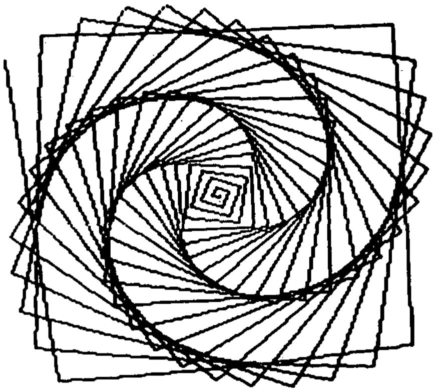
PEEK (65)

The Unofficial OSI Users Journal

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

★★ $1.75 ★★

Editor: Al Peabody
Vol. 2, No. 7, July 1981



See Article page 5

Column One

The dam is bursting. This month's PEEK(65) contains an article by Jim Sanders on the device #5 (parallel) printer driver in OS65U. I am typing this column in WordStar. Rich Comeau of OSI has started leaving messages on the CBBS describing the new things coming down the line from the factory, including a real-time clock, standard, on each floppy disk system, a new high-resolution graphics video board, the new 710 board which will run Oasis, carries 5 processor chips including your choice between a Z8000 and 68000, and lots more. With MA/COM in the system, many of the good things we predicted are happening now. Watch PEEK(65) and be glad you had the sense to buy OSI!

A book I am reading describes the "electronic cottage," a new mode of working in which office workers like most of us have become will no longer go downtown to work, but will stay at home and do our work over the phone with terminals and modems. In fact, I already do that much of the time, working on the CBBS from my home office. It is not entirely satisfactory as yet: when I really mess things up and lock up the system, I would need a 40-mile arm to reboot. But those details will be worked out, and within a few years more and more of us will go to the office primarily for office parties.

This month's issue contains something I am very excited about -- our first CP/M product report (a WordStar review). Not only is WordStar an exciting product (though not without its faults); also, and more importantly, the conduit is now apparently open to supply us with CP/M products for review; and we are learning more each day about CP/M and how it works. The CP/M-OSI connection is tremendously important for all of us, and I will be telling you more about it each month. For now, let me just say in this prominent position that Bonita Taylor and Mike Offe of Lifeboat Associates have been extremely helpful to us. Bonita has worked with us in obtaining products to review, Mike in making the few alterations needed to make Lifeboat's CP/M 2.23 work with our particular configuration of equipment. When they advertise "software with full support," they are not kidding.

This issue also contains a couple of letters from old faithful PEEKers which are a bit angry. I worried about printing them, but not too much. I figure if somebody is excited or angry about something, let him tell us about it, within reason. Others will respond next month (there is one of those in here too) and try to set us straight. The main thing is, by keeping the channel open and still relatively unedited, we hope to encourage open interchange of ideas, opinions, even hot tempers. It is bound to do the OSI community good.

Just who is this famous "OSI community," anyway? This is a question which intrigues me for more than professional reasons. How many of you are there, what kinds of gear do you have, what do you use it for, what would you most like to see from us and from the factory? We will try to find out in the next few months, and will of course let you know what we learn.

Al

DEVICE 5 PRINTER DRIVER

by Jim Sanders
2338 Riviera Drive
Vienna, VA 22180

I have an MX-80 printer with a parallel interface. This low cost printer is a fine machine for just about everything you might want to do, but it has the slowest form feed around. You can also get it to print in letter quality type by moving slightly and reprinting the same line. The listing was printed with this feature on an MX-80. The time it requires is greater than the time-out in OSU for the 'PRINTER 5 STALLED' message, however, and it will drive you out of your mind. Therefore, I decided to change the drivers for the parallel printer. I was aware that OSU was full of patches and code that was of little use, but was not prepared for the tragedy I found during the disassembly. This article presents enough information for a self-study course in assembler programming for the 6502, but please use something else for a style guide.

*** BUGS REVEALED ***

When the printer routine is called, the character to be printed has been stored in CHAR (38B6), and the X register contains the count of the characters already printed on this line. If the printer is not ready, it is necessary to retain these two bytes while sending the 'stalled' message. Notice from the description of the code at 3E25 that the X register is saved in a temporary byte I call X2. Notice also that subroutine CONOUT at 3AE2 is used to display this message to the console, and that CONOUT uses X2. The result is that when the printer is placed online and a key is pressed, the line printed will be shortened by the length of the 'stalled' message. To fix this, use CHANGE to modify the system as follows:

```
RUN"CHANGE"//MODEH//OFFSET
C00
ADDRESS ? 3E2A
00003E2A  B9  ? BB
00003E2B  38  ? .
ADDRESS ?  3E40
00003E40  B9  ? BB
00003E41  38  ? X
CLOSE
```

Copyright ©1981 by DBMS, Inc. All Rights Reserved.

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

** NEW MESSAGE INSTALLATION **

If you are getting tired of the message and would like to change it, here is what you need to know... The text is stored beginning at address 3E49 (15945). Any text you like that will fit may be entered. The last entry must be a zero to end. The standard is 0D (return), 0A (linefeed), PRINTER 5 STALLED, 07 (bell), 00 (end). Play with it! Here's how to use the monitor to evaluate modifications:

Boot OSU. Put the parallel printer offline. Enter the following program:

```
10 PRINT#5,"NOW IS THE TIME..."
```

RUN this program and observe the message after a few seconds. Put the printer online and press any key. Take the printer offline again. Reset the machine and type M to enter the monitor. Type P3E499. (The second nine is used only to stop the printout.) Verify the message is as shown above. Now type L3E49 4F 66 66 4C 69 6E 65 0D 00 R (Which will change the message.) Type G to return to Basic. You must type CLEAR to reset the trash generated by resetting. Now RUN the program we left and notice you have made a new message.

*** OMIT THE KEYBOARD WAIT ***

I like for the system to sit quietly and proceed when I get the printer ready. There doesn't seem to be any good reason for having to press a key, especially with the MX-80. You can omit the keyboard lockup subroutine by changing loation 3E32 from 09 to 0C (POKE 15922,12). This does not eliminate the delay and the message, however. You can do that by changing location 3E18 from 0C to EE (POKE 15896,238) which is how I run my system.

OK, BUT WHAT'S IT GOOD FOR?

A few strange and wonderful surprises showed up after an analysis of the disassembly. First, there is a line counter for the console that operates just like the one for the device 5 printer. It keeps count of linefeeds and may be poked to prevent printing over the perforations on your glass. Try this little test... POKE 15141,24 (actual size of screen, corresponds to 14387). POKE 15100,20 (lines to display before ejecting, corresponds to 14457). Enter in the immediate mode FOR I=1 TO 200: PRINT"LINE";I: NEXT I <cr>. This not so handy feature (unless your console is a teletype) is brought to you by subroutine CONOUT at 3AE2 (see listings).

There are two unusual null generators for the console (ONLY) at 3B02 (also in CONOUT). One produces nulls after linefeeds, the other after 'return' characters. These operate exactly the opposite of the desired way. A teletype device needs a delay to allow the carriage to get back for the next line, and the longer the line, the more delay required. This routine adds one to the counters for each character displayed CNULL(0) and (1), then sends the difference, if positive, between the specified number of nulls TNULL(0) and (1) and the length of the line printed. The result is that the shorter the line, the more nulls that are sent. Do you agree that there is a slight coding error?

*** LET EVERYONE PLAY ***

Be cautious when modifying the code you find in OSU. Those wonderful folks that brought you this patch-box enter routines at any old place, wherever a bit of code can be used by another routine far away. They also are guilty of "Programmer's Sin #6: using instructions for data. For example, Subroutine GETKEY at 3A86 (not shown) needed a hex 94. The nearest 94 was found at 3C8C in another routine as the offset of a relative branch instruction during the test for a 0D (BEQ $94). So in GETKEY we find LDA 3C8C. Who knows if a byte of the code you are trying to clean up is used by another routine? And you paid money for this?

CONTINUED

Software Packages

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

ask us about **BUS-II**
*DMS Compatible

Affiliated Dealers

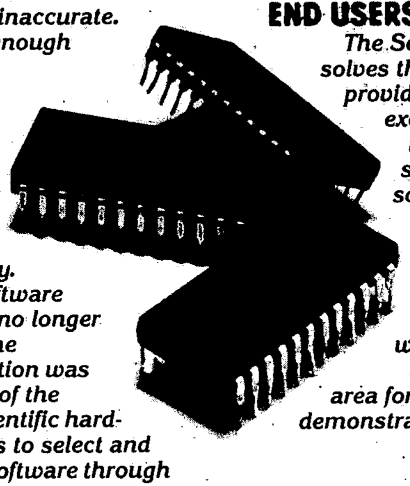
Business Computer of Joliet	815/797-6500
Business Data Systems, Inc.	303/444-3652
Computer Management Systems	605/996-1300
Computer Systems Co.	713/972-1983
Data Buss	312/680-1133
Data Services Computer Corp.	303/371-4300
Computers Plus, Inc.	414/321-1770
Frisch Computer	612/770-5600
Custom Computer 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	303/561-8269
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
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

