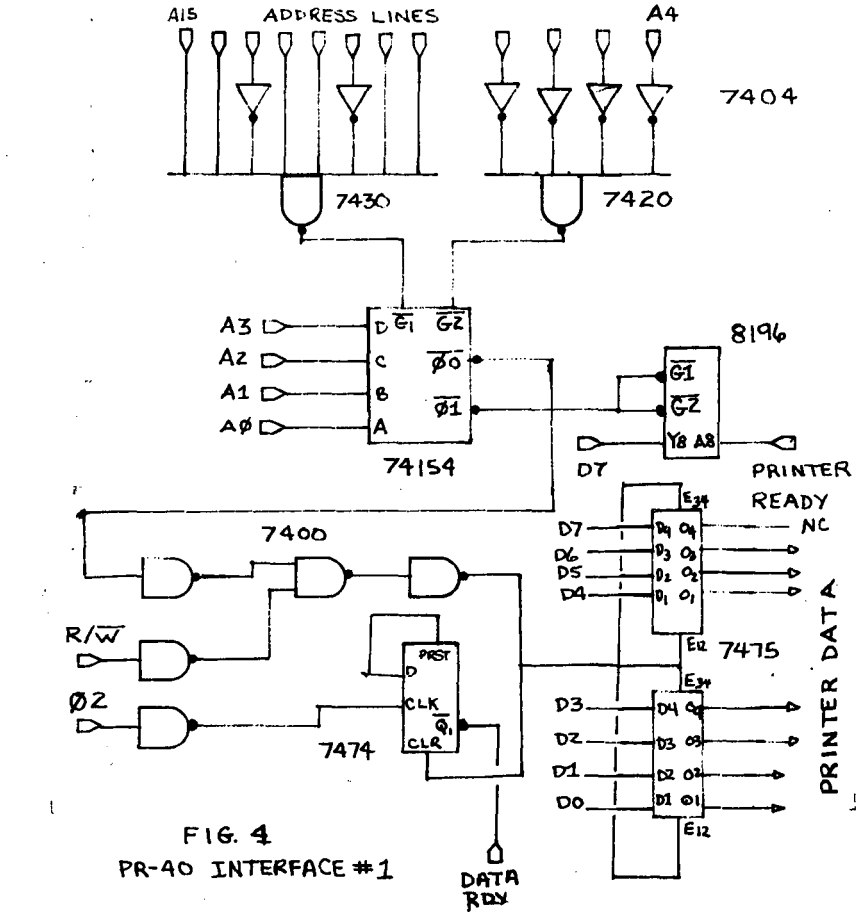


FIG. 4
PR-40 INTERFACE #1

which way they should be talking. Unfortunately, OSI doesn't ship these buffers (8T28's, U6 and U7, located just next to the 6502 CPU chip) with the C1/SB; you have to buy the 610 board to get them. When I started to experiment with the expansion socket I couldn't find a source for them and ended up putting in two Radio Shack 16 pin DIP headers wired to always pass data instead (see figure 2 for wiring details). In doing so I was limiting the number of devices I could hang on the bus, but I was only interested in wiring a printer interface so this was no hardship.

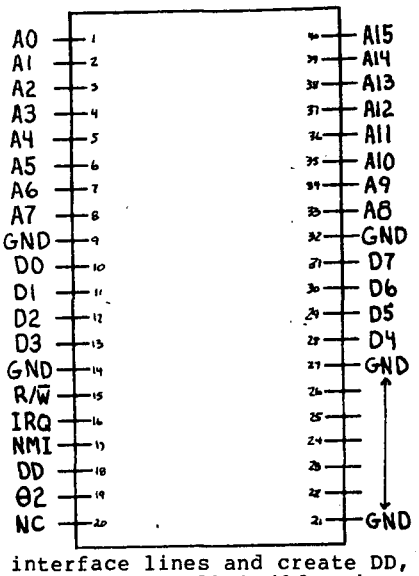
After deciphering all this, the next step is to bring all the signals to the outside world. I did this by buying an A-P Products cable with 40 conductors and a male 40 pin plug on each end, plugging it into J1, and bringing it out the front of the case, under the keyboard. I plugged it into a Vectorboard and was all set up for the next step which was...

To create a parallel interface for a SWTPC PR-40. This is a forty column inpact printer, costing $250 in kit form (i.e., CHEAP!) Since it is upper case only, its main use is in program listing which was all I cared about. Figure 3 shows its signal requirements.

After much plugging and chugging I came up with a circuit (figure 4) that would allow me to use the PR-40. Address $DB00 is the data port and the high bit ($80) of address $DB01 is printer status. The circuit worked and I was marvelously happy for eight months.

That is, until I bought a 610 expansion board. The 8T28 buffers I had bypassed were now supplied (and required), so at a minimum I had to generate another signal (DD). To be on the safe side I reviewed the 610 schematics for the new expansion socket pin-outs (J2). Imagine my shock when I discovered the signals were arranged in a completely different order! (See figure 5) I was devastated, but forging ahead I decided that since I was going to rewire all the

FIGURE 5
610 BOARD EXPANSION SOCKET J2



interface lines and create DD, I might as well build a home for the circuit board the interface was now living on. Several weeks passed and everything was completed. I plugged in the interface and started a program to generate some print lines. Unfortunately, the printer was printing bad data. Further investigation revealed that data was arriving at the printer correctly but the printer ready status bit was only showing up correctly about 40% of the time. Because only the inbound data was wrong I thought that the problem might be the fact that I wasn't gating the signals correctly. I redesigned and rewired the interface (inbound only, see figure 6) but it still doesn't work. So, finally, my questions are:

Has anyone used the expansion interface on the 610 board? If so, how?

Has anyone bought the OSI expansion cable, the one that goes between the 610 board and the OSI 48 pin Molex bus? Any comments? I noticed 'snow' on my video display whenever I hooked up my 40 conductor cable; no problem existed before the 610 board was installed. Would some hardware guru examine figure 6 and tell me if I've done anything wrong?

Comments, anyone??

Michael B. Carroll
Plainsboro, NJ

* * * * *

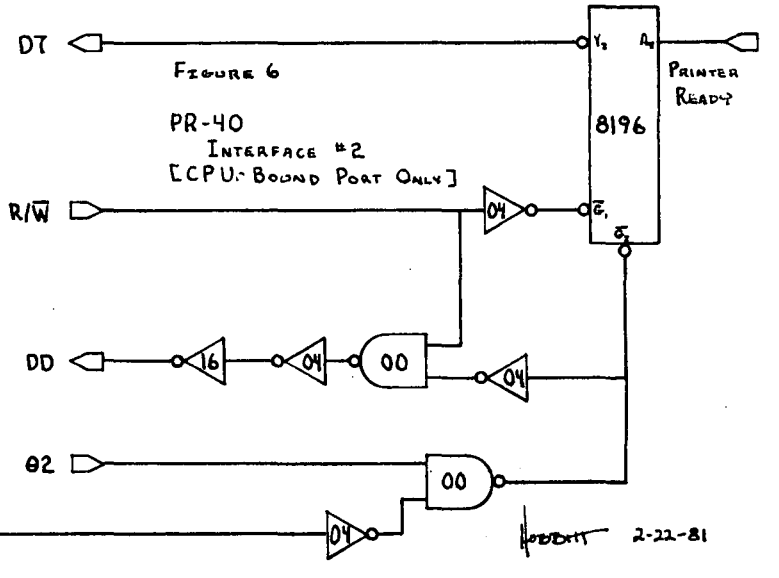
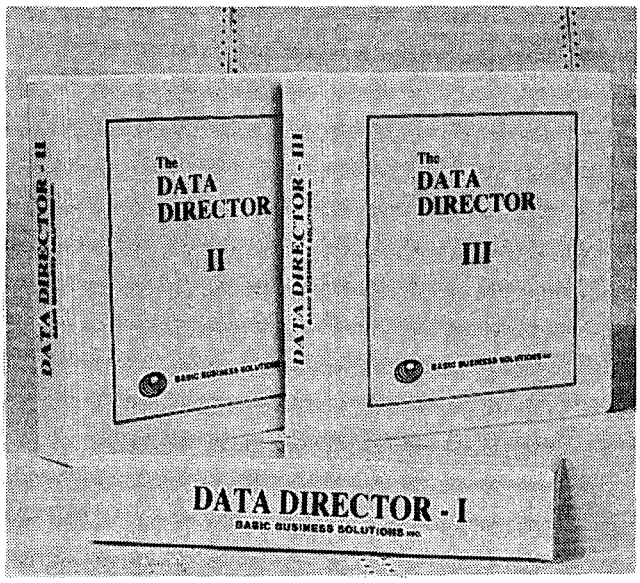


FIGURE 6
PR-40
INTERFACE #2
[CPU-BOUND PORT ONLY]



The Data Director

easy to use Professional Software for easy to get Professional Results

THE DATA DIRECTOR PROJECT

The Data Director is a file management system designed for associations, direct mailers, sales & marketing departments, recruiters, and companies that have the following profile:

- Maintain 1 to-5 files on paper or through a service bureau.
- Files are the heart of your operation (membership lists, subscription lists, sales inquiries, resumes, etc.).
- First time computer user.
- Staff characterized by high turnover.
- One major task associated with each file. Renewals, mailing labels, inquiry response, regular peak periods, etc.

The system is based upon work that began on Burroughs mainframe computers in the mid-1970s. In 1979 it was installed on OSI computers at four carefully selected alpha-test sites. These sites ranged from a church in a Gary, Indiana ghetto to the fourth-ranked graduate business college in the nation. One site, a direct mail service in Chicago, entered over 300,000 names and addresses into their system during 1979 and 1980.

Before a line of code was written, we profiled the average operator. He/she was a high school graduate, an average typist, low paid, low seniority, filling a high-turnover position. How could we work with this person?

We began by isolating the disk operating system, thereby reducing the training time and the chances of a costly mistake. The system includes a diskette formatter, file-to-file copier, directory report, file create, file rename, file delete, disk packer, and diskette copier. It traps disk errors and displays error messages in English. For example, an open disk drive prompts the message, "Drive door open. Close door and try again."

Next we developed a machine language terminal controller to simplify data entry. Displays are paged, not scrolled. Records are presented as forms automatically. If your terminal offers it, we use full and half intensity to highlight data, and cursor control keys to move around the display. We emulate all the features found on the most expensive terminals—character insert/delete, forward/reverse tab, field erase, strike-over, rubout, etc. Existing data is edited, not retyped.

The bottom line of each display is reserved as a status and command line. The operator uses a vocabulary of 30 English command words to begin each task. For example, ADD adds a new record to a file, REBUILD reconstructs an existing file into a new format. SORT sorts a file on up to 5 keys in ascending or descending order for each key.

Help is available on-line through a HELP dictionary which explains the purpose and function of each command in the vocabulary.

A 225-page manual thoroughly documents system operation, and supplies additional information to get you started properly.

HARDWARE REQUIREMENTS

Computers: C2-OEM, C2-D, and all C3 models running under the OS-65U operating system. Data Director II and III support up to 8 users.

Terminals: All 24 by 80 cursor addressable terminals are supported. Models are selected from a menu. Multi-user systems may use different terminal models at each station.

Printers: All printers supported by the OS-65U V1.2 DOS. Printers are selected from a menu. Line and page lengths are establish globally.

THE DATA DIRECTOR I

Our base system is optimized for floppy disk systems. Records may contain up to 99 fields of information. Each field may contain up to 71 characters. Three field types are recognized alpha, numeric, and MM/DD/YY dates.

Existing OS-DMS compatible files can be read and maintained by the system (although the reverse is not true). We hope that OS-DMS users will consider upgrading to our system.

The REPORTS command offers an inquiry report that can be sent to the console or printer, a mailing label generator, and a conditional report writer with statistical analysis. All reports, and most of the utilities, feature a program halt on CTRL-C which allows you to halt the report and abort or continue at your leisure.

THE DATA DIRECTOR II

Although it runs on a floppy disk system, our second system is optimized for a hard disk system. It supports up to 8 users (16 upon request), and was designed for files up to 20,000 records long. All version I features are incorporated.

The operating system utilities are extended to include a fast floppy dumper to back up hard disk files to floppy diskette. (Mag tape support is available separately.)

A duplicates report scans files for duplicate records. As an option, it can count all the occurrences of a duplicate field, like breaking down zipcode distributions.

The report saver captures report definitions and saves them by name. Our users have defined reports with exotic names like "In Work," "Delinquents," "Approved Loans," "Past Due," and "Prod Work Orders." The reports offer conditional selection and statistical analysis.

The mailing label generator is expanded into a complete subsystem aimed at professional mailers. The operator defines a label definition, giving it a name, the label's size, fields which are to appear on it, messages like "After 5 days return to:" and even default values like "Occupant" that are to appear if the data is missing. By selecting a definition, the operator can print on pressure sensitive labels, envelopes, 3 by 5 cards, stationery, etc. Print options include printing labels 1 to 5 across and repeating labels up to 99,999 times.

THE DATA DIRECTOR III

Our top of the line system is designed for very large files, 20,000 to 100,000 records long. It incorporates all of the features of versions I and II, and adds a new "linked list" storage technique.

From the operator's viewpoint, a linked file appears to be resorted automatically whenever a record is added or deleted. A file can be ordered in up to 5 different ways. For example, you could order a membership roll by (1) zipcode, (2) zipcode and name, (3) name, (4) renewal date, and (5) sex, age, and marital status.

Seek times are vastly reduced. You could find the first occurrence of renewal date 11/01/81 in a 100,000 record file in under 6 seconds, then switch keys and browse forward or backward through the file by name.

CONSIDER THESE FACTS

It costs money to put data into a computer. As a rule of thumb, expect to pay $125 per 1000 records in labor just to gather and key in your data. Ask any data processing professional; in less than two years your data will be worth more than the cost of your computer and software combined. This is without consideration of cost savings or additional income you can generate with the use of this information. You need the best software on the market to look after that asset.

Data Director I	$ 495.00
Data Director II	$ 995.00
Data Director III	$1995.00
Manuals	$ 30.00

Dealer inquiries invited.

Basic Business Solutions, Inc.
831 N. 3rd Ave.
St. Charles, IL 60174
Bus. Offices (312) 377-0813
Support Center (312) 746-8736

STOP

PUTTING UP WITH PROBLEM SOFTWARE . . .

Inefficient routines bogging you down? If so, it's time for a change... to 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 your OSI software. To keep up with the requests for more "good OSI stuff," we've been working overtime finishing several new products, including one we're introducing right here.

With the additions we're making to our product line, we've got a unique problem...how do we fit everything in our ads??!! There's just not enough room to fully describe each item, so we've cut down each description. To get the whole story on each product, just call or write us and we'll promptly mail you a copy of the latest SOFTWARE CONSULTANTS product catalog. It'll give you all the facts on each item you'll want to order.

SOFTWARE CONSULTANTS offers you OSI software that's cost-effective, reliable, and above all, complete. Take a look at this brief list, then ask yourself...Can you really afford not to use SOFTWARE CONSULTANTS as your source for 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 $24.95.

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 $29.95.

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 $79.95.

5. VIDEO ROUTINE
This convenient program really spices up your video system with little niceties like 24 separate control codes, horizontal vertical plotting, and many variable screen parameters. Software extensions are available to connect this with the graphics resolution booster. The routine alone is $24.95, 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 $10, 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, 7053 Rose Trail, Memphis, TN 38134 USA. TN residents add 6% sales tax. Phone us at 901/377-3503 for rush orders.

Dear Software Consultants:
Sounds great. Here's my order. I want:

_____DISASSEMBLY MANUAL(S) @ $24.95 ea
_____CROSS REFERENCE UTILITY(S) @ $29.95 ea.
Disk size 5¼"_____or 8"_____
OS-65D_____or OS-65U_____
_____SPOOLER/DESPOOLER UTILITY(S) @ $59.95 ea.
Std. parallel interface_____
Serial interface w/430 board or equivalent_____
Serial interface w/CA10X board_____
_____FIG FORTH(S) @ $79.95
_____VIDEO ROUTINE(S) @ $24.95 ea.
With software extensions @ 29.95
_____SOURCE CODE @ $10.00 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: