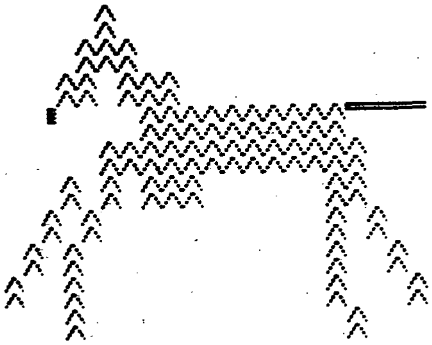
PEEK (65)

The Unofficial OSI Users Journal

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

★$1.50★

Editor: Al Peabody
Vol 2, No.4, April 1981



See Article page 2

Column One

First things first:

THE PEEK(65) CBBS
WILL MOVE TO (301) 363-4867
AS OF MAY 1

This is a big step for the CBBS. By moving to the offices of DBMS, we will get to use the C3-B there, giving us essentially unlimited file space, plus much longer hours. Starting May 1, you may call the board any time after 5:30 Eastern time, right up to 7AM the next morning. Insomniac computer freaks around the nation will be able to leave messages for each other _all night long_!

While we are on the subject of communication, a status report on the OSI communications protocol is in order. The consensus so far is that our protocol should provide for transmission of messages from terminal to terminal without disabling the normal use of the computer (without requiring a RESET to implement communications or return to computing) and should also allow file sharing without requiring the rather clumsy procedure of indirect files. Error checking once per line seems OK, preferably with retransmit of the line several times before breaking off. The receiving computer should be able to stop incoming data, restart transmission, or interrupt the entire process.

What we need, then, is a set of programs capable of running on both serial and video systems, which can:

1) Accept input from the keyboard and send it into the modem;
2) Accept input from the modem and display it;
3) Transmit data from RAM into the modem, a "line" at a time, waiting for the receipt of a code before transmitting the next line, retransmitting a line if a "not OK" code is received perhaps 3 times, then printing an error message on both terminals and breaking;
4) Receive data from the modem and store it in RAM, checking the data received in some way (perhaps a parity check, perhaps a checksum check) and transmitting the OK code if it is correct, or preparing to receive the last line over again if it is not;
5) Go to tape or disk to get the next batch of data to be transmitted after the RAM buffer is empty on the transmit end;
6) Send a "wait" code when the buffer is nearly full on the receive end, store the data away on disk or tape, then reset the buffer for the next batch;
7) Wait as neccesary on the transmit end while the receiving computer puts data away;
8) Allow the operators on both ends to select the operations to be performed;
9) And, of course, monitor the various processes, printing out the appropriate error messages not only if the computer can't understand what is coming over the line, but also if, for example, both ends select to receive data simultaneously: otherwise, you and I might both wait quite a while in this mode!

What control codes do we need for these purposes? At least these:

1) Control-S ("wait")
2) Control-Q ("OK-proceed")
3) Control-C ("break")
4) Control-E ("error")

Now for the next phase. Have a good long look at these specs and see what I have forgotten or planned poorly and write us a note (or leave a message on the CBBS!). Once we have everyone's input, we will write the programs and give them away ... free.

Al

GALLOPING HORSE

by W.G. Libby
12 Tranmore Way
City Beach, W.A. 6015
Australia

The following BASIC program demonstrates a simple means of animation by drawing a horse which gallops across your video monitor. The orginal intent of the program was to passivate a daughter who wanted a real horse as a pet. "Galloping Horse", as written, runs on OSI computers using the 540 video board.

The program contains three data sets describing the horse: one for the head and two for the legs in different postitions. As the horse moves, the leg data is alternately displayed giving the impression of movement. The data consist of offsets from a starting position. Since the 540 video board has 64 characters per line, an offset of 64 or multiple thereof will create a character directly below the starting position. If you are converting this program for use with another computer, you will have to draw the horse out on graph paper and recreate the data files.

This same program can be used to move other objects if you enter the data required. However, the exact number of entries in each file must agree with the dimension statement and with the limits of the FOR-NEXT loops which use the data. The program can be expanded to display more than two positions of movement if you desire. The constant in line 80 controls the speed while line 200 determines the character used to plot the figure. The variable "X" controls the position of the plot and is the base address to which the offset data is added. As "X" is decremented, the horse moves across the monitor screen.

Even this simple animation is taxing the limited speed of BASIC. If you need faster, more detailed movement, you better start learning machine code. The principle would be the same: create several data files of your object which can be alternately displayed at different positions on your video screen.

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

```
10 REM      GALLOPING HORSE
20 FORN=1TO13:PRINT:NEXT
30 PRINT TAB(24);"GALLOPING HORSE"
40 PRINT TAB(24);"==============="
50 PRINT:PRINTTAB(24);"by Will Libby"
60 FORN=1TO13:PRINT;NEXT
70 FOR N=1 TO 1000: NEXT N
80 D=200:REM HORSE SPEED
90 REM SET UP MACHINE SCREEN CLEAR
100 POKE11,64:POKE12,2
110 FORN=576TO599:READI;POKEN,I:NEXT
120 REM CLEAR SCREEN
130 Y=USR(8)
140 DIMT(19),A(36),B(36)
150 REM CONSTANTS FOR POKING HORSE
160 FORN=1TO19:READT(N):NEXT
170 FORN=1TO36:READA(N);NEXT
180 FORN=1TO34:READB(N):NEXT
190 X=54632
200 J=29:REM SYMBOL USED
210 REM BEGIN CYCLE DRAWING BOTH HORSES
220 FORF=1TO100
230 FORN=1TO19:POKEX+T(N),J:NEXT
240 POKEX+190,166
250 FORN=205TO208:POKEX+N,145:NEXT
260 FORN=1TO36:POKEX+A(N),J:NEXT
270 X=X-6:REM NEXT POSITION
280 FORN=1TOD:NEXTN
290 Y=USR(8)
300 REM SECOND HORSE
310 FORN=1TO19:POKE+T(N),J:NEXT
320 POKEX+190,166
330 FORI=205TO208:POKEX+I,145:NEXT
340 FORN=1TO34:POKEX+B(N),J:NEXT
350 X=X-6:REM MOVE TO NEXT POSITION
360 FORN=1TOD:NEXTN
370 Y=USR(8)
380 NEXTF:REM END TWO POSTITION LOOP
390 GOTO190:REM RESTART IN INITIAL POSITION
400 REM DATA ASSOCIATED WITH CLEARING OF SCREEN
410 DATA 169,32,160,8,162,0,157,0
420 DATA208,232,208,250,238,72,2,136
430 DATA208,244,169,208,141,72,2,96
440 REM TOP OF HORSE DATA
450 DATA 1,64,65,66
460 DATA 127,128,130,131,132
470 DATA 195,196,197,198,199,200,201,202,203,204
480 REM BOTTOM OF FIRST HORSE
490 DATA 257,258,259,260,261,262,263,264,265,266
500 DATA 267,268,269
510 DATA 320,322,323,324,325,331,333
520 DATA 384,386,394,396
530 DATA 448,451,457,459
540 DATA 513,516,520,523
550 DATA 578,581,583,587
560 REM BOTTOM OF SECOND HORSE
570 DATA 257,258,259,260,261,262,263,264,265,266,267,268,269
580 DATA 319,321,323,324,325,332,333
590 DATA 382,384,396,398
600 DATA 445,447,460,463
610 DATA 508,511,524,528
620 DATA 575,589
```

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

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

Subscription rates:
US (surface mail) $12/year
Foreign (air mail) $20/year
Back issues .... $1.50/ea.

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

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

WP-3 MODIFICATIONS TO SUPPORT OTHER TERMINALS

by: Daniel C. Smith
DCS Software Products
2729 Lowery Ct.
Zion, IL 60099

PROBLEM. WP-3 as supplied from the factory supports only the Hazeltine 1420 terminal. Terminal-dependent functions used in WP-3 include:

FUNCTION	KEY-STROKE	HEX VALUES SENT FROM TERMINAL
Forward space	<right arrow>	10
Backspace	<left arrow/CTRL-H>	08
Tab right	<TAB/CTRL-I>	09
Skip to start of line	<CTRL-F>	06
Skip to rear of line	<CTRL-R>	12
Display next line	<down arrow>	1B 0B
Display previous line	<up arrow>	1B 0C

Unfortunately, the last two functions cannot be easily modified. The 1420 terminal sends an escape sequence when either the down or up arrow is struck. Since WP-3 expects two characters from the terminal, these keys are handled in a special fashion that is very difficult to modify.

On the other hand, this is a relatively minor loss. All of the functions available under WP-2 are still present for displaying text.

The line-editing features can be modified as follows. First, refer to your terminal manual to determine the values of the codes which must be changed. In particular, find the value for backspace and for forward space. The values for tab and for the skips to start and end of line are not echoed back to the terminal. Instead, the program receives them and sends the terminal a number of forward space or backspace characters.

Second, make a copy of your WP-3 diskette. Boot the copy. When the "." prompter appears, enter into the following dialogue:

```
COMPUTER      YOUR RESPONSE
   .           EXXX<return>
  A*           RE M<return>
```

This puts you into the monitor PROM. The computer sends a <return> to the terminal leaving an empty line with the cursor on the left.

To change the backspace, location $1974 must be changed from its current value of $08 to its new value, say $19. Enter:

```
L197419R
```

When the computer receives the "R" it performs a carriage return.

To change the forward space, locations 1978, $19DE, and $1A2B must be changed from their current value of $10 to your new value, say $19.

```
L197819R
L19DE19R
L1A2B19R
```

Each "R" performs a carriage return to a new line.

To change the character received for tab, location $1988 must be changed from its current value of $09 to the new value, say $19. Enter:

```
L198819R
```

To change the character received for skip to front of line, location $1980 must be changed from its current value of $06 to the new value, say $19. Enter:

```
L198019R
```

To change the character received for skip to end of line, location $1984 must be changed from its current value of $12 to its new value, say $19. Enter:

```
L198419R
```

When you have made your last change, enter:

```
L012E2A51R
G
```

The computer responds with an A*. Enter

```
A*SA 07,1=1800/B<return>
```

The changes are now permanently recorded on disk, and your WP-3 should work normally.

A special problem occurs with Microterm terminals. These terminals use $18 for forward space. Unfortunately, WP-3 interprets $18 as the character used to recall text from an indirect buffer, <CTRL-X>. Therefore, the <CTRL-X> must be changed to some other character. This is accomplished as follows.

Load a standard OS-65D diskette into drive A and boot the system. Put your WP-3 diskette into drive B. Enter in the following dialogue.

```
A*CA 0200=01,2
A*SE B
B*GO 0200

 - DISKETTE UTILITIES -
 1.COPIER
 2.TRACK 0 READ/WRITE
 ? 2

 -  TRACK  ZERO  READ/WRITE
UTILITY -
COMMANDS:
Rnnn-READ INTO LOCATION nnnn.
Wnnnn/gggg,p - WRITE FROM nnnn
               FOR p PAGES
               WITH gggg AS
               THE LOAD VECTOR
E - EXIT TO OS-65D
COMMAND? R6000
(redisplays track 0 menu)
COMMAND? E

B*RE M
L637A7ER   (Note: puts in the
value for "~". "~" becomes the
new replacement for CTRL-X.)
L012E2A51R
G
B*SE A
A*CA 0200=01,2
A*SE B
B*GO 0200
(displays copy menu)
?2
(displays track 0 menu)
COMMAND? W6000/2200,8
```

Finished. Load the WP-3 disk into drive A and boot the system.

TWO DOLLAR BSR X-10 INTERFACE

by Jim Williams
262 Chappel
Calumet City, IL

If you're interested in interfacing the BSR (or Sears, or Radio Shack) wireless AC remote control system to your machine, first read the excellent articles by Steve Ciarcia. The September, 1980 Radio Electronics article includes a good explanation of the system, schematics of the command console, lamp module, and appliance module, and the codes and information formats used for command communication. The January, 1980 BYTE column includes computer controlled hardware to emulate the handheld ultrasonic remote controller. The R-E article is probably sufficient background.

THE HARDWARE

My scheme uses a command console NOT equipped for ultrasonic remote control (they're cheaper) by injecting an appropriately coded pulse train into the chip in the command console through an optoisolator. I mounted my isolator on the circuit board in the command console, using empty holes in the unpopulated area normally occupied by the ultrasonic receiver circuitry.

The normal ultrasonic remote transmitter sends a five-bit code representing which key was pressed in a redundant, pulse-width modulation, serial scheme. In the command console, the received pulses are amplified and shaped and sent to pin 7 of the main IC (still at 40 KHz!). I just replaced the last transistor of the amp/shaper with the phototransistor of the optoisolator.

Diagram 1 shows the last stage of the receiver circuitry. In the non-ultrasonic equipped command consoles, the 8.2K pullup is replaced with a wire jumper. You must remove this jumper and install your own pullup. I went down to 4.7K to help speed up my isolator. Diagram 2 shows the whole interface.

THE SOFTWARE

Listing 1 is the pulse-train generator. It drives the LED in the isolator through one bit of a 6821 PIA at $F700. It expects the 11-bit command string (starting with the LSB: start bit (a 1); five bit key code; logical inversion of five bit key code) to be passed through the USR function.

Each time through the main loop ($1607-$161E), it rotates one of the 11 bits into Carry, and passes (through X and A) the on and off times for the pulse train generator to the generator routine at $1624. After 11 times through, it sets up times for a stop character and falls right into the generator routine. The generator puts out about 0.1 msec of 50KHz for each count in X, and pauses about 0.1 msec for each count in A before returning. (Times based on 1MHz CPU clock.)

The code is quick and dirty: it was easier to make up the command string in BASIC, and it uses the fact that only one bit of the PIA is output as a mask. Actually, my C24P is much too valuable to use to control the BSR; this was just done to debug the hardware. A dedicated MICRO ACE will be in charge ultimately. I plan to do an article for KILOBAUD on it but I planned to get the garage door fixed before winter, too...

```
1600 2005AE JSR $AE05    get USR argument to AE,AF
1603 A90B   LDA #$0B     set up for 11 times thru loop
1605 85B0   STA $B0
1607 66AE   ROR $AE      16 bit rotate into cary
1609 66AF   ROR $AF      16 bit rotate into carry
160B B006   BCS $1613
160D A20C   LDX #$0C     ON time for zero
160F A944   LDA #$44     OFF time for zero
1611 D004   BNE $1617    always
1613 A228   LDX #$28     ON time for one
1615 A92D   LDA #$2D     OFF time for one
1617 202416 JSR $1624    call pulse generator
161A C6B0   DEC $B0
161C A5B0   LDA $B0      (I thought DEC didn't set flags)
161E D0E7   BNE $1607    loop 11 times
1620 A264   LDX #$64     ON time for stop
1622 A964   LDA #$64     OFF time for stop
1624 A007   LDY #$07     set up for 4 cycles of 50KHz
1626 8C00F7 STY $F700    output to PIA
1629 2400   BIT $00      time equalizer
162B 88     DEY          toggle output bit
162C 8C00F7 STY $F700
162F 88     DEY
1630 10F4   BPL $1626    loop at 50 KHz
1632 CA     DEX
1633 DOEF   BNE $1624    do it X times
1635 AA     TAX
1636 A014   LDY #$14     delay loop: 0.1 msec each loop
1638 88     DEY
1639 DOFD   BNE $1638
163B CA     DEX
163C DOF8   BNE $1636
163E 60     RTS
```

LISTING 2

```
2 REM      BSR X-10 DRIVER DEMO
3 REM      JIM WILLIAMS 1/1/81
4 :
9 :   REM SET UP PIA AT $F700
10 PO=63232
20 POKEPO+1,0:POKEPO,1:POKEPO+1,4
29 :   REM SET UP USR ADDR FOR $1600
30 POKE11,0:POKE12,22
34 : REM SET UP TABLE OF KBD CODES TO CYCLE THRU
35 DIM A(20)
40 READQ:FORI=1TOQ:READA(I):NEXT
50 FORI=1TOQ:N=A(I)
60 PRINT N
99 :   REM MANUFACTURE 11-BIT CODE STRING
100 AR=1 OR 2*N OR 64*NOT N
300 Z=USR(AR)
304 :   REM GIVE IT A CHANCE TO TALK TO RCVR MODULES
305 FORZ=1TO300:NEXT
310 NEXTI:GOTO50
499 :   REM DATA FOR 1,ON,2,ON,1,OFF,2,OFF
500 DATA 8,6,20,7,20,6,28,7,28
```

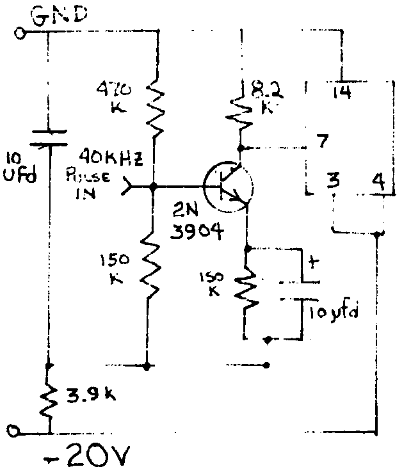
Listing 2, the BASIC driver, is pretty straightforward. If line 100 throws you, you should play with the logical functions of BASIC for a while. (They all operate on 16 bit integers.)

Here, stolen right from R-E, are the key codes:

KEY	"N"	KEY	"N"
1	6	11	12
2	7	12	13
3	4	13	0
4	5	14	1
5	8	15	2
6	9	16	3
7	10	ALL OFF	16
8	11	ALL ON	24
9	14	ON	20
10	15	OFF	28
		DIM	18
		BRIGHT	26

OTHER STUFF

About the command console: To open the box, remove the one screw (not sealed!) in a deep hole in the bottom. The case splits where it looks like it should. To release the catches that hold it together, press in on the bottom half, on the front or back side, about 3/4 inch from the sides, while gently prying the case apart. There are four catches, two each front and back. Yes, you have to remove the AC cord strain relief to get the circuit board out. Yes, you have to remove the jack you put in the back if you want to pull the keyboard. Yes, the LED on the front lights when it gets a valid pulse train on pin 7. All the info in Steve's articles checked out on mine except for the colors of the wires between the main board and the keyboard.

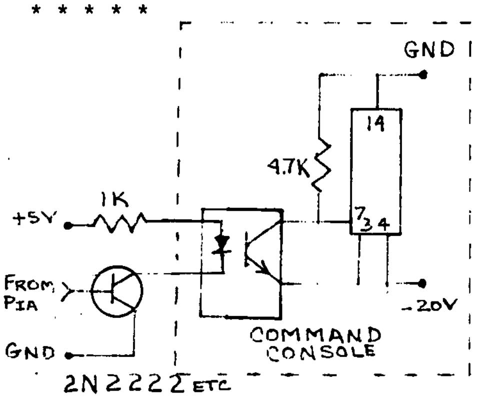


About timing: That 28 pin IC in the command console will accept a pulse train on pin 7 from 40 KHz up to at least 70 KHz. (That's where my optoisolator couldn't keep up.) The program runs around 50 KHz. If you trim it down to 40 KHz, you should be able to use it to drive an ultrasonic transducer. The other timing windows weren't too restrictive, either. For the ON time to send a zero (LDY# at $160D), it worked from decimal values of 8 to 32 (with OFF time always adjusted so ON plus OFF was 80.) The ON time for a one (LDY# at $1613) worked from 34 to 79! (With its OFF time adjusted to give a sum of 80.) The ON time of the stop character (not bit--it was more than one bit time) could be from 80 to 255; since I delay three tenths of a second in BASIC, the stop character OFF time is hard to talk about.

About logic levels: The chip in the console runs with one side of the AC line as ground, and a minus 20 V supply. The switching threshold at pin 7 is about -5 V. I had to play with the value of the 1K LED current limiting resistor to get my slow isolator to reliably cross that threshold. (Too much LED drive current and it never got close enough to ground.) I originally thought pin 7 expected the pulse train as just logic levels at the data rate--not at 40KHz, so I didn't bother to use a very fast isolator. I'd recommend not using a darlington type isolator, both because they're slower and they give you less leeway to use LED current as a control of the phototransistor.

Watch out for damage to pin 7 while you're soldering. After you cut its jumper to ground, it's completely floating and very subject to static damage, at least until you get the pullup in. Good Luck!

* * * * *



C1P AS A DUMB TERMINAL

by James Loos
750 S. Dickerson St. #413
Arlington, VA 22204

"Welcome to the wired nation." Thus begins an article in a recent issue of CABLE TV magazine. In this article the author observes that: "The microprocessors that revolutionized everything from calculators to blenders will...upgrade TV from it's 'boob tube' status into a window on the world, a data bank filled with information on education, security, local events, personal shopping and accounting services." The author goes on to describe an information revolution that has only just started. If you'd like to join the shock troops of this revolution, you'll need a few pieces of combat gear: namely a telephone, a modem, and a terminal. Nearly everyone has access to a telephone. Modems are affordably priced or easily constructed. Your OSI C1P can be persuaded to act as your terminal with the following dumb terminal program.

This particular program has a few features that distinguish it from some of the others you may have seen. First, the program is interupt driven. Incoming data is held in a buffer until the procesor is free to handle it. Data won't be lost if it arrives when the CPU happens to be busy with some other task such as scrolling the screen. Second, the keyscan routine is an enhancement of OSI's polling routine. The difference is that the keyboard looks like a standard typewriter keyboard to the user. With the shift lock key released, you won't need to worry about which shift key shifts what. Finally, the user has the option of specifying from the keyboard whether or not a line feed will be automatically printed after a carriage return. Some systems send a linefeed after each carriage return, others don't. This program has the flexibility to accommodate both.

The listing is the output of a disassembler, hence the lack of comments (I don't have an assembler). The program occupies upper memory locations in an 8K machine but can easily be re-located. The program starts at 1E8F with an initialization routine. Entering control zero clears the screen and restarts the program. Entering control * (asterisk) toggles the line feed flag. The program will normally print a line feed after each carriage return. To defeat this feature, enter control*. The program emulates a full duplex terminal. Data entered from the keyboard won't appear on the screen until it is echoed by the host system.

There is one hardware modification made necessary by the interupt driven nature of this program. Pin 7 of the ACIA (U 14) must be jumpered to pin 4 of the CPU (IRQ).

Since the IRQ output of the 6850 is open-drain, it can be freely connected in "wired-OR" fashion with other sources of interupts.

* * * * *

```
*Q:1E7B-1EFF


1E7B 48        PHA
1E7C 8A        TXA
1E7D 48        PHA
1E7E AD 01F0   LDA $F001
1E81 A6 EA     LDX $EA
1E83 95 E0     STA $E0,X
1E85 E8        INX
1E86 8A        TXA
1E87 29 07     AND #$07
1E89 85 EA     STA $EA
1E8B 68        PLA
1E8C AA        TAX
1E8D 68        PLA
1E8E 40        RTI
1E8F A2 08     LDX #$08
1E91 A0 D0     LDY #$D0
1E93 84 EB     STY $EB
1E95 A0 00     LDY #$00
1E97 84 EA     STY $EA
1E99 A9 20     LDA #$20
1E9B 91 EA     STA ($EA),Y
1E9D C8        INY
1E9E D0 FB     BNE $1E9B
1EA0 E6 EB     INC $EB
1EA2 CA        DEX
1EA3 D0 F6     BNE $1E9B
1EA5 A9 0D     LDA #$0D
1EA7 20 2DBF   JSR $BF2D
1EAA A2 13     LDX #$13
1EAC BD 7B1E   LDA $1E7B,X
1EAF 9D C001   STA $01C0,X
1EB2 CA        DEX
1EB3 10 F7     BPL $1EAC
1EB5 A9 03     LDA #$03
1EB7 8D 00F0 STA $F000
1EBA 85 EA   STA $EA
1EBC 85 EB   STA $EB
1EBE A9 8D   LDA #$8D
1EC0 8D 00F0 STA $F000
1EC3 8D 3002 STA $0230
1EC6 58      CLI
1EC7 A6 EB   LDX $EB
1EC9 E4 EA   CPX $EA
1ECB F0 0D   BEQ $1EDA
1ECD B5 E0   LDA $E0,X
1ECF 20 E41F JSR $1FE4
1ED2 E8      INX
1ED3 8A      TXA
1ED4 29 07   AND #$07
1ED6 85 EB   STA $EB
1ED8 10 ED   BPL $1EC7
1EDA A9 01   LDA #$01
1EDC 20 BEFC JSR $FCBE
1EDF 20 C6FC JSR $FCC6
1EE2 D0 05   BNE $1EE9
1EE4 0A      ASL A
1EE5 D0 F5   BNE $1EDC
1EE7 F0 43   BEQ $1F2C
1EE9 4A      LSR A
1EEA 90 09   BCC $1EF5
1EEC 2A      ROL A
1EED E0 21   CPX #$21
1EEF D0 F3   BNE $1EE4
1EF1 A9 1B   LDA #$1B
1EF3 D0 21   BNE $1F16
1EF5 20 C8FD JSR $FDC8
1EF8 98      TYA
1EF9 8D 1302 STA $0213
1EFC 0A      ASL A
1EFD 0A      ASL A
1EFE 0A      ASL A
1EFF 38      SEC
1F00 ED 1302 SBC $0213
1F03 8D 1302 STA $0213
1F06 8A      TXA
1F07 4A      LSR A
1F08 20 C8FD JSR $FDC8
1F0B D0 1F   BNE $1F2C
1F0D 18      CLC
1F0E 98      TYA
1F0F 6D 1302 ADC $0213
1F12 A8      TAY
1F13 B9 CFFD LDA $FDCF,Y
1F16 CD 1502 CMP $0215
1F19 D0 16   BNE $1F31
1F1B CE 1402 DEC $0214
1F1E F0 1B   BEQ $1F3B
1F20 A0 05   LDY #$05
1F22 A2 C8   LDX #$C8
1F24 CA      DEX
1F25 D0 FD   BNE $1F24
1F27 88      DEY
1F28 D0 F8   BNE $1F22
1F2A F0 9B   BEQ $1EC7
1F2C A9 00   LDA #$00
1F2E 8D 1602 STA $0216
1F31 8D 1502 STA $0215
1F34 A9 04   LDA #$04
1F36 8D 1402 STA $0214
1F39 D0 8C   BNE $1EC7
1F3B A2 96   LDX #$96
1F3D CD 1602 CMP $0216
1F40 D0 02   BNE $1F44
1F42 A2 14   LDX #$14
1F44 8E 1402 STX $0214
1F47 8D 1602 STA $0216
1F4A A9 01   LDA #$01
1F4C 20 BEFC JSR $FCBE
1F4F 20 CFFC JSR $FCCF
1F52 4A      LSR A
1F53 B0 36   BCS $1F8B
1F55 D0 16   BNE $1F6D
1F57 A0 00   LDY #$00
1F59 AD 1502 LDA $0215
1F5C C9 D0   CMP #$D0
1F5E F0 09   BEQ $1F69
1F60 AD 1502 LDA $0215
1F63 30 38   BMI $1F9D
1F65 C9 40   CMP #$40
1F67 30 34   BMI $1F9D
1F69 A0 20   LDY #$20
1F6B D0 30   BNE $1F9D
1F6D A0 F0   LDY #$F0
1F6F AD 1502 LDA $0215
1F72 C9 D0   CMP #$D0
1F74 F0 11   BEQ $1F87
1F76 AD 1502 LDA $0215
1F79 30 22   BMI $1F9D
1F7B A0 00   LDY #$00
1F7D C9 40   CMP #$40
1F7F 10 1C   BPL $1F9D
1F81 A0 10   LDY #$10
1F83 C9 21   CMP #$21
1F85 B0 16   BCS $1F9D
1F87 A0 00   LDY #$00
1F89 F0 12   BEQ $1F9D
1F8B AA      TAX
1F8C 29 03   AND #$03
```

```
1F8E F0 0B    BEQ $1F9B
1F90 A0 10    LDY #$10
1F92 AD 1502  LDA $0215
1F95 10 0C    BPL $1FA3
1F97 A0 F0    LDY #$F0
1F99 D0 08    BNE $1FA3
1F9B A0 00    LDY #$00
1F9D E0 20    CPX #$20
1F9F D0 02    BNE $1FA3
1FA1 A0 C0    LDY #$C0
1FA3 AD 1502  LDA $0215
1FA6 29 7F    AND #$7F
1FA8 C9 20    CMP #$20
1FAA F0 07    BEQ $1FB3
1FAC 8C 1302  STY $0213
1FAF 18       CLC
1FB0 6D 1302  ADC $0213
1FB3 8D 1302  STA $0213
1FB6 C9 F0    CMP #$F0
1FB8 D0 03    BNE $1FBD
1FBA 4C 8F1E  JMP $1E8F
1FBD C9 FA    CMP #$FA
1FBF D0 13    BNE $1FD4
1FC1 A2 00    LDX #$00
1FC3 AC 3002  LDY $0230
1FC6 F0 06    BEQ $1FCE
1FC8 8E 3002  STX $0230
1FCB 4C C71E  JMP $1EC7
1FCE 8D 3002  STA $0230
1FD1 4C C71E  JMP $1EC7
1FD4 48       PHA
1FD5 AD 00F0  LDA $F000
1FD8 29 02    AND #$02
1FDA F0 F9    BEQ $1FD5
1FDC 68       PLA
1FDD 8D 01F0  STA $F001
1FE0 4C C71E  JMP $1EC7
1FE3 00       BRK
1FE4 C9 20    CMP #$20
1FE6 B0 14    BCS $1FFC
1FE8 C9 0A    CMP #$0A
1FEA F0 10    BEQ $1FFC
1FEC C9 0D    CMP #$0D
1FEE D0 0F    BNE $1FFF
1FF0 AD 3002  LDA $0230
1FF3 F0 05    BEQ $1FFA

1FF5 A9 0A    LDA #$0A
1FF7 20 2DBF  JSR $BF2D
1FFA A9 0D    LDA #$0D
1FFC 20 2DBF  JSR $BF2D
1FFF 60       RTS
*P:
```

CASSETTE CORNER

by David A. Jones
8902 SW 17 Terrace
Miami, FL

For the SII/C1P owner who doesn't have the Extended Monitor, saving machine language programs presents a real problem. OSI made no provisions to save such programs nor gave hints on how to do it. Eventually one learns to save and load these programs via BASIC PEEK and POKE methods but this has its drawbacks. The BASIC program must be loaded both to save and to load programs, it occupies memory that may be required for other purposes, page zero locations are modified, etc.

The following program saves code in the 65VP format so no special routine is required to reload the desired code the next time. The program makes extensive use of existing routines in the OSI Monitor and BASIC ROM's. Specifics of these routines are: ACIA at $FCB1 sends one ASCII character out the serial port. LEGL at $FE93 converts an ASCII character to a HEX nibble, or returns a negative number if not a legal HEX character. DSPL at $FEAC splits an 8 bit word into 2 nibbles, converts them to ASCII and displays the ASCII characters at $DOC6-DOCD. ROLA at $FEDA assembles ASCII characters into 8 bit words. The resulting word is left in FC,X. The original contents of FC,X are rolled into FD,X. Both X and Y registers are used. Note that X must be set to some predetermined value for this routine to provide consistent results.

INPT at $FEE9 gets one character from the keyboard or ACIA. DISP at $BF2D displays the accumulator character on the screen.

By using these proven routines, programming is quickened and the resulting code is already partially debugged upon writing. Familiarity with the Monitor is improved and additional uses for the original routines are found. The binary to ASCII conversion routine at $FEAC is a prime example. I doubt that OSI intended it to be used in the manner this program does. (The video display memory is used as a temporary store.)

To use the program after it is loaded, go to the monitor and type 18AOG, an S should be displayed. Type in the starting address of the program to be saved followed by a dash then the ending address followed by a carriage return. If an error is made, just retype the complete address before the dash or carriage return as the program only accepts the last 4 entries for each address. You will see the starting address displayed at the top of the screen as if you were in the monitor address mode and also at the bottom of the screen. The ending address will be displayed only at the bottom. While the tape is being recorded you will see the address and data changing on the screen. The final address displayed is the final address location that was recorded. The program then branches back to the start should you want to record twice. To load a program from a recorded tape go to the monitor and type L, as explained in your user's manual.

Once a program is loaded the monitor will automatically branch to the entry point. Just type in the starting address of your routine to run it.

The code is assembled at $18AO for easy conversion if one wants to put it into EPROM immediately following the Checksum Load routine described in last month's column.

A word of advice though. While this program does the job, there is no check for accuracy while loading and a tape dropout can cause the entire program past the dropout to be in error. Also, since each word is followed by a carriage return, the storage method is not very efficient. I heartily recommend the purchase of the Extended Monitor by anyone who has more than just a passing interest in assembly language programming. Both the SAVE and LOAD routines in the ExMon are in the checksum format.

CASSETTE CORNER
LISTING ON PAGE 21

FAST & VERSATILE SCREEN CLEAR
===CLS VER 2.2 (1981)===

by Yasuo Morishita
405 Lively Boulevard
Elk Grove Village, IL 60007

I have been developing my software for personal use. I have written many versions of CLEAR SCREEN subroutines like everybody has been trying. Here I want to show you one of my favorite screen clear routines which can fill the screen with any ASCII character (in case of ASCII 32, it is a screen clear) or change screen color with the C4P. After initialization what you have to do is insert a USR(P)Q statement at any location you want in your BASIC program;

```
 U=USR(P)SYMBOL
  P:if P=208  ($D0),fill
    screen with CHR$(SYMBOL).
    if P=224  ($E0) fill
    screen with the color
    defined by code SYMBOL.
```

For example, if you want to clear the screen and change the color to green you may write as follows:

```
10 D=208:E=224:CL=32:GR=4:A=65
20 U=USR(D)CL :REM clear
   screen
30 U=USR(E)GR :REM green color
   if you want to fill the
   screen with "A", then

40 U=USR(D) A
```

If you want to change the color quickly for a game effect;

```
50 FOR COL=4TO20:
   U=USR(E)COL :NEXT COL
```

This may be too fast to recognize color change, so you may insert a time delay loop in a FOR NEXT loop such as FOR T=0 TO 100:NEXT T.

Before using this feature you have to POKE the machine language subroutine shown below into RAM such as starting at $0235(565 decimal) in page 2, using a BASIC program with DATA statement or the MONITOR. Then the USR starting address into $0B(11), and $0C(12). If it starts at $0235 then we POKE as follows:

9 POKE11,53:POKE12,2:REM 11 for LOW byte and 12 for HIGH byte.

$0235

```
1 FOR A=565 TO 565+31:READ M
  :POKE A,M:NEXT
2 DATA 32,5,174,165,175,133,
      250,32,173,170,32,5,
      174,165,175
3 DATA 160,0,132,249,162,8,
      145,249,200,208,251
4 DATA 230,250,202,208,
      246,96
```

```
CLS2.2 JSR $AE05 ;GET VARIABLE
       LDA $AF   ;FROM BASIC
       STA $FA   ;SET SCREEN
                 PAGE
       JSR $AAAD ;EVALUATE
                 EXPRESSION
       JSR $AE05 ;GET VARIABLE
       LDA $AF   ;FROM LINE
       LDY #0    ;RESET LOW
                 BYTE
       STY $F9
       LDX #8    ;NO. OF PAGES
LPCLS  STA ($F9),Y
       INY
       BNE LPCLS ;#$FB
       INC $FA   ;NEXT PAGE
       DEX
       BNE LPCLS ;#$F6
       RTS
```

For MONITOR use, the HEX codes are;

```
 20 05 AE,A5,AF
 85 FA   ,20 AD AA
 20 05 AE,A5 AF
 A0 00   ,84 F9
 A2 08   ,91 F9
 C8      ,D0 FB
 E6 FA   ,CA
 D0 F6   ,60
```

Good luck!

COMMENT TO MR. HOOPER'S VERY fine article vol.1,#12.Page 7

Though his article is super, there are a couple of typos such as in Listing #2;

```
 1002 DATA 3,23,166
   should be 32,23,166
 1007 DATA 156,20
   should be 165,20
Listing #3;
 1003 DATA 165,175
   I guess 165,174(LDA $AE)
   for High byte,
 1005 DATA 165,174
   I guess 165,175(LDA $AF)
   for Low byte, isn't it?
```

If you use CEGMON instead of SYNMON like me, you have to change line 1001 in listing 2 to 1001 DATA 162,26(LDX #$1A) because subroutine $FEDA rolls data from $F9 & $FA instead of $FC & $FD.

Since the OSI MONITOR strips off bit 7 when a program is loaded through the ACIA, I do not think line 1 can be loaded. This can be saved without any trouble, but we will have big trouble when we try to load it with the standard LOAD command. Another problem; when we write several subroutines in one line using ",", as he mentions, you have to put JMP $030A; we cannot modify them if they are in ROM.

NEW MONITOR "CEGMON"

As you know, the OSI monitor SYNMON does not allow us to edit BASIC programs easily. This is very inconvenient if you have to modify your program. I recently saw an ad in PEEK (65) regarding a new monitor called CEGMON (C2E) from Aardvark Technical Services. It was $59.95 plus handling charge and seemed fairly expensive. But overall I'm satisfied with it.

Since my machine is a C4P (ROM), I have several problems, which I'm going to show you here. 1) For a C4P you have to modify the 502 board with messy wires and piggy back. It is not just a -replace- type fix! But the instructions supplied are good. For me to have a schematic drawing is lots easier. I hope they will add it from now on!
2) You have to buy another chip (74LS154) to complete the job. 3) Though they claim this has a fix for the string handling bug in OSI BASIC, it does not!! When my programs have string manipulation in them, they still hang up! I have to set string array dimensions at 3*N+2 and PRINT FRE (0) once in a while. Is something wrong with my C2E chip? Or is Aardvark guilty of false advertising? Please advise me if anyone knows about this!

Except for the above mentioned problem, this is a great tool. It will work with BASIC and OSI Assembler. Break point, Block move, Usr(x), near complete ASCII conversion from key board, Instant screen clear and so on. I disassembled CEGMON and I have a BASIC program of DISASSEMBLER (takes about 4.5K RAM) and a BASIC RENUMBER (takes about 3K RAM). If anyone is interested, let me know. (Since I don't have a printer yet, I had to write it down by hand! ----Oh poor Yasuo!!----)

Yasuo:

Aardvark has always been scrupulously honest, so I am very surprised that you have a problem with your CEGMON. I suggest you contact them directly - let us know what happens.

AL

* * * * *

Art Work for PEEK(65)

Art work submitted for publication in PEEK(65) should be 3" (one column), 6 1/2" (two columns) or 9 3/4" (full page) in width by not over 13 1/2" high (full page).

Assembler Programmer's Guide to OSI Board Interfacing

Tabulated for Digital Technology, Inc.
by:
Ken Holt of Virginia Computer Consultants

OSI 550 Board (CA-10)

The 550 board, also known as the CA-10, is a multiple-port serial interface board. It supports up to sixteen serial devices, each connected to its own ACIA. Each of the sixteen ports is given a "port number". The specific designation is in a state of confusion, as some OSI documentation refers to them as 0 thru 15 while other OSI sources say 1 thru 16. To permit further discussion, we must choose one of the two. The designation of 1-16 will be used in order to be consistent with OS-65U documentation. To those of you who prefer the 0-15 instead, we offer our apologies.

The CA-10 board can be ordered with the number of ports desired, the remaining areas of the board being available for later expansion. The CA-10 is then designated as a CA-10-4 if it has four ports, CA-10-1 if only one, and so forth. Often the board is called CA-10-X when it is uncertain how many ports it will contain.

Each port uses two addresses: the first is for control/status; the second is for the data path. The ports occupy a contiguous range of addresses. Thus, adjacent port numbers have addresses which are two apart.

550 Board Addressing (CA-10)

Address	Read	Write
CF00	Port 01 Status Reg.	Port 01 Control Register
CF01	Port 01 Data Path	Port 01 Data Path
CF02	Port 02 Status Reg.	Port 02 Control Register
CF03	Port 02 Data Path	Port 02 Data Path
CF04	Port 03 Status Reg.	Port 03 Control Register
CF05	Port 03 Data Path	Port 03 Data Path
CF06	Port 04 Status Reg.	Port 04 Control Register
CF07	Port 04 Data Path	Port 04 Data Path
CF08	Port 05 Status Reg.	Port 05 Control Register
CF09	Port 05 Data Path	Port 05 Data Path
CF0A	Port 06 Status Reg.	Port 06 Control Register
CF0B	Port 06 Data Path	Port 06 Data Path
CF0C	Port 07 Status Reg.	Port 07 Control Register
CF0D	Port 07 Data Path	Port 07 Data Path
CF0E	Port 08 Status Reg.	Port 08 Control Register
CF0F	Port 08 Data Path	Port 08 Data Path
CF10	Port 09 Status Reg.	Port 09 Control Register
CF11	Port 09 Data Path	Port 09 Data Path
CF12	Port 10 Status Reg.	Port 10 Control Register
CF13	Port 10 Data Path	Port 10 Data Path

Address	Read	Write
CF14	Port 11 Status Reg.	Port 11 Control Register
CF15	Port 11 Data Path	Port 11 Data Path
CF16	Port 12 Status Reg.	Port 12 Control Register
CF17	Port 12 Data Path	Port 12 Data Path
CF18	Port 13 Status Reg.	Port 13 Control Register
CF19	Port 13 Data Path	Port 13 Data Path
CF1A	Port 14 Status Reg.	Port 14 Control Register
CF1B	Port 14 Data Path	Port 14 Data Path
CF1C	Port 15 Status Reg.	Port 15 Control Register
CF1D	Port 15 Data Path	Port 15 Data Path
CF1E	Port 16 Status Reg.	Port 16 Control Register
CF1F	Port 16 Data Path	Port 16 Data Path

OSI CA-9 Board

The CA-9 board is a 470 board wired to interface with a Centronics-compatible printer. The interface consists of two eight-bit paths to the printer: one for control, another for the printed data. A PIA is used, with ports A and B used for printer control and printer data, respectively. An additional signal not derived from the PIA is used to strobe the printer data interface.

PIA Data Register A Layout:

PRIM	1	LCP	FALT	SLCT	POUT	ACK	BUSY
(Out)	(In)	(In)	(In)	(In)	(In)	(In)	(In)

PIA Data Register B Layout:

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

PRIM - Prime (Clear) (Negative Logic)
LCP - Line Count Pulse
FALT - Fault Detected (Negative Logic)
SLCT - Selected
POUT - Out of Paper
ACK - Character Acknowledged (Negative Logic)
BUSY - Printer Buffer is Busy
DBn - Printer Data Bit n

CA-9 Board Addressing

Address	Read	Write
F400	Port A: Prt Cntrl	Port A: Printer Control or DDA0 thru DDA7
F401	PIA: Port A Ctrl	PIA: Port A Control
F402	PIA: PB0 thru PB7 (Normally unused)	Port B: Printer Data
F403	PIA: Port B Ctrl	PIA: Port B Control
F420	Data Strobe	Data Strobe

digital technology

inc.

SOFTWARE CATALOG #7

APRIL 1, 1981

an advertising supplement to PEEK (65)

BUS-II BOOKKEEPING & ACCOUNTING SYSTEM

The BUS-II turn-key multi-client accounting package is the leading OSI business software package. Version 3.Ø is considered to be free of problems, surprises, and bugs. Version 3.1 (under development) will contain a number of enhancements, including integrated Order Entry w/ Inventory.

BUS-II is a batch-oriented accounting system--the simplest and safest to use. The system is ideal for use by CPA firms, bookkeeping services, or small businesses. The BUS-II package provides the user with highly flexible formatting of financial reports. The system allows branch accounting (or multiple profit centers) within a firm. Client accounting for bookkeepers or CPA firms provides multi-client operation.

BUS-II modules can now be purchased separately--although the BUS-II "special" which includes the G/L, A/R, A/P, O/E, and PAYROLL (retail price $995) is by far the most popular configuration.

GENERAL LEDGER
ACCOUNTS RECEIVABLE
ACCOUNTS PAYABLE
ORDER ENTRY W/ INVENTORY
PAYROLL

POINT-OF-SALE TERMINAL (see below)
CPA EXTENSIONS (see below)

Note: BUS-II operates on floppy- or hard disk-based systems under the OS-65U V1.2 operating system (Level I, II, or III). Multi-client use can accommodate any number of client companies on floppy disk systems or hard disk systems with H/D/E. Files are limited in size for floppy disk back-up; floppy disk operation continues in case of hard disk failure.

BUS-II LEVEL II

BUS-II LEVEL II (under development) is designed for much larger businesses. Expanded file size and special operations allow virtually unlimited numbers of accounts and transactions. BUS-II LEVEL II requires BUS-II LEVEL I. Minimal back-up is data casette (tape) or floppies--although multiple Winchester disk operation is recommended (provides ability to continue computerized bookkeeping functions in case of hard disk failure).

POS-1 POINT-OF-SALE TERMINAL

POS-1 is an on-line multi-store point-of-sale terminal program with integrated inventory designed for cash register emulation. POS-1 controls cash drawer and ticket printer (or system printer). Automates taxable or non taxable sales, cash transactions, credit sales (with verification operations). POS-1 is interactive with the BUS-II BOOKKEEPING & ACCOUNTING SYSTEM; optionally, it may be used in stand-alone mode.

CPA EXTENSIONS PACKAGE

CPA EXTENSIONS (under development) is designed for public accounting firms. A number of special operations are provided: Statement of Changes in Financial Position, Changes in Components of Working Capital, Cash Flow Analysis, Departmentalized Sales Analysis, Comparative Income Statement, Budgetary Analysis, Asset Depreciation, and Loan Amortization. CPA EXTENSIONS is interactive with BUS-II and requires BUS-II LEVEL I or LEVEL II.

TAXMAN-1040 PERSONAL INCOME TAX PREPARATION

TAXMAN-1040 is designed for tax practitioners and public accountants. TAXMAN-1040 the leading tax package for OSI microcomputers--the package has been in use on OSI, Hewlett-Packard, DEC and IBM. Designed and supported by a CPA tax expert. This package automatically prepares FORM 1040 and 28 schedules. Individual state tax option available. Support includes annual forms & tax tables revisions. Purchasers of 1980 TAXMAN will automatically receive 1981 revisions at no extra charge.

TAXMAN-1120 CORPORATE INCOME TAX PREPARATION

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

OS-DMX DATABASE MANAGEMENT SYSTEM

Command-oriented OS-DMS compatible database management system. OS-DMX operates under the OS-65U V1.2 (LEVEL I, II, or III) operating system. Features such as control files, extensive operational commands, and the innovative HELP feature, in addition to Digital Technology's exclusive on-line documentation, make this one of the most usable--as well as powerful--systems available for microcomputers. OS-DMX may be used instead of, or in addition to, OS-DMS Nucleus, Query, Sort; OS-DMX will replace virtually all of the specialized OS-DMS modules--and in most applications will provide greatly improved performance.

ECR-1P ELECTRONIC CASH REGISTER POLLING

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

ECR-1C DATA CASETTE POLLING

ECR-1C provides data cassette polling, allowing multi-store cash register polling. ECR-1C is recommended when diverse store locations make telephone line communications prohibitively expensive.

SALES-1 SALES ANALYSIS

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

INV-1 RESTAURANT INVENTORY & MENU EXPLOSION

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

H/D/E HARD DISK EXECUTIVE

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

H/D/M MULTI-USER MANAGER

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

BISYNC-8Ø/HASP

BISYNC-8Ø/HASP is a full-function Multileaving Workstation package which allows communication with any remote CPU that supports a HASP Multileaving Workstation, and, as such, is ideally suited to Remote Job Entry applications. A User Program Interface allows user-written programs to provide reader stream records to be transmitted to the remote and allows user processing of print and punch stream records received from the remote.

BISYNC-8Ø/ASYNC

BISYNC-8Ø/ASYNC is a full-function asynchronous communications package which allows microcomputers to communicate asynchronously with a mainframe or other microcomputers. This package is an ASYNC adaptation of BISYNC-8Ø/378Ø terminal emulation program, providing asynchronous communications at 75 to 96ØØ baud, using full IBM BISYNC protocol. In microcomputer-to-mainfrome mode, the program gives enhanced asynchronous communications ability--the software emulates a "dumb" asynchronous terminal, but allows transmitting files to the remote and allows data from the remote to be directed to the microcomputer console or an attached printer. A User Program Interface allows user-written programs to provide data to be transmitted to the remote and allows user programs to process data received from the remote, and to initiate and terminate data exchange.

BISYNC-8Ø/378Ø

BISYNC-8Ø/378Ø is a full-function IBM 278Ø/378Ø emulator allowing the microcomputer to communicate over point-to-point telephone lines with any CPU or device that provides standard IBM support for:

IBM 278Ø Models 1, 2, 3 or 4
IBM 378Ø w/ or w/o 3781 card punch
IBM CPU to CPU BSC communications

The package is ideally suited to Remote Job Entry and can be used to transmit and receive large volumes of batched data. RJE console support greatly simplifies operator remote status inquiries and control of remote processing jobs. A User Program Interface allows user-written programs to supply data records to be transmitted to the remote and to process data records received from the remote.

BISYNC-8Ø/327Ø

BISYNC-8Ø/327Ø is a full-function IBM 327Ø terminal emulator which allows the microcomputer to communicate over point-to-point telephone lines with any IBM S/36Ø, S/37Ø, or S/3Øxx CPU that provides standard IBM support for one of the following:

IBM 3275 Model 2
IBM 3271 Model 2 or control unit w/ attached 3277 Model 2
IBM 3284 or 3286 printer

The software is ideally suited to matching transaction-oriented IBM systems with the local processing power of a microcomputer. The software supports transmitting and receiving video display screens of data in IBM 327Ø format. A User Configuration Module allows any key of an addressable-cursor video terminal to be defined as the equivalent of any 327Ø key. The 3284/3286 printer may be a real printer or a pseudo-printer, defined as a file, allowing the collection of mainframe data for later offline processing. A User Program Interface allows user-written programs to provide the data to be transmitted to the remote and to process data received from the remote, and to initiate and terminate data exchange.

Digital Technology, Inc., is the largest independent supplier of OSI software with hundreds of business packages in use around the world. Digital Technology software is sold by a growing number of conscientious OSI dealers and OEMs. Every package is backed by the finest support program in the microcomputer industry. All "bugs" are fixed free of charge. Updates (fixes to bugs, minor enhancements, new product announcements) are provided to all dealers and licensed users free of charge. And upgrades to new versions are encouraged (at nominal charge).

Digital Technology's software is user-oriented. In fact, no one else provides such expansive features as on-line documentation, idiot-proof prompting, and operator's manuals that are comprehensive, detailed, and accurate.

All Digital Technology software systems allow the operator to "set" the programs to the type of video terminal and printer used. The operator selects the terminal and printer types from the list provided in the "TERMINAL & PRINTER OPTIONS" program. Screen formatting and printer control are provided automatically yet may be redefined through user subroutines.

Nearly all popular video terminals, including those listed below, are supported and are user-selectable from the Terminal Options menu (drivers for most other terminals available at nominal charge):

NULL (NO SCREEN FORMATTING)
MICRO-TERM
LEAR SIEGLER
BEEHIVE
HEATH
HAZELTINE
INFOTON (GENERAL TERMINALS)
INTERTEC INTERTUBE
ADDS REGENT SERIES
SOROQ
TELEVIDEO
CDC VIKING
VISUAL

Currently supported output devices are user-selectable from the Printer Options menu:

PORT 3	CA-6	SERIAL (RS-232)	OSI 43Ø OR EQUIVALENT
PORT 5	CA-9	8-BIT PARALLEL	LINE PRINTER INTERFACE
PORT 6	CA-9D	12-BIT PARALLEL	NEC 55ØØ, DIABLO, QUME
PORT 8	CA-1Ø	SERIAL (RS-232)	SERIAL PORTS Ø1-16

A number of additional features are provided by the terminal and printer control program--DEVSET, an exclusive feature of Digital Technology software. DEVSET controls printer operation ad paging for all printers and screen formatting through generalized subroutines, so program modification is unnecessary. DEVSET user notes and technical information are provided with every system.

OSI SOFTWARE PRICE LIST

APRIL 1, 1981

BUS-II - G/L, A/R, A/P, O/E, PAYROLL	$ 995	
BUS-II GENERAL LEDGER	$ 495	
BUS-II ACCOUNTS RECEIVABLE	$ 495	
BUS-II ACCOUNTS PAYABLE	$ 495	
BUS-II ORDER ENTRY W/ INVENTORY	$ 495	
BUS-II PAYROLL	$ 495	
BUS-II LEVEL II	$ 995	(a)
CPA EXTENSIONS	$ 995	(a)
POS-1 POINT-OF-SALE TERMINAL	$ 995	(b)
TAXMAN-1040 PERSONAL INCOME TAX PREPARATION	$ 1495	(b)
TAXMAN-1120 CORPORATE INCOME TAX PREPARATION	$ 1995	(a)
H/D/E HARD DISK EXECUTIVE	$ 395	(f)
H/D/M TIME-SHARE MULTI-USER EXECUTIVE	$ 795	
OS-DMX DATABASE MANAGEMENT SYSTEM	$ 695	
OS-DMX-MAIL MAIL LIST MANAGEMENT	$ 495	(c)
OS-DMX-STAT STATISTICAL ANALYSIS	$ 195	(c)
ECR-1P CASH REGISTER POLLING	$ 995	(c)
ECR-1C DATA CASSETTE POLLING	$ 995	(c)
SALES-1 SALES ANALYSIS	$ 495	(c)
INV-1 REST INV. & MENU EXPLOSION	$ 995	(d)
OS-BISYNC-80/HASP	$ 800	(d)
OS-BISYNC-80/3780	$ 550	(e)
OS-BISYNC-80/3270	$ 550	(e)
OS-BISYNC-80/ASYNC	$ 95	(e)
OS-BISYNC-80 SYNCHRONOUS INTERFACE ASSY	$ 895	(e)

(a) BUS-II required
(b) BUS-II optional
(c) DMX required
(d) ECR-1 recommended
(e) Subject to demand
(f) C2-, C3-D: $295 / C3-B, C, C´: $395

digital technology
inc.

P.O. BOX 178590
SAN DIEGO, CA 92117
(714) 270-2000

CA-9 Interface Pinout

Cable Pin	Interfaced to	Description
1	17 Addr $F420	Data Strobe (Negative)
2	08 PB0	Data Bit 1
3	07 PB1	Data Bit 2
4	06 PB2	Data Bit 3
5	05 PB3	Data Bit 4
6	04 PB4	Data Bit 5
7	03 PB5	Data Bit 6
8	02 PB6	Data Bit 7
9	01 PB7	Data Bit 8
10	23 PA1	Acknowledge (Negative)
11	24 PA0	Busy
12	22 PA2	Paper Out
13	21 PA3	Selected
14	GD GROUND	GROUND
15	NC	Interface Clock
16	GD GROUND	GROUND
17	NC	NOT USED
18	+5 +5 Volts	+5 Volts
19	GD GROUND	GROUND for Twisted Pair
..	. .	. . . . .
..	. .	. . . . .
30	GD GROUND	GROUND for Twisted Pair
31	18 PA7	Prime (Clear) (Negative)
32	20 PA4	Fault Detect (Negative)
33	NC	Paper Jammed
34	19 PA5	Line Count Pulse
35	GD GROUND	Line Count Pulse Ground
36	NC	NOT USED

OSI CA-9D Board

The CA-9D is a 470 board wired as a Diablo-compatible 12-bit parallel interface. Since the computer is an 8-bit device, some special tricks are required. The eight low-order bits are sent out directly on PIA port B. The four high-order bits are first sent to Port B in the four low-order bits. Then a special strobe (via a bit in port A) transfers them to a latch where they are held as an extension to PIA port B.

There are three strobes which cause the printer to act upon the 12-bit data word in different ways. They are: carriage strobe to move horizontally, feed strobe to feed the paper (for vertical movement), and print strobe to print a character.

The sequence to send a 12-bit data word to the printer is as follows: (1) Put the four high-order bits in PB0 thru PB3. (2) Now, strobe the approprite bit in port A, causing the latch to be loaded with the four high-order bits. (3) Put the eight low-order bits in PB0 thru PB7. (4) Activate the proper data strobe to send the data to the printer.

Because of the large number of strobe and status lines required, PIA port A does not provide quite enough lines. To solve the problem, a few of the special handshake features of the PIA are used. Bits 2, 4, and 5 of both PIA control registers must be set to 1. This allows the CA2 and CB2 pins of the PIA to be directly controlled by bit 3 of the respecitve control register. That gives the board two more strobe output lines, which is just enough. Bits 2, 4, and 5 of the control registers must be set to 1 to allow CA2 and CB2 to be used as strobes. If this requirement is ignored, the interface will not work properly.

Diablo Interface

The Strobe Signals:

Restore - Active Low. Restores the carriage and print wheel to their home positions. Also clears a check condition.

Prt Strobe - Active Low. Interprets the low order 7 bits of the 12-bit data register as a printable ASCII character.

Feed Strobe - Active Low. Interprets the low order 11 bits of the 12-bit data register as a paper movement command.

Carr. Strobe - Active low. Interprets the 12-bit data register as a carriage movement command.

Other Status Signals:

Check - Low indicates failure of previous command. Disables carriage, feed, and print ready. Clearable only by Restore strobe.

Device Ready - Low indicates device is powered on.

Carriage Ready - Low indicates carriage ready to accept commands.

Feed Ready - Low indicates feed circuits ready to accept commands.

Prt Wheel Rdy - Low indicates print wheel ready to accept commands.

12-bit Data Register Usage for Print Wheel Strobe:

NU	NU	NU	NU	NU	DB7	DB6	DB5	DB4	DB3	DB2	DB1

DB1 thru DB7 are used to specify the ASCII character to be printed. (Printable char only.)

**Note: Bits are negative logic.

12-bit Data Register Usage for Feed Strobe:

NU	FMDS	FM9	FM8	FM7	FM6	FM5	FM4	FM3	FM2	FM1	FM0

FMDS - Feed Movement Direction Specification
0 - Forward
1 - Reverse

FM0 thru FM9 represent a 10-bit binary quantity of vertical feed units. (1/48th inch each)

**Note: Bits are negative logic.

12-bit Data Register Usage for Carriage Strobe:

CM0	CMDS	CM10	CM9	CM8	CM7	CM6	CM5	CM4	CM3	CM2	CM1

CMDS - Carriage Movement Direction Specification
0 - Forward
1 - Reverse

CM0 thru CM10 represent an 11-bit binary quantity of horizontal carriage units. (1/120th inch each)

**Note: Bits are negative logic.

**Please note the odd placement of the CM0 bit.

LETTERS

ED:

I read with interest your plan to obtain OS-CP/M and start an OS-CP/M column in PEEK (65).

I run CP/M and Microsoft BASIC-80 on my C3-OEM, starting first with OS-CP/M 1.4 from OSI. When CP/M 2.x was announced I waited for OSI to offer it as an upgrade to licensed OS-CP/M 1.4 users. And waited, and waited, ...finally, I purchased the new version from Lifeboat. I am very pleased with the new CP/M but even more pleased with the Lifeboat implementation of CP/M as compared to the OSI implementation. In addition to some errors and omissions, the OSI version does disk I/O by reverting back to the 6502 track-at-a-time code of OS-65U. That may be fine for a simplistic file structure like OS-65U has but can cause very slow disk operations when used with a dynamically expandable file structure such as CP/M's.

Your readers may be interested in a Computerworld (Feb. 23, 1981) benchmark evaluation of an OSI C3 as compared to several other popular micros and minis. While characterizing the hardware as "reliable, cost effective" the independent consultants said of the OSI documentation, "an uncomplete collection of ideas, suggestions and features randomly assembled and poorly reproduced" and of OSI applications offered, "together with the documentation problem, the software was of questionable value". To which I can only add: Hear! Hear!

In my opinion, OSI would do well to get out of the software business (system and application) and leave that business to companies and publishers that have demonstrated quality products in these areas; such as Digital Research, Microsoft and Lifeboat.

Thomas L. Robb
White Bear, MN

* * * * *

ED:

Reading "Column One" I was surprised to see what could have been at least in part, my own column. I own a C3 OEM and at one time had some wild dreams such as that also. Although one might say I may have ultimately realized them, I have my reservations. I would like to share some of the trials and tribulations I've had and maybe you can avoid them. I ordered my C3 in November 1979, but I didn't get my machine until March 1980. By then I was quite anxious about it since at the time I was going to school and desperately needed it to back up my school files. I was taking evening courses and the school was very restrictive on evening computer time.

This is where I got my ideas about communicating with other machines but ignorance is bliss and not knowing the pitfalls I'd encounter, I blundered on. One school quarter passed after another. Summer passed, school started again and I got so involved with this project that I decied to skip the winter quarter. I've finally reached a workable solution but certainly not in time for the courses that I was taking.

I suppose that in many of my problems most people might say "It serves you right for being so stupid", but then I was learning. I chose to write what most have called the "Modem" in assembler. Most of my friends encouraged me to write it in BASIC but I felt that BASIC created more problems than it would solve. First, if BASIC doesn't get the input it expects it either gives an error message and waits for correct input or takes control and terminates the program. When dealing with a mainframe somewhat the same results occur and you may end up with an error handshake loop. I found that OSI assembler has many pitfalls also that are sometimes difficult to trace. Not knowing how the assembler generates the source file in memory I started the routines in location 4000 hex and soon found out that parts of my programs quit working. Since most of my associates worked with BASIC I had to discover on my own that the source program was eating up my object code or vice versa. After that bout and after about four hours of generating source code I tried to save my hard efforts. But the machine would not save it. After a !PUT FILENAME <CR> all I got was a high pitched whine. Needless to say this really upset me. I couldn't get any answers from anyone as to the cause of this. Like the intimidations from a vengefull slayer, I couldn't tell when this would crop up again and it did, over and over again. I was told by the dealers that since it was out of warranty it would cost a king's ransom to locate the problem since I didn't know how to recreate the condition at will. Well to make this very long story short, I finally found out on my own that this condition is caused by exiting and/or entering the assembler through nonstandard corridors. That is, if the source has been assembled, the reset button pressed and the assembler is re-entered by machine level methods. Eg. L012D282A51RG. And then RE A will return to the assembler and the source may even be intact. However, loading or saving at this point is impossible.

I preferred to use the OS-65D assembler over the one in the word processor because it appeared easier to use. However, hidden undocumented problems really raise havoc when least expected. When I started generating a file disk for the files to ultimately be saved, I felt I needed a disk with many small files. I found however that even though I had 8" double sided disks, OS-65D CREATE would not let me create more than 32 of the available 64 files. Again with no help from others I had to debug OSI's Create program to be able to create all 64 files.

One of the difficulties I had with my "Modem" program was that on occasion there would be a need to reboot the system and OSI designed the disks to reboot in BASIC and run BEXEC* before anything else happened. If a file existed in memory that you've just down-loaded from another system at 300 baud, you've lost that file if you have to reboot. At this point I had invested in OS-65D DISASSEMBLY MANUAL from SOFTWARE CONSULTANTS. Although this manual is incomplete leaving out several important routines, I never-the-less found it to be indispensable in integrating my modem routine into the disk boot and bypassing BASIC.

I also found the need to go to CP/M because available software for OSI is very limited. But that ended up being a whole new bag of worms for me again. I ordered it from Lifeboat and after a somewhat lengthy mail interchange I received my CP/M and since it wasn't cheap I handled it with my nerves enflamed. I no longer trusted the disk drives and was afraid I would blow the CP/M. I read the instructions carefully which were to copy CP/M before anything else. But I found I couldn't Boot CP/M. Well there is about 6 months of story here and the ultimate results (I think) are that I had insufficient head pressure to read disks that would have a slight wobble to them. My machine is in the garage now getting fixed, I hope.

OSI BASIC doesn't permit print using statements and if software was available with print using statements in them it may be quite difficult to program around that. I chose to get Microsoft's Basic 80 that included print using statements. Even if there are variations in the way the statements are written, this can easily be resolved. There is a loose end however, in the way Lifeboat handles the sales of this software. After ordering this you have to sign a software registration form. Lifeboat has no way of linking this to the order and in my case, when I returned it, it was ignored.

Now that I have CP/M going I found it desirable to be able to exchange files with OS-65D and vice versa. I found that it can be done but you have to have your head screwed on right while you're doing it because there is a lot of monkey motion involved in getting the files into the right memory area, sized and trimmed. OS-65D starts files at 317E hex and requires start and end addresses. CP/M starts files at 0100 hex and requires only to know how many pages.

I'm sure happy however, that I've found someone now though that has somewhat the same inclinations that I have. I found it very discouraging pioneering by myself. Even in the local OSI User Group there are only two other C3 users and one of them is a BASIC only business machine and the other is an OSI salesman's toy, that is not used for much else than playing with. I would like to see much more information on C3s. Maybe when you get your system going we can become electronic pen pals, although I just hate writing letters electronically or any other way. Good luck with your system.

Arthur Goeres
Portland, OR

Arthur:

Wow! You have taught yourself to program in BASIC and Assembler, written a communications program, fixed 65D CREATE, learned how to bypass running BEXEC*, improved on Larry Hinsley's disassembly manual and figured out how to store fixes between 65D and CP/M. Thanks very much for sharing your experiences. Now how about sharing the exact details so we who are following after you won't have to make all the same mistakes? What did you change in CREATE? How do you keep from running BEXEC*? Would you share your modem program? What, exactly does it do?

AL

* * * * *

ED:

This is to inform you that a contract has been let on the life of Dick McGuire, your technical editor. After having spent many, many, hours of puzzling over why the PACKER utility was screwing up my disks, several days ago while taking a shower I discovered the problem and immediately (after turning off the water) made the fix Dick mentioned in the January issue of PEEK (65). I was in the process of composing the letter to you to inform the world of my discovery, when PEEK (65) arrived and dashed my hopes of making a splash in the OSI world. At this point, I'm not sure if I'm more aggrieved because of being beat out or because he didn't find it early enough to spare me my agony!

A question or two for either you or your readers: 1. On page 3 of the January issue, a reference is made in the Software Federation, Inc. ad, to OS65U version 1.3111 Does it exist? Is it more than a bunch of fixes of the bugs in 1.2 (complete with their own bugs)? If it's useful, how do I get it? 2. My clever students have discovered that the WRITE protection that one can give to programs under 65U can easily be circumvented, as follows: Create a scratch file of 24K or at least large enough to hold the program one wishes to pirate. Then run the program desired, using RUN"name"; just as soon as the disk starts loading the program, depress and hold CTRL C until BREAK is reported (this is before the program starts running, so disabling CTRL C in the first line of the program is not useful). Then SAVE the program to the scratch file, reload, and voila, the "protected", "run only" program has been obtained. Short of simply disabling CTRL C in BEXEC* (which I don't wish to do, or else the kids can't terminate programs) and also disabling POKE (so they can't restore CTRL C before stealing the program), what can I do?

Robert Camner
Washington, D.C.

Robert:

1. The ad is talking about WP-6502 V1.3, which runs under OS-65U, any version - there is no 65U V1.3. 2. Load "BEXEC* and install the following routine early-line 1 or so:

```
    POKE 2093, 76
    POKE 2094, 160
    POKE 2095, 255
```

This replaces the first 1 1/2 instructions after a control - C is detected with the instruction JMP FFA0, which will, in effect, reboot the computer and give you "H/D/M?" when control - C is pressed.

This not foolproof, but will allow programs to be aborted while still protecting your files.
Who can think of a better way? Best ideas will be printed.

AL

* * * * *

OSI

SOFTWARE FOR OSI

OSI

A JOURNAL FOR OSI USERS!!

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

1) Using String Variables.

2) High Speed Basic On An OSI.

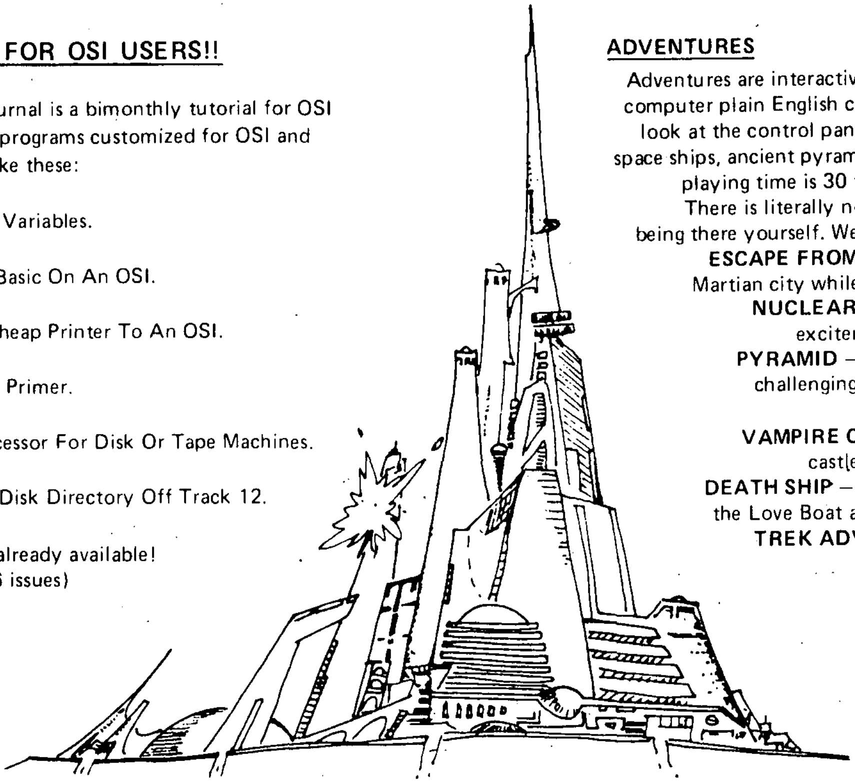
3) Hooking a Cheap Printer To An OSI.

4) An OSI Disk Primer.

5) A Word