*** TRY THIS FOR FUN ***

Both the system and Basic call (3A64) DCTEST (see listing) to get input. I have not figured out if the routine at 3A6F is ever used, but it doesn't look like it. So, you can make an "Input With Timeout" routine. First, blow away the interrupt on test with NOP's and change the default character from a space to a 'return'. This will give you 5 seconds at 1 Mhz between characters typed. Any slower typing will insert an automatic return. I put this in a batch program where optional input is allowed. If no key is pressed in five seconds, the return causes a nullstring and the program takes the default path. Yes, you can do that with peeks to the ACIA, but this is another (easier) way. Try the following program for OSU.

```
1 POKE 14962,234: REM NOP was
  80
2 POKE 14963,234: REM NOP was
  245
3 POKE 14975,13: REM <CR> was
  32
4 POKE 14965,100: REM DELAY
  was 50
5 POKE 2888,0:  REM PERMIT
  NULL$
6 INPUT"Any messages";Y$
7 IF LEN(Y$) THEN PRINT
  Y$:GOTO 6
8 PRINT"NO MESSAGE RECEIVED"
9 (poke everything back to
  normal)
```

```
2D5A CA         DEX
2D5B 38         SEC
2D5C 60         RTS

3B17 48         PHA
3B18 A902       LDA #$02
3B1A 2C00FC     BIT $FC00
3B1D F0FB       BEQ $3B1A
3B1F 68         PLA
3B20 8D01FC     STA $FC01
3B23 60         RTS

3832 A942       LDA #$42
3834 8D243E     STA $3E24
3837 D03F       BNE $3878
3839 BD01CF     LDA $CF01,X
383C 297F       AND #$7F
383E 60         RTS

3871 D00C       BNE $387F
3873 CE243E     DEC $3E24
3876 F0BA       BEQ $3832
3878 A93C       LDA #$3C
387A CD243E     CMP $3E24
387D 3001       BMI $3880
387F 60         RTS
3880 4C023E     JMP $3E02

3A64 A9C8       LDA #$C8
3A66 8DB738     STA $38B7
```

```
3A69 AD00FC     LDA $FC00
3A6C 4A         LSR A
3A6D B013       BCS $3A82
3A6F 2CB838     BIT $38B8
3A72 50F5       BVC $3A69
3A74 A932       LDA #$32
3A76 20B83E     JSR $3EB8
3A79 CEB738     DEC $38B7
3A7C D0EB       BNE $3A69
3A7E A920       LDA #$20
3A80 D003       BNE $3A85
3A82 AD01FC     LDA $FC01
3A85 60         RTS

3AE2 8EB938     STX $38B9
3AE5 ADB638     LDA $38B6
3AE8 201738     JSR $3817
3AEB A200       LDX #$00
3AED C90D       CMP #$0D
3AEF F011       BEQ $3B02
3AF1 E8         INX
3AF2 C90A       CMP #$0A
3AF4 D024       BNE $3B1A
3AF6 CE1638     DEC $3816
3AF9 F029       BEQ $3B24
3AFB A942       LDA #$42
3AFD CD1638     CMP $3816
3B00 30E3       BMI $3AE5
3B02 BDFC3E     LDA $3EFC,X
3B05 FD124D     SBC $4D12,X
3B08 48         PHA
3B09 A9FF       LDA #$FF
3B0B 9DFC3E     STA $3EFC,X
3B0E 68         PLA
3B0F B009       BCS $3B1A
3B11 AA         TAX
3B12 A900       LDA #$00
3B14 201738     JSR $3817
3B17 E8         INX
3B18 D0FA       BNE $3B14
3B1A EEFC3E     INC $3EFC
3B1D EEFD3E     INC $3EFD
3B20 AEB938     LDX $38B9
3B23 60         RTS

3B24 A942       LDA #$42
3B26 8D1638     STA $3816
3B29 D0D0       BNE $3AFB

3B2B 2C20F4     BIT $F420
3B2E C90A       CMP #$0A
3B30 4C7138     JMP $3871

3E02 A900       LDA #$00
3E04 8DB738     STA $38B7
3E07 AD00F4     LDA $F400
3E0A 4A         LSR A
3E0B 900C       BCC $3E19
3E0D A90C       LDA #$0C
3E0F 20B83E     JSR $3EB8
3E12 CEB738     DEC $38B7
3E15 D0F0       BNE $3E07
3E17 F00C       BEQ $3E25

3E19 ADB638     LDA $38B6
3E1C 297F       AND #$7F
3E1E 8D02F4     STA $F402
3E21 4C2B3B     JMP $3B2B

3E25 ADB638     LDA $38B6
3E28 48         PHA
3E29 8EB938     STX $38B9
3E2C A200       LDX #$00
3E2E BD493E     LDA $3E49,X
3E31 F009       BEQ $3E3C
3E33 8DB638     STA $38B6
3E36 20E23A     JSR $3AE2
3E39 E8         INX
```

```
3E3A D0F2       BNE $3E2E
3E3C 20643A     JSR $3A64
3E3F AEB938     LDX $38B9
3E42 68         PLA
3E43 8DB638     STA $38B6
3E46 4C023E     JMP $3E02

3EB8 8EBB38     STX $38BB
3EBB A238       LDX #$38
3EBD 205A2D     JSR $2D5A
3EC0 D0FB       BNE $3EBD
3EC2 E901       SBC #$01
3EC4 D0F5       BNE $3EBB
3EC6 AEBB38     LDX $38BB
3EC9 60         RTS

3E49 0D              DATA
3E4A 0A              DATA
3E4B 50     P        DATA
```

```
3E4C 52     R       DATA
3E4D 49     I       DATA
3E4E 4E     N       DATA
3E4F 54     T       DATA
3E50 45     E       DATA
3E51 52     R       DATA
3E52 20             DATA
3E53 35     5       DATA
3E54 20             DATA
3E55 53     S       DATA
3E56 54     T       DATA
3E57 41     A       DATA
3E58 4C     L       DATA
3E59 4C     L       DATA
3E5A 45     E       DATA
3E5B 44     D       DATA
3E5C 07             DATA
3E5D 00             DATA
```

```
(3E02)  Let COUNT=0                ,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
(3E07)  If printer  IS  ready  then  (3E19) put CHAR to acia at F402        ¦
        else:                      ¦ (3B2B) toggle acia at F420             ¦
(3E0F)  call DELAY with 0C         ¦ (3B71) if CHAR not linefeed : Return   ¦
        decrement COUNT            ¦     else:                              ¦
        if COUNT not 0 goto 3E07   ¦     decrement LINES                    ¦
        else:                      ¦     if LINES=0 then: LINES=PAGESIZE    ¦
(3E25)  push CHAR to stack         ¦     if LINES<=PAGE then: return        ¦
        save X reg. in X2          ¦     else:                              ¦
        let X=0                    ¦     goto 3E02 (more linefeeds)         ¦
(3E2E)  Fetch MESSAGE (X)          ,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
        (get "stalled" report)     ,,,,,,,,,,,,,,,,,,,,,,,,,,
        If MESSAGE (X)  NOT  0 then  Let CHAR=MESSAGE (X)    ¦
        else:                      ¦ (3E36) Call CONOUT      ¦
(3E3C)  Call DCTEST                ¦        increment X reg. ¦
        Restore X reg. from X2     ¦        goto 3E2E        ¦
        Restore CHAR from stack    ,,,,,,,,,,,,,,,,,,,,,,,,,,
        goto 3E02

,,,,,,,,,,,,,,,
¦ SUBROUTINES ¦
,,,,,,,,,,,,,,,

(3EB8) DELAY                       ¦  (3A64) DCTEST
        Save X reg. in X1          ¦          Let COUNT=C8
(3EBB)  Let X=38                   ¦  (3A69) If any key pressed
        Decrement X reg.           ¦          then: Load A=key
(3EBD)  If X not 0 goto 3EBD       ¦                  Return
        Decrement A                ¦          else:
        If A not 0 goto 3EBB       ¦  (3A6F) If interrupt on (?)
        Restore X reg. from X1     ¦          then: Call DELAY with 32
        Return                     ¦                Decrement COUNT
----------  ----------  --------¦                If COUNT=0
(3B17) COUT                        ¦                then: Load A='space'
        Push A to stack            ¦                      Return
        Wait for console ready     ¦          else:
        Recover A from stack       ¦             Goto 3A69
        Send A to console acia     ¦
        Return                     ¦
----------  ----------  ----------  ----------  ----------  ----------
(3AE2) CONOUT                                ¦ (3B02) Let A=CNULL(X)-TNULL(X)
        Save X reg. in X2                    ¦        Push A to stack
(3AE5)  Fetch CHAR                           ¦        Let CNULL(X)=-1
        Call COUT                            ¦        Pop A from stack
        Let X=0                              ¦        If A <= 0 then:
        If CHAR IS 'return' goto 3B02        ¦           (3B11) Let X=A
        Let X=1                              ¦           (3B14) Call COUT (null)
        If CHAR NOT 'linefeed' goto 3B1A     ¦                  Increment X
(3AF6)  Decrement C-LINES                    ¦                  If X<0 goto 3B14
        If CLINES=0 then CLINES=SCREENSIZE   ¦        else:
        If CLINES <= SCREEN goto 3B02        ¦ (3B1A) Increment CNULL(0)
        else:                                ¦        Increment CNULL(1)
            Goto 3AE5  (more linefeeds)      ¦        Restore X reg. from X2
                                             ¦        Return
-----------  ----------  ----------  ----------  ----------  ----------  ----
  (38B7) COUNT       (38B6) CHAR      (3E24) LINES (line count for printer)
  (38BB) X1          (38B9) X2        (3879) PAGE  (desired lines to print)
  (4D12) TNULL(0)    (4D13) TNULL(1)  (3833) PAGESIZE (actual size of paper)
  (3EFC) CNULL(0)    (3EFD) CNULL(1)  (3816) CLINES (line count for screen)
  (3AFC) SCREEN (desired lines console) (3B25) SCREENSIZE (size of glass)
----------  ----------  ----------  ----------  ----------  ----------
```

★

HIGH RESOLUTION GRAPHICS FOR OSI COMPUTERS

by E.D. Morris
3200 Washington
Midland, MI 48640

OSIers have long had to take a back seat in the area of high resolution graphics. About the only option available was to buy an Apple and forsake OSI entirely. A few hardware hackers have pieced together custom boards to produce high resolution graphics. This photograph demonstrates what can be done with a custom 256 x 256 graphics board. However, recently the dam has burst. There are now several options for graphics open to OSI owners.

OSI is now listing in a current sales brochure a "256 x 512 dot accessible high resolution graphics board". So far no information on price or availability.

Several other vendors are making OSI bus compatible boards. GRAFIX (911 Columbia Ave N., Bergen NJ. 07047) is selling a high resolution color graphics board. The board uses a CRT controller and thus has many modes of operation. An alphabetic mode is also included. This board generates a video signal entirely independent of OSI's video. You need either a second monitor or a switch to display one or the other. The output can be set for RF or direct video. Available bare board, kit or assembled in superboard or C2 versions. High resolution graphics requires 6K of 2114 memory. The superboard version has 16K of additional memory for program storage. The C2 version has a floppy controller.

Mittendorf Engineering (905 Villa Neuva Dr., Litchfield Park, AZ 85340) sells a board which is only 256 x 256 high resolution. The board derives its timing from your existing OSI video. The output is combined with your present video to be displayed all on one monitor. Eight K of 2114 memory is required. This memory is added onto your present memory and can be used for program storage when not using the High resolution output. Bare board or kit available for superboard or 540 video.

Another source for OSI compatible boards is D & N Micro Products (3684 N. Wells St., Fort Wayne, IN 46808). D & N does not have a graphics board, but sells boards for memory, I/O, floppy disk controller and prototyping. Available bare or assembled. Boards for 48 pin bus only.

BASIC PLUS
(O.S.I. C-1-P)

by Patrick Townsend
P.O. Box 1003
Chicago, IL 60626

BASIC PLUS is a machine code routine designed as an addendum to the Basic loaded into pages zero, one and two upon cold start. It needs to be loaded only one time, upon power up, and will remain until power is turned off. Subsequent cold and warm starts need only have the vectors re-pointed.

In the past, a variety of utility routines have been devised to help the C-1-P programmer. Most of them require calling by USR(X), and the locations for the routines were not standardized to allow all of them to run at one time. The program below does this. Once loaded, any of the special routines for screen clear, list, port audit, load and save can be accomplished with one button control. USR(X) is freed up to do other jobs. It need not be typed in. Pokes 11, 12 need not be pointed at any special location (except as you wish for your own jobs elsewhere).

The program below erases itself and leaves a poked up machine code in page two and in a small portion of page zero. The results are:

Control L to load from tape. (Exit via space bar, as usual)

Control Z to Save to tape. (Exit via Control L and space bar, as usual)

Control R to Run program. (Exit as usual -- via Control C or normal termination)

Escape Key to exit from Basic and audit activity at the ACIA port. (To return to Basic, press Space Bar one time)

Rub Out Key For instant clearing of screen (or scroll window).

Line Feed To List program, and while listing:

Control S to temporarily stop the listing flow; Control Q to resume the listing flow.

(You may alternate back and forth between S/Q as desired). However, for selective line listing (e.g. "List 100") you must still use the word "List" followed by specifics, since Line Feed in effect calls for List and gives CR at the same time, without allowing specifics to be requested. Thus, the S/Q feature.

After a warm or cold start: You must re-position the Vectors, as follows: AS ONE LINE IN IMMEDIATE MODE: POKE 536,78:POKE 537,2; POKE 538, 48: POKE539,2

Don't forget that on some older model C-1-P machines, the first command entered after Warm Start gives an OM error. So to avoid typing the above line twice, first enter some dummy command, like Print P.

Although most machine codes offering utility functions seem to be located beginning at Hex 0222 (Decimal 546), my own particular machine has the Aardvark Edit Rom located (in part) at those locations. So I have devised the program below to begin at 560 Decimal -- where Aardvark's ROM leaves off. However, the user can place the routine wherever desired -- even in the high portion of RAM. But the most logical place is page two (and part of page zero) since these areas are rarely used otherwise, and don't waste any usable memory (since the Basic program below washes itself after running.)

If the user wants to re-locate the program (for example to begin at Hex 0222) care should be taken to change all the JSR's, JMP's, etc. to relate everything to each other. In particular, lines 150 - 160 will need attention. Also, lines 6 and 170 will have to be carefully rearranged. Caution should be observed in moving the routine to other locations, since I have carefully woven them together relative to the 560 - 660 area of page two. Many of these routines were available elsewhere, quite independent of each other. The JSR's in particular just might cause a lot of trouble if you break up (or move) the program.

For ease in loading, you might want to have the program self start after the tape has been run. While still in the Save mode (when first writing the enclosed), after line 200 has passed, type in the immediate mode, (Carriage Return) (Carriage return) Poke 515,0:Run (carriage return). The above added to the tape will cause the machine to execute and erase the program as soon as you have loaded it. If all goes well, you should then be able to press "Rub Out" and get an immediate screen clear.

The so called Port Auditing Routine (Escape Key) allows the programmer to review the contents of a tape (or any activity at the Back Door) while another program is operating, without loading the activity at the port inadvertently onto the current program. To use it, start running the tape to be examined, then hit "Escape". The characters on the tape will fill up your screen, until the space bar is hit.

If you don't like the use of L/Load; Z/Save and R/Run, you can change them as follows: (after program is run). From A to Z is from one to twenty-six. To change Load: Poke 594, xx - Save:Poke 601, xx - Run: Poke 622, xx. "xx" is your choice of letter to be pressed with the control key.

Normal values of these locations, of course, is 594(12); 601 (26) ; 622 (18). Line Feed value is 10, which means it can also be worked by Control J.

BASIC PLUS

A machine code routine which provides additional control code features to users of O.S.I. C-1-P Basic in Rom machines. By Patrick Townson, 5/81.

PURPOSE: When operative, provides user with Control L to load; Control Z to Save; Control R to Run; (Rubout) for instant screen clear; (Escape) to audit port without loading or disturbing current run.

CAUTION: Basic does not load over this, so it does not have to be re-run after a cold start. But after a cold or warm start, you must reset vectors: Poke 536,78:Poke 537,2:Poke 538,48:Poke 539,2

Line # 16176 TO 16276

```
 6 For X = 560 to 660
 10 Read A : Poke X,A: Next
 14 REM 560-589 is Control
    Output S/Q
 15 Data 72, 169, 246, 141, 0,
    223, 169, 192, 44, 0
 20 Data 223,208,12, 169, 252,
    141, 0, 223, 169, 192
 30 Data 44, 0, 223, 208, 244,
    104, 76, 105, 255, 0 : REM
    589 REM S/Q listings
 70 Data 32, 186, 255, 201,
    12, 208, 3,32,139,255,201
    :  REM 12 is L/Load
 80 Data 26, 208, 3, 32, 150,
    255, 201, 127, 208, 3, 76:
    REM 26 is Z/Save          63
 90 Data 127,2,201,10, 208, 3,
    76, 181, 164 : REM 620REM
    10 is Line Feed/List
 95 Data 201, 27, 208, 3, 76,
    216, 0  REM  27 is ESC
    Port Audit
100 Data 201, 18, 208, 6,32,
    119, 164, 32, 194, 165,
    96: REM 18 is R/Run
110 REM Lines 120 - 130 for
    Rubout Key Screen Clear
    routine
120 Data 72, 169, 32, 162, 0,
    157, 0, 208, 157, 0, 209,
    157, 0, 210
130 Data 157, 0, 211, 232,
    208, 241, 104, 96
140 REM Lines 150 - 160 Set
    Input and Output vectors.
150 Poke 11, 78: Poke 12, ~~2~~ : 63
    ~~Poke 536, 78: Poke 537, 2~~
160 Poke 538, 48 : Poke 539, 2
165 REM Pokes 216-235 is the
    Port Auditing Routine
170 For X equals 216 to 235:
    Read Y: Poke X, Y : Next
180 Data 169, 255, 141, 3,2,
    44, 3,2,16, 9, 32,186,
    255,32,45,191,24
190 Data 144,242,96
200 New   : REM No usable
```

memory lost. This erases itself and leaves a poked up machine code in pages 2 (560-660) and a little in page 0 (216-235).

To self start, while saving to tape, add "CR/CR poke 515,0: Run CR" * * * * *

160 POKE 133, ~~30~~ 40 : POKE 134, 63

E-X-T-E-N-D your BASIC!!

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

Extend BASIC (in ROM)? That's impossible!! Unless you have BASIC in RAM or you have an input/output JMP vector in RAM (the C1P has its JMP vector in RAM, but the C4P does not, I do not know why.)

I have been under this illusion that I could not get any useful extra routines such as CALL, GET, TIME, SET, RESET, etc. I think many of you thought the same.

But Peek (65) is the greatest magazine!! The article by Mr. Hooper (Vol. 1, #12) encouraged me to study OSI BASIC in ROM very hard.

After an extensive study of the ROM, I found it impossible to add any extra commands except by modifying USR(X), or reprogramming the ROM.

Since I do not have an EPROM programmer, what I can do is to modify the USR(X) routine. (With Mr. E. Morris's great information, I could get ROM#3 and a character generator ROM replaced.)

Have you read my screen clear routine (CLS2.2)? It was about half way to doing the job. Now perhaps you can imagine how I can extend my BASIC in ROM.

So I call this "EXTENDED USR". You can have as many extra routines as you have utility programs. The format is:

USR(CALL),A,B,C,D,E,F,G,H

This is a CALL command in the decimal mode; CALL can be a variable in BASIC or a number in 2's complement which points to your machine language subroutine start address. Also, you can pass BASIC variables or numerals (-32768 to +32767) into your machine language routine. Those are represented as A,B,C,D,E,F,G,H in the above example. The program shown allows 8 variables, but you can modify it very easily. The Hex version is:

USR(xx)!XXXX,A,B,C,D ....

XXXX :Subroutine address in hex.

xx :Can be any alphanumeric character (dummy)

?USR(xx)D,A

```
     ; ======== EXTENDED USR(X) =========
     ;
     ;          by YASUO MORISHITA (MAY 1981)
     ;
     ;?USR(CALL),A,B,C,D,E,F,----- CALL subroutine defined by
     ;          numbers (in decimal); uses BASIC
     ;          variables A,B,C,D,E,F (up to 8 variables).
     ;?USR(O)!ABCD,E,F----------- CALL subroutine defined by
     ;          hex value $ABCD; can pass BASIC variables
     ;          E,F.(up to 8 16-bit variables)
     ;?USR(O)HABCD or A=USR(O)HABCD--- Convert
     ;          $ABCD into 2's complement decimal.
     ;?USR(O)D,A ---------------- Convert decimal value
     ;          defined by numeral A or variable A into hex
     ;          and display it on the screen as XXXX.
     ;
     ;
     ; --- Data registers in page 0 and 2.---
     LPTR1 =$OOD8 ; Address pointer (L)
     HPTR1 =$OOD9 ;    "       "    (H)
     CTRO  =$OODA ; Data counter
     LPTR2 =$OODB ; Used for real time clock display (L)
     HPTR2 =$OODC ;   "     "          "            (H)
     DTABF =$O2EO ; 8 16-bit data registers (L,H,L,H,......)
                  ;  ($O2EO - $O2EF)
     ; --- Subroutines from BASIC in ROM ---
     NXTCHR=$OOBC ; Get next character from BASIC line.
     GETCHR=$OOC2 ; Get current character from BASIC line.
     GETVAR=$AEO5 ; Get variable from BASIC.
     TOHEX =$FE93 ; Convert ASCII to O - 15.
     ROLNBL=$FEDA ; Roll nibble into $FC+(X)--for OSI SYNMON
                  ;                  $F9+(X)--for C2E CEGMON
     CHKCMA=$ACO1 ; Check "," and get next character from
                  ;   line, else syntax error printed.
     EVLATE=$AAAD ; Evaluate BASIC expression
     BERROR=$AE85 ; B S ERROR, data too many (> 8 pairs)
     INKEY =$FDOO ; Receive character from keyboard.
     OUTVAR=$OOO8 ; or =$AFC1, pass value into BASIC.
1FOO * = $1F00    ; Main program start address
     ; 52 3F
1FOO 20 ~~6A~~ 1F  USR(X) JSR BFCLR     ;Clear data buffers
     20 O5 AE          JSR GETVAR    ;Get variable from BASIC
     A5 AE             LDA $AE       ;Get high byte (FACHI)
     A4 AF             LDY $AF       ;Get low byte (FACLO)
     85 D9             STA HPTR1     ;Set address pointer (H)
     84 D8             STY LPTR1     ;       "            (L)
     20 C2 OO          JSR GETCHR    ;Get character from line
1F11 48                PHA           ;Save it in stack
     C9 21             CMP #$21      ;USR(xx)! ?
     FO O4             BEQ NXT1
     C9 48             CMP #$48      ;USR(XX)H ?
     DO O3 3F          BNE NXT2      ;else check more
     20 5F 1F   NXT1   JSR HEXDML    ;Convert hex address into
                                     ; decimal value and store
                                     ; at  $D8 (L), $D9 (H).
     68         NXT2   PLA
     C9 48             CMP #$48      ;USR(xx)H ?
1F2O DO O7             BNE CHKDT1    ;else check more
     A5 D9             LDA HPTR1     ;Prepare to pass value
     A4 D8             LDY LPTR1     ;into BASIC
1F26 6C O8 OO          JMP (OUTVAR)  ;Output value into BASIC
1F29 48         CHKDT1 PHA
     C9 44             CMP #$44      ;USR(xx)D ?
     DO O3             BNE NXTDTA    ;else check more
     20 BC OO          JSR NXTCHR    ;Get next character
1F31 20 76 1F   NXTDTA JSR CHKDTA    ;Check data and store it
     68    3F          PLA
     C9 44             CMP #$44      ;USR(xx)D ?
     DO 16             BNE OUTUSR    ;else,execute USR CALL
     A2 C5      HEXDSP LDX #$C5      ;DEC-HEX conversion.
     AO D7             LDY #$D7      ; Display hex at $D7C5,X
     86 D8             STX LPTR1     ; Set up display address
     84 D9             STY HPTR1     ; pointers.
1F41 A2 FF             LDX #$FF
     AO O3             LDY #$O3
     E8         NXTBYT INX
     BD EO O2          LDA DTABF,X   ;Get decimal data
     20 9F ~~1F~~3F    JSR DMLHEX    ;Convert decimal into hex
     10 F7 ~~3F~~      BPL NXTBYT    ; and display on screen
     60                RTS
     ;
```

This is decimal to hex conversion which prints the result in hex on the screen (lower left corner). It can be used as a calculator. This is very useful when you deal with machine language programs.

USR(xx)HXXXX

This is a hex to decimal conversion. XXXX should be a 16-bit hex number. If you write U=USR(xx)H1F00 in your BASIC program, this will convert $1F00 to "7936" and the BASIC variable "U" will be set to 7936 decimal. This statement is used to set up a CALL address such as; CLS=USR(0)H1FD8 in the BASIC initialization program. Now you are free from tedious hexdecimal conversion!!

Only 4 utilities? Yes!! But USR(CALL) and USR(xx)!XXXX will extend your BASIC infinitely!

You can assign your routine a name in your BASIC and then use it. U=USR(CLS) will clear your screen, U=USR(CCL), 8 will change your screen color to blue if you have a C4P. KY=USR(GET) will give you the ASCII value of the next key struck. This may help you to work on a word-processing program. LETU=USR(SYSTEM) will jump to the monitor without hitting the BREAK key.

Please note that when you assign your routine, name, you should avoid using the same variables in your own BASIC program, or your routine will be messed up!

If you do edit your program or push BREAK key or do CLEAR, you have to re-initialize (RUN44444 or RUN) to set up the command names. In the immediate mode, you can type ? USR(CLS) :this will print a meaningless number on the screen, so you may be better off to type LET U=USR(CLS).

In a program, try 100 U=USR(CLS). Using an assigned name such as SYSTEM is surely easier than memorizing $FE0C (Monitor warm start).

How do you proceed? First, you have to write these machine language routines into memory. You can do this using the monitor or you can convert them to decimal, put them in a DATA statement and poke them into RAM. In my case I have a C2E CEGMON new monitor ROM, so I saved the machine language program in auto start format on tape along with my BASIC

```
1F4F 6C D8 00    OUTUSR JMP(LPTR1)     ;Execute USR CALL
     ;
     ; USR(xx) Subroutines
     ;
1F52 A0 0F       BFCLR  LDY #$0F       ;Number of buffer - 1
     A9 00              LDA #$00       ;Clear to #$00
     99 E0 02    LPBF   STA DTABF,Y    ;Clar buffers
     88                 DEY
     10 FA              BPL LPBF       ;repeat 16 times
     85 DA              STA CTR0       ;Reset data counter to 0
     60                 RTS
     ;
1F5F A2 DC       HEXDML LDX #$DC       ;Offset from $FC to $D8.
                                       ;If CEGMON do LDX #$DF
     20 BC 00    HEXDM1 JSR NXTCHR     ;Get next character
     20 93 FE           JSR TOHEX      ;Convert ASCII to 0-15
     30 0C              BMI RTN        ;Illegal hex, do exit
     20 DA FE           JSR ROLNBL     ;Roll nibbles into $D8,D9
     F0 F3              BEQ HEXDM1     ;(=JMP)
     ;
1F6E 20 C2 00    CHKEND JSR GETCHR     ;Get character from line
     F0 02              BEQ RTN        ;End of line/input.
     C9 3A              CMP #$3A       ;(:),end of block
     60          RTN    RTS
     ;
1F76 A0 00       CHKDTA LDY #$00       ;Reset data counter to 0.
     84 DA       NXTDTX STY CTR0
     20 6E ~~1F~~ 3F     JSR CHKEND     ;Check if line/block end.
     F0 F6              BEQ RTN        ;Yes, it is.
     20 01 AC           JSR CHKCMA     ;Check (,),else SN-error
1F82 20 AD AA           JSR EVLATE     ;Evaluate expression
     20 05 AE           JSR GETVAR     ;Get variable from BASIC
     A4 DA              LDY CTR0
     C0 10              CPY #$10       ;Data too many?
     30 03              BMI DTASTR     ;OK, store data in buffer
1F8E 4C 85 AE           JMP BERROR     ;Else BS -error
1F91 A5 AF      DTASTR  LDA $AF        ;Get low byte (FACLO)
     99 E0 02           STA DTABF,Y    ;Store it in data buffer
     C8                 INY
     A5 AE              LDA $AE        ;Get high byte(FACHI)
     99 E0 02           STA DTABF,Y    ;Store high byte
     C8                 INY
     D0 D9              BNE NXTDTC     ;(=JMP)
     ;
1F9F 48          DMLHEX PHA
     20 B5 ~~1F~~ 3F            JSR GETNBL     ;Convert low nibble
     20 AA ~~1F~~ 3F            JSR DSPLY0     ;Display on screen
     68                 PLA
     20 B1 ~~1F~~ 3F            JSR HINIBL     ;Convert high nibble
1FAA 91 D8      DSPLY0  STA(LPTR1),Y   ;Display on screen
     2C 91 DB           BIT $DB91      ;Trick to hide in BIT
                                       ;DSPLYI STA (LPTR2),Y
                                       ;for real time display
     88                 DEY
     60                 RTS
     ;
1FB1 4A          HINIBL LSR            ;Divide data by 16
     4A                 LSR
     4A                 LSR
     4A                 LSR
1FB5 29 0F       GETNBL AND #$0F       ;Erase high nibble
     09 30              ORA #$30       ;Convert to ASCII data
     C9 3A              CMP #$3A       ;Check if data is 0 - 9
     30 03              BMI RTNHNB     ;Yes,data is 0 - 9
     18                 CLC            ;Else convert to
     69 07              ADC #$07       ; A,B,C,D,E,F
     60          RTNHNB RTS            ;Data is in ACC.
     ;
1FC1 20 00 FD    GETKEY JSR INKEY      ;Receive a character from
                                       ; key board.
     A8                 TAY            ;Prepare to send data to
     A9 00              LDA #$00       ; BASIC, CH=USR(GETKEY):
     6C 08 00           JMP (OUTVAR)   ;        CH$=CHR$(CH)
     ;
1FCA 48          GETHEX PHA            ;This Decimal-Hex conver-
     20 B5 ~~1F~~ 3F            JSR GETNBL     ; sion routine is for
     20 AD ~~1F~~ 3F            JSR DSPLYI     ; real time clock display
     68                 PLA            ;If you do not have real
     20 B1 ~~1F~~ 3F            JSR HINIBL     ; time clock(OSI method).
1FD5 4C AD ~~1F~~               JMP DSPLYI     ; delete this routine.
     ;
```

OSI **AARDVARK** OSI

NOW MEANS BUSINESS!

WORD PROCESSING THE EASY WAY— WITH MAXI-PROS

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

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

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

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

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

MAXI-PROS – $39.95

THE EDSON PACK ALL MACHINE CODE GAMES FOR THE 8K C1P

INTERCEPTOR —You man a fast interceptor protecting your cities from Hordes of Yukky Invaders. A pair of automatic cannon help out, but the action speeds up with each incoming wave. It's action, action everywhere. Lots of excitement! $14.95

MONSTER MAZE – An Arcade style action game where you run a maze devouring monsters as you go. If one sees you first, you become lunch meat. Easy enough for the kids to learn, and challenging enough to keep daddy happy. $12.95

COLLIDE – Fast-paced lane-switching excitement as you pick up points avoiding the jam car. If you succeed, we'll add more cars. The assembler code provides fast graphics and smooth action. $9.95

SPECIAL DEAL—THE ENTIRE EDSON PACK— ALL THREE GAMES FOR $29.95

THE AARDVARK JOURNAL

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

1) A tutorial on Machine Code for BASIC programmers.

2) Complete listings of two word processors for BASIC IN ROM machines.

3) Moving the Directory off track 12.

4) Listings for 20 game programs for the OSI.

5) How to write high speed BASIC – and lots more –

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

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

24K and OS65D required, dual disks recommended. Specify system.
Accounts Receivable. $99.95

*** * * SPECIAL DEAL – NO LESS! * * ***

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

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

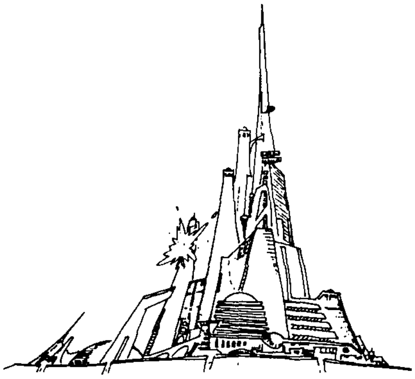
SUPERDISK II

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

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

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

AND FUN, TOO!



BOOKKEEPING THE EASY WAY – WITH BUSINESS I

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

This package is built around a **GENERAL LEDGER** program which records all transactions and which provides monthly, quarterly, annual, and year-to-date PROFIT AND LOSS statements. GENERAL LEDGER also provides for cash account balancing, provides a BALANCE SHEET and has modules for DEPRECIATION and LOAN ACCOUNT computation.
GENERAL LEDGER (and MODULES) $129.95.

PAYROLL is designed to interface with the GENERAL LEDGER. It will handle annual records on 30 employees with as many as 6 deductions per employee.
PAYROLL - $49.95.

INVENTORY is also designed to interface with the general ledger. This one will provide instant information on suppliers, initial cost and current value of your inventory. It also keeps track of the order points and date of last shipment.
INVENTORY - $59.95.

GAMES FOR ALL SYSTEMS

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

MINOS - 8K - – Features amazing 3D graphics. You see a maze from the top, the screen blanks, and when it clears, you are in the maze at ground level finding your way through on foot. Realistic enough to cause claustrophobia. – $12.95

NEW – NEW – NEW

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

TIME TREK - 8K - Real Time and Real graphics Trek. See your torpedoes hit and watch your instruments work in real time. No more unrealistic scrolling displays! – $9.95

SUPPORT ROMS FOR BASIC IN ROM MACHINES – C1S/C2S. This ROM adds line edit functions, software selectable scroll windows, bell support, choice of OSI or standard keyboard routines, two callable screen clears, and software support for 32-64 characters per line video. Has one character command to switch model 2 C1P from 24 to 48 character line. When installed in C2 or C4 (C2S) requires installation of additional chip. C1P requires only a jumper change. – $39.95
C1E/C2E similar to above but with extended machine code monitor. – $59.95

Please specify system on all orders

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

OSI

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

OSI

program. LOAD the machine language program first and then LOAD the BASIC program, then RUN or RUN 44444.

I also have a real time clock in my C4P. This is the CM9-FR board (Floppy controller & 24K RAM & real time clock) from D&N Micro Products, Inc. (Fort Wayne, IN). I wrote a program for it and using this USR(X) I can start the clock:

?USR(TIME),12,34,56

Starts the clock at 12:34:56. I can set up an alarm as ?USR(ALARM),12,40,00.... etc.

My next step to improve this USR(X) would be to burn them in a ROM and have a machine language program which sets up these variable names in the BASIC automatically (I'm not sure if I can do this.)

Following are the programs for USR(X) which contain several useful utilities.

P.S. :To relocate this program you need to change the addresses of subroutines, such as:
BFCLR,HEXDML,VHKDTA,DMLHEX,CHKEND,GETNBL,DSPLY0,HINIBL,DSPLYI,CLS2.3,GETKEY,CCL.

Please enjoy it, and if you come up with better ideas, write to Peek (65).

```
1FD8 AO DO        CLS2.3 LDY #$DO        ;Screen page
     2C AO EO            BIT $EOAO       ;Trick to hide in BIT
                                         ;CCL LDY #$EO ;Color page
     84 D9               STY HPTR1       ;Set address for screen
     AD EO O2            LDA DTABF       ;Get character/color code
     A4 DA               LDY CTR0        ;Check if any data was
     DO O2               BNE CLEAR       ; given or not.
     A9 2O               LDA #$2O        ;If not, it is screen
                                         ; clear. #$2O=space mark
1FE8 AO OO        CLEAR  LDY #$OO
     A2 O8               LDX #$O8        ;Number of page cleared
     84 D8               STY LPTR1       ;Set screen low address
     91 D8        LPC    STA(LPTR1),Y    ;Do clear/print/painting
1FFO C8                  INY             ;
     DO FB               BNE LPC         ; ?USR(CLS);Clear screen
     E6 D9               INC HPTR1       ; ?USR(CCL),C ;Change
     CA                  DEX             ;   color into C
     DO F6               BNE LPC         ; ?USR(CLS),65;Fill
1FF8 6O                  RTS             ;   screen with letter A.
     ;
     ;
```

```
 ======= INITIALIZATION IN BASIC ==========
   1  GOSUB 40000 :REM USR(X) INITIALIZATION

      -------------------------------------
      Your own program, for example;

 100      U=USR(CLS)    :REM Clear screen
 110      U=USR(CCL),4  :REM Change screen color, if color mode.
       ...................................
       ...................................
      -------------------------------------
39999 END : USR(X) INITIALIZATION
40000 POKE 133,O: POKE 134,31:REM Set reserved memory.
40010 POKE 11,0 : POKE 12,31 :REM USR(X) starts at $1F00 65036
40020 SYSTEM = USR(O)HFEOC :REM Jump to monitor, USR(SY) 16344
40030 CLS    = USR(O)H1FD8 :REM Screen clear   , USR(CL) 16347
40040 CCL    = USR(O)H1FDB :REM Change color   , USR(CC),color
40050 GETKEY = USR(0)H1FC1 :REM Get character from keyboard. 16321

    ..............(Initialize your own subroutines here in same)
                  (manner as shown above.                      )
44443 RETURN

44444 GOSUB 40000 : STOP :REM For immediate mode use,
                         :Input RUN 44444 whenever you destroyed
                         :BASIC variable table.
```

THE CP/M CONNECTION

** a WordStar review **

by Al Peabody

* * * * *

Several months ago, we learned that Lifeboat Associates, the leading CP/M software house, had produced a version of CP/M, the most popular operating system for microcomputers, which would not only run on OSI's C3 computer line, but also control OSI's 74 and 36 Mbyte hard disks. At that time, I became determined to review some of the hundreds of software products available through Lifeboat and others to run on CP/M.

Now, at last, after over two months, WordStar, our first disk, has arrived, largely due to the efforts of Bonita Taylor of Lifeboat. I have worked with it for several days, and think I almost understand it. With a couple of reservations, I love it, and with no reservations at all, I am delighted OSI'ers (with C3's) can use it at last.

WordStar will accept input of a continuous stream of words, wrapping around to the next line without carriage returns, and even justifying the right margin right before your eyes. At all times, the "status line" at the top of your screen shows you what document you are working on, what page you are on, what line, and even what column. Yes, as you type along, the cursor instantly bounces up to the top of the screen with every letter, updating the column number. Then when you get to the end of a line, the last word, the one which wouldn't fit, jumps down to the next line, the old line reforms itself to be right justified, and on you type.

As you are entering text, you can input embedded or "dot" commands which WordStar will recognize at printout time to do whatever your printer can do, and probably more. Boldface, subscripts, underline, strikeover, boldfacedunderlinedoverstruck, you name it, WordStar can do it.

After the text has been entered, you can scroll through it, move the cursor around and delete, type over, block move, change your mind, insert without limitation...in short, edit the dickens out of your masterpiece, all with ease.

One beautiful feature is called "hyphen help." Suppose you are entering text which will eventually be printed out just 30 columns wide. As you enter the text, any word which will not fit within 30 columns will be wrapped to the next line, sometimes requiring quite a bit of stretching of the previous line. Don't like that? No sweat, just go back to the beginning of any paragraph and type a control-B and the paragraph will be "reformed." This is particularly useful if you have inserted some text in one or more of the lines, but even if not, if WordStar detects that a wordwrap has left too many spaces on one line, it will display a suggested hyphenation point, which you can accept, move or reject with a single keystroke. The amazing thing is, about 4 times out of 5, WordStar picks an acceptable hyphenation point! The algorithm used is certainly more than just counting letters. It recognizes prefixes, double letters and some more stuff I haven't figured out yet.

Has WordStar no faults? Of course it has. My grand-daddy told me "there is no such thing as a free lunch." Everything has its price. First off, WordStar costs quite a bit in cash. Secondly, because it does so much between every letter struck on the keyboard, fast typists will get ahead of it and lose letters. Not too important for clutzes like me who just peck along, but a lawyer I heard of who tried to use WordStar had to hire three typists before he found one who could slow down enough for it! Also, since WordStar will work on any terminal, it can't display things which many terminals can't display, like sub or superscripts, bold type, etc. These things are displayed as control codes which appear on the screen, somewhat annoying, but the only solution I can think of is to get a dedicated word processor like an IBM or NBI which really won't do any more for you and will cost thousands of dollars more!

Conclusion: If you have a C3, are willing to spring for CP/M and need to do heavy original text composition and editing, WordStar beats OSI's WP-2 hands down. It has its faults, fast typists will be frustrated with it, but all in all it is the best system I have tried ... so far.

Next month -- Microspell.

GOODIES for OSI Users!

from

master charge
THE INTERBANK CARD

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

VISA

( )	**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, GOSUB.s & GOTOs, variables by line number, resequences parts of programs or entire programs, handles line 50000 trap. Best debug tool I've seen. MACHINE LANGUAGE - VERY 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 ________

LETTERS

ED:

In response to Arthur Goeres' letter in the April issue, you asked for the fixes to the bug in CREATE and the problem of booting up to BEXEC*. I struggled with and solved the same problems with my C2-8P.

My version of CREATE had the following bug in line 580:

```
580 DISK!"CA 2E79=08,2"
:GOSUB10030
```

This line is supposed to open and search the second 32 directory slots. However, the subroutine call is wrong, and the effect is the same as if all 32 entries were full; hence, the "FULL" message after only 32 program names. To fix it, just substitute 20030 instead of 10030 in line 580.

The BEXEC* boot is a little more complicated. OS65D3's cold start routine sets up the message "RUN BEXEC*" in the KERNEL buffer at 2E1E. The code for this routine is at track 1, sector 1. To change to a non-destructive boot, you must change the code as follows:

```
A*CA 5000=01,1
A*EM
EM V2.0
:@5425
5425/52 0D
:EXIT
A*SA 01,1=5000/5
```

If you examine the code from $5425 to $5430, you'll see that it holds the message RUN"BEXEC* (CR). We have merely deleted that message by inserting the carriage return at the beginning. Of course the code can be called to an address different from 5000, if more convenient, or other messages can be inserted, such as EXIT, or any other BASIC file can be booted providing it has the following BASIC code line in the beginning:

```
X=PEEK(10960):POKE8993,X:POKE8
994,X
```

Now when that newly typed in BASIC program locks up, and in your haste to try it out you have not yet recorded it, do the following:

1) Press reset
2) Go to the monitor
3) Copy the data in addresses 3179 through 3180
4) Reboot your disk

When you boot up you get an error message, but you're in BASIC with any source code in the workspace still intact, except for the end address in the header at 317B and 317C, the number of tracks needed by the program (at 317D) and the address of the second line of the BASIC source code (at 317F and 3180). These are easily fixed, however, by going to the extended monitor, inserting the code which you have recorded before booting and then saving the program to disk. If your program was using a disk buffer, you will also have to fix the start address in 3179 and 317A, since the BASIC boot will reset that to 317F. If you forgot to record the code before booting, you can still recover the end address by using the EM binary search (code N) to find three nulls (00) in a row. The end address is the address of the second null plus 3. This is the start of the variable table. The second line address to be inserted at 317F and 3180 is the address of the code immediately following the next null, which should be within about 70 characters or so from the beginning. Now save the program to disk as stated above, go back to BASIC, call the file and you're in business. I've found the non-destructive boot invaluable in saving time and effort when programs bomb, and sooner or later, they always do.

Assembly language source code can similarly be saved. Note that address 3179 should contain 7E (the BASIC boot sets it to 7F) and that the start and end addresses can also be obtained from addresses 12FE and 12FF (in the assembler). Also, instead of the address of the second line of source code, you must enter at addresses 317E and 317F your first line number in hex.

With respect to Mr. Boyd's questions about his RCA TV set, there are various methods of fixing overscan, but most depend on having a schematic diagram of the chassis. If it's Solid State and uses an adjustable voltage regulator, you merely lower the voltage. This is a simple screwdriver adjustment on my Heath set. Lowering the voltage from the specified 12 volts to 9, shrinks both the horizontal and vertical dimensions. Other methods are to insert a resistor of adequate power rating in series with the vertical yoke leads or to use an adjustable isolation transformer to lower the line voltage to the set (which may not work on a Solid State voltage-regulated chassis). If you're only interested in the vertical sweep, the simplest solution, of course, is to adjust the vertical size control. This assumes you have one and can find it. It might be worth investing a few bucks in a SAM's Photofact folder for your set. Since the problem is in the sweep circuits, direct video amplifier entry (which I am using) does not help.

Finally, Mr. Baum's complaint about only being able to see 24 lines at a time reminded me that some of your readers may not know that the OS65D3 disassembler will disassemble up to 129 lines of code without touching the line feed control. Just poke the hex value (maximum, $80) of the number of lines you desire into address 18DA.

Sidney Sosin
Glenview, IL

Sidney:

Thanks for the excellent info! One more detail..., how, when and from what RAM locations does what code insert the RUN "BEXEC* message in the buffer? Software Consultants' excellent 65D Disassembly manual contains every line of 65D, but nothing about this, so it must be a separate routine-tell us about it!

AL

* * * * *

ED:

Regarding the expansion interface on the 610 Board, I have experienced exasperations similar to those of Michael B. Carroll (Vol.2, No.5). Although I had Sam's Manual close at hand, it took a great deal of inner conflict and turmoil before I checked the pinouts against the normal I/O socket pinouts. The two (J2 and J1 respectively) are very, very different.

Then I discovered the 620 Bus Expander Board and things began to fall into place. The 620 interfaces between the 40-conductor ribbon cable of the 610 Board and the 48-pin molex connectors of the OSI Bus. 610 J2 connects to 620 J1, and 620 J3 connects directly to the Bus. 620 J3, a 16-pin DIP socket, is there to handle the four upper level addresses, A16 to A19.

I really don't know whether it all works though, because I don't have any 48-pin Bus Boards yet. The schematics I received with the 620 were Early Japanese Photocopy, and fail to indentify components beyond their standard reference. All the active components seemed to be in place, but there are at least 50 resistors NOT in place. The 620 Board and an 8-slot motherboard have been sitting around collecting dust for several months now; any reasonable offer would be gladly accepted. Or conversly, I would be most happy to receive any advice on how I could put these two things to some use in my system.

I am struck by the similarities between owning OSI equipment and wandering through an Adventure. I'm not surprised that they call it the "Challenger " line!

John Kula,
Victoria, BC. Canada

* * * * *

ED:

In your summary of the indirect-file operation for transferring BASIC files to and from WP-2 for editing, you asked for response from anyone whose experience was different. Here's one.

We (the Naval Research Lab) are running a C3 with the TVI-12 terminal, under OS-65D. The indirect file operation for 65D differs from 65U only in the POKE locations. These can be omitted anyway since the machine comes up with suitable default values.

Going from WP-2 to BASIC works as described. P CTRL-\ sends the file up in RAM, CTRL-] marks the end, CTRL-X brings it back to BASIC. The trouble comes in going from BASIC to WP-2. LIST[sends the ASCII file up,]] marks the end, and CTRL-X brings it down for WP-2--and brings it down, and on and on forever. The machine reads that indirect file again and again, even indicating the CMD ERROR each time it sees the]], but never quits. I've let it run for ten minutes this way.

A peculiar thing is that this procedure will, on rare occasion, work. But this is so rare that WP-2 is useless as an editor for our BASIC programs. We will welcome any suggestions.

Jack McKay
Washington, D.C.

* * * * *

ED:

I have a C3S1 and I am using a Centronics 737 (Parallel Interface) and I am a very frequent user of DMS software, v. 9/79 doing mailing lists and the like.

As you know, Centronics 737 has 3 type-fonts, the normal 10CPI (default) print, 16.7 CPI Condensed print, and a proportional font. These fonts can be selected by POKEing in the information, which, up to now I've done by going into the immediate mode. This becomes rather cumbersome when printing horizontal reports which are of different line lengths, and I found a very simple solution, which I would like to share with you and your readers.

I made a slight modification to the STATO3 program. Following are the lines as the were originally (before I made the change):

```
450 IF QA$="1" THEN GOSUB
    60010:GOTO 480
460 IF QA$="2" THEN SW=79
    ;GOSUB 60010;GOTO 480
470 GOSUB 61010: PRINT:
    PRINT: GOTO 410
480 PRINT: INPUT"SINGLE
    OR DOUBLE SPACED REPORT
    (S/D) ":SKIP$
etc
```

I made the following modification:

```
450 IF QA$="1" THEN GOSUB
    60010: GOTO 475
460 IF QA$="2" THEN GOSUB
    60010: GOTO 478
470 GOSUB 61010: PRINT:
    PRINT: GOTO 410
475 POKE 11686,17: PRINT
    CHR$(27);CHR$(20)
476 POKE 11686,1
477 GOTO 480
478 POKE 11686,17:PRINT
    CHR$(27):CHR$(19)
479 POKE 11686,1
480 (as above)
```

Now when I select (1) 132 column printer output, the printer uses 16.7 CPI Monospace condensed print and I get true 132 column output on 8-1/2" wide paper. Similarly, when I select (2) 80 column printer output, the printer reverts to 10CPI print on 8-1/2" paper, which is always more legible than the smaller print size. By placing this fix in this program, one has the option to select it here, rather than having to reboot (reset) the system in order to get back to the printer initialization routine which is located in BEXEC*; or, on the other hand, going to (99) immediate mode, and poking in the required poke. The latter has the setback that it's easy to make a typing error while doing this.

F.S. Schaeffer
Jamaica, NY 11435

* * * * *

ED:

I've always got to get my two cents worth in. In the April issue, Jim Zajac wrote about using the OS65D system command "Go hhhh" as a type of "USER" function. This is a very good thing to know, as a little thought will reveal. I have been using this system in file handling to keep the disk head down on the disk, instead of banging up and down with every "get" and/or "put" by calling the head drop/raise routines, then blocking them out, and restoring the routines when done. Saves a lot of wear and tear, both on the drive and my ears. But...

Jim didn't go far enough. There is a bug to look out for. If you use his form 'XXX DISK!"GO FD00"', you can get into trouble. This routine at $FD00 uses 4 bytes in low memory for temporary storage. The bad thing is where these 4 bytes are located. They are $0213-$0216, which happens to be right in the middle of BASIC'S command/function vector table! To be exact, they are in the vectors for the commands "RUN", "IF" and "RESTORE". If you don't use these 3 commands in your program, and remember to re-boot before trying to run any programs that do, you are o.k.

His form 'XXX DISK!"GO 252B"', works alright, because entering the operating system at this point calls a routine at $2644 which swaps these 4 locations at $0213-$0216 out to $2657-$265A. When the routine at $FD00 is done, the swapper at $2644 is called again to restore BASIC'S vector table to normal.

So, my advice is to avoid calling $FD00 directly, unless you first call $2644, and call it after $FD00, too. Not worth the hassle, when calling $252B does it all for you, and a lot faster.

"Stretch" Manley
Rogue River, OR

* * * * *

ED

AN OPEN LETTER TO JEFF BEAMSLEY

I have just finished reading your column screaming about copyright protection for software and could not help wondering why, as a purveyor of software, you avoided addressing yourself to the very real problem in the marketplace - the garbage that masquerades as 'programs'!

The quick buck artists will always find ways to ignore moral and ethical laws so it is pointless to concern ourselves with them. However, many people - ordinarily ethical and moral - have been so damaged financially by the canned junk on the market that they have become their own vigilanties - out to 'get even' with the guys responsible for their problems.

Where is the protection for the ultimate user who purchases a canned package that does not work - and finds that the warranty only stipulates that (if he's lucky) the package contains a disk and some printed material purporting to be an operating manual? I quote from a 'standard software warranty':

"Refuse Systems, Inc., makes no representations or warranties with respect to the contents hereof and specifically disclaims any implied warranties of merchantability or fitness for any particular purpose."

Why shouldn't the purchaser of this garbage be angry?

Where is the protection for the ultimate user who purchases an OSI package such as AMCAP 1.5 (to be specific) only to find that it doesn't work as advertised and that OSI has 'copped-out' by boldly printing on the front page of their manual PRELIMINARY which, according to the former president of OSI, means that they have no responsibility whatever?

Where is the protection for the ultimate user who is told "...Yes, we know that version 1.0 of the system doesn't work - but, if you'll send us $100 we'll be happy to send you version 1.1 -- which we don't guarantee either!"

It is our feeling - no, belief - that any house which sells (knowingly or otherwise) a software package which does not work as advertised does not deserve the protection of the copyright laws. Your screaming 'copyright violation' at the guy who spent his money and time to make a package work is no different than the man who murders his parents and then throws himself at the mercy of the court because he is an orphan.

Change your ways software purveyors - truly guarantee your packages to work as advertised - refund customer's money promptly - and you'll find most of your problems with copyright violators disappearing.

Joe Brook
Work Force, Inc.
Farmingdale, NY

Joe

The two problems are interrelated. Good authors hesitate to invest 400 hours developing a software system of which they will sell 1 copy per dealer. Slobs will turn out trash at whatever they can get.

Some software purveyors (DBMS, Software Consultants and MANY others) DO stand behind their stuff. We do not accept ads from those we know to be ripoff artists. We DO publish complimentary letters about outstanding products. So do write about specific dealers and products which are good (or bad!) and keep reading PEEK (65) - we will protect each other!

AL

* * * * *

ED:

Reader Fijalkowski has discovered two things most other OS-CP/M users have had to contend with at one time or another.

1. "Cannot get either disk to copy"
2. "No one else around has CP/M"

The OSI disk controller requires and creates a unique disk format. CP/M software disks billed as "standard 8 inch, soft-sectored" won't work on OSI hardware. Most other major hardware vendors offering CP/M use the standard format - but not OSI. However, Lifeboat can deliver all their programs in the unique OSI format.

With the availability of a good implementation of CP/M from Lifeboat (2.2), I think more and more OSI users will switch to CP/M. CP/M's biggest selling boost is OS-65U and its accompanying 8K BASIC.

Thomas L. Robb
White Bear, MN

* * * * *

ED:

I have every copy of Peek (65), to date, and hope to continue receiving helpful information.

Some comments, requests and questions. Remember some of us are relative neophytes. What is CP/M, PIA, and V.I.A.? Tell us what to do if a program listing says for C1P, if we want to run it on our C4Ps!!!

How about some information about various approaches to using a disk based system, which is better?, OS65D or OS65U, and what are the differences?

Finally, can anyone help me figure out how to hook up a Shugart SA400 5" disk to my C4P. I have the disk controller on order. How much RAM do I need? Where do I get a book on the SA400? And what the hell is a C3??

Benjamin F. DeClue III
Peabody, MA 01960

Benjamin:

I can answer some of your questions; I hope readers will answer the others.

CP/M is a Control Program for Microcomputers written by Digital Research in Pacific Grove, California. For disk systems only, CP/M is the program which controls disk input and output, terminal, printer and other input and output and interprets commands input from the terminal. It also stores things like BASIC interpreters on disk and supervises running of programs. A lot of software has been written in several languages to run under the control of CP/M. It only works on machines with the 8080 or Z80 computer chip, like Cromemco, TRS-80 and the OSI C3, Ohio Scientific's professional series of computers which have 3 chips: the 6502 (same as C4P) 6800 and Z80.

PIA: peripheral interface adapter, a chip in a computer progressive computing

Z-FORTH IN ROM by Tom Zimmer	
5 to 10 times faster than Basic. Once you use it, you'll never go back to BASIC!	$ 75.00
source listing add	$ 20.00
OSI FIG-FORTH True fig FORTH model for 0S65D with fig editor named files, string package & much more	$ 45.00
TINY PASCAL Operates in fig-FORTH, an exceptional value when purchased with forth.	
TINY PASCAL & documentation	$ 45.00
FORTH & TINY PASCAL	$ 65.00
SPACE INVADERS 100% machine code for all systems with 64 chr. video. Full color & sound on C2, 4P & 8P systems. The fastest arcade program available.	$ 14.95
PROGRAMMABLE CHARACTER GENERATOR	$ 99.95
Use OSI's graphics or make a complete set of your own! Easy to use, comes assembled & tested.	
2 Mhz. boards	$109.95
PROGRAMMABLE SOUND BOARD	$ 74.95
Complete sound system featuring the AY-3-8910 sound chip. Bare boards available.	$ 29.95
32/64 CHARACTER VIDEO MODIFICATION	$ 39.95
Oldest and most popular video mod. True 32 chr. C1P, or 32/64 chr. C4P video display. Also adds many other options.	
ROMS!!!	
Augment Video Mod with our Roms. Full screen editing, print at,selectable scroll, disk support and many more features.	
Basic 4 & Monitor	$ 49.95
Basic 3	$ 18.95
All 3 for	$ 65.00
65D DISASSEMBLY MANUAL. by Software Consultants	
First class throughout. A must for any 65D user.	$ 24.95

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

CHARGEX VISA

master charge

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

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

which talks to certain devices which work with the computer.

Some programs for C1Ps will also work on C4Ps, some won't, try them out. Who knows how to modify ones that won't for various reasons?

OS65D is a simple Disk Operating System. It handles storing of program and data files on disk and interprets console input commands, just about what CP/M does. It works on mini floppies and 8" floppies in OSI format.

OS65U is a DOS sort of like 65D but: 1) It won't work on minifloppy disks; 2) It contains a BASIC interpreter instead of calling one like 65D can; 3) It does work on OSI's hard disks; 4) It has some powerful extensions like the FIND command which zips through files super fast to locate a string of characters.

Neither is better. 65D won't work on hard disks, doesn't have the FIND command, but is easier to modify and uses less memory.

Readers: is it possible to hook up an SA400?

AL

* * * * *

ED:

After reading Tom Stover's letter in the May issue of Peek (65), I find his views to be most narrow minded. Also, the basis used for determining what is or is not a "high level" language is nothing short of ludicrous. Syntax readability of a high level language is the lowest determining factor I can think of. I wonder if Mr. Stover has ever read APL? This is most certainly a high level language. I believe that Tom has simply missed the whole point of Greg Stevenson's fine article.

It is a well known fact (especially among OSI users) that most personal computer enthusiasts demand all that their system can offer and then some. The most efficient use of a microcomputer system can be realized by effectively balancing the available hardware with well written software. The throughput of any system is usually the greatest when the software is written in the CPU's native code (machine language). This, however, is slow and tedious work for the programmer. Using BASIC, PASCAL, or other "conventional" language greatly simplifies the programmer's task at a great sacrifice in speed. FORTH presents an acceptable compromise between these two extremes. A programmer, using FORTH, may go as high level as he wishes, or as far "down into the machine" as he needs. Many specialized functions can be defined and easily used in a well written FORTH program. This type of extensibility can ease any programmer's task.

To summarize, business programs, games, utilities and specialized control functions can all thrive in a FORTH environment. Unlike the more "conventional" languages used in personal computer systems, FORTH can be tailored to suit each individual's needs and style. Can the same be said for Mr. Stover's "high level languages"?

Rick Lotoczky
Pontiac Twp., MI

* * * * *

ED:

I was somewhat taken aback by a comment in the April '81 issue of Peek (65) by Mr. Arthur Goeres concerning our OS-65D V3.2 Disassembly Manual. He states "Although this manual is incomplete, leaving out several important routines, I never-the-less...". While the rest of his comment on our manual was quite complementary, I would like to take issue with the "incomplete" portion.

Our manual contains absolutely every line of OS-65D V3.2, completely commented. Possibly Mr. Goeres was expecting a disassembly of BASIC or the system ROMs as well, but neither of these is, of course, a part of the operating system. If Mr. Goeres feels that our manual does not live up to his expectations, we will gladly refund his money.

I apologize for using the pages of Peek (65) to answer a customer complaint, but Mr. Goeres must have purchased our manual from a dealer, since he is not in our direct customer file. Our reputation for providing quality products is our most carefully guarded asset, so I felt that I must respond.

J. Larry Hinsley
President
Software Consultants

ED:

Regarding article in Vol. 2, No. 4, April issue, page ten. I called Aardvark regarding C2E ROM, and they said the ROM did not include fix for BASIC STRING BUG.

They said if I want to have fix, I have to buy another ROM which replaces BASIC ROM 3 at $8.00. I saw their ad in Peek (65) in October, and I bought it right away. $8.00 is cheaper than their original BASIC ROM 3 price ($19.95), or almost the same as raw ROM price, however, should I have to pay $8.00 for what they should have included in my C2E? Is that usual in this country that the consumer is cheated by false advertisement? Usually I don't like to say this kind of nasty thing, but the guy who answered my question, (unfortunately I could not get his name), was impolite and made me angry.

I hope Aardvark will be as good as you have been thinking.

Yasuo Morishita
Elk Grove Village, IL

Yasuo:

I took a look at Aardvark's

October ad, and it clearly says the STRING BUG fix is NOT in the C2E ROM, but in a separate ROM which costs $19.95. If they sold it to you for only $8.00, it sounds like they were being more than fair. Let us know how it works.

AL

* * * * *

ED:

In response to Robert Camner's plea for better program copy protection, I offer the following scheme. Though I do not have 65U, this should be easy to adapt to it and also should prove quite effective.

Save the BASIC program. Let us assume it is called PRIVATE. Run SECDIR and make a note of the tracks on which PRIVATE resides. Delete PRIVATE from the directory.

Now write another BASIC program titled PRIVATE (different disk location) and have THIS program, poke a machine-language program into position and USR(X) to it.

Have the machine language program: disable as many of the OS commands as you wish, including probably PUT, SAVE, POKE, perhaps even LOAD and CALL to prevent spying. Then have it place the start and end tracks for the invisible PRIVATE into the A register and $00E5, respectively. Set the X register to 0, and JSR to the load routine of the OS, but skip the first three bytes which otherwise set up the above parameters from the input buffer. Be sure to use BCD for the track numbers. After this machine code has loaded the BASIC program it should be easy to have it jump directly to the start of the BASIC RUN routine. Perhaps even BASIC's LIST should be disabled. At any rate, it should take the students several times as long to crack this scheme as it would to implement it.

Phil Hooper
Northfield, VT

P.S. Tell him not to let his students read Peek (65)!!

Phil:

Right on. 65U does not use whole tracks, but does of course keep track (in DIREC*) of what files are on the disk. Earlier articles and letters in Peek (65) have explained DIREC*. You should be able to hide programs easily enough, so long as the disk is never repacked. Who else has ideas?

AL

* * * * *

ED:

Concerning the OSI communications protocol: I don't know much about microcomputer telecommunications, but I'm afraid we may be on the verge of developing some odd-ball protocol that's completely out of step with what the other microcomputerists are using.

How do these proposed handshaking signals compare to what the ABBS or Forum-80 systems are using? Using these standards, will we be able to interconnect with the Source or Compuserve?

I feel we should make sure we are totally compatible first and then start to make improvements.

David Roha
Oakland, CA

David:

Good thoughts. We will be sure that we are compatible with at least the Source and Micronet, in fact, we will probably wind up using a subset of a very standard protocol designed for large computers.

AL

* * * * *

USER GROUP NOTES:

Schedule for 1981/2: September 5, December 5, March 6, June 5. For more information call Dr. N. Slontseff or Mr. C. Bryce, Unit for Computer Science, McMaster University, Hamilton, Ontario L8S 4K1, (416) 525-9140 Ext. 4689 or 2065.

USED COMPUTER EXCHANGE BEGINS

Frustration has been common among people who want to sell their microcomputer system or to upgrade their peripherals.

Using the classified section of a big city newspaper often takes several weeks. One seller reports that it took four weeks, $100 of advertising and 40 hours talking to "tire kicking" buyers on the phone to sell his (Northstar) system.

Buyers seldom even consider buying a used system because they can't find what they want.

The USED COMPUTER EXCHANGE intends to change all of that.

Buyers, Sellers and people who just want to learn about the market can call the Toll Free number of the Exchange (800-327-9191 Ext. 61).

The UCE is really simply a referral service: quotes and listings are generated using your own criteria. The Exchange gives you complete listings and then you must contact Sellers (Buyers) directly.

$5 buys Quick Quotes. This includes the listing for the item you want with the lowest current offer, the average price of the last month and of the most recent sale and the listing of the Seller who has listed for the longest.

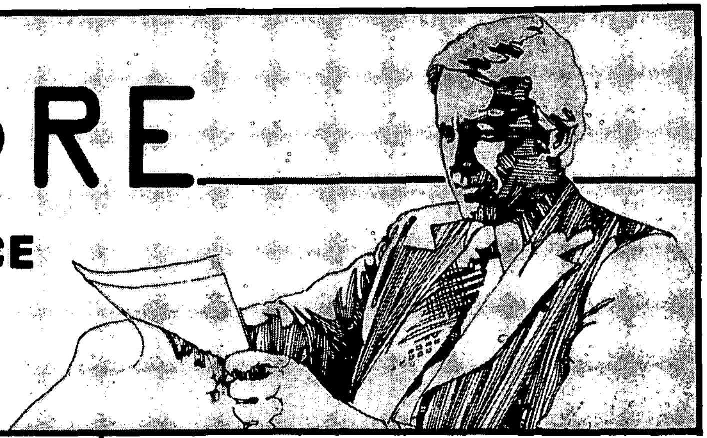
For $10 a customer gets much more information, including: Price History, Current Prices and Additional Listings if needed.

Copies of listings given over the phone are sent in the mail along with a checklist for Buyers and Sellers which includes issues to negotiate, and things to watch out for.

Sellers pay only when their sale is completed with a USED COMPUTER EXCHANGE Buyer. The fee is 5% of the sales price subject to a minimum of $25.

RESTORE

YOUR CONFIDENCE IN SOFTWARE — WITH OURS!



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

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

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

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

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

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

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

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

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

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

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

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

SOFTWARE CONSULTANTS
HARDWORKING *software*

Enclose your check with coupon and mail to: SOFTWARE CONSULTANTS, 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) @ $25.95 ea.

_____CROSS REFERENCE UTILITY(S) @ $31.95 ea.
Disk size 5¼"_____or 8"_____
OS-65D_____or OS-65U_____

_____SPOOLER/DESPOOLER UTILITY(S) @ $69.95.
Std. parallel interface_____
Serial interface w/430 board or equivalent_____
Serial interface w/CA10X board_____

_____FIG FORTH(S) @ $89.95 ea.

_____VIDEO ROUTINE(S) @ $25.95 ea.
With software extensions @ $29.95 ea.

_____SOURCE CODE @ $12.00 ea.
For_____

_____GRAPHIC RESOLUTION BOOSTER @ $49.95 ea.
With Video Routine and extensions @ $79.95 ea.

_____SOFTWARE CONSULTANTS Product Catalog Free

NAME _____

ADDRESS _____

FIRM _____

CITY/STATE _____ **POSTAL CODE** _____

PEEK (65)
The Unofficial OSI Users Journal

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

DELIVER TO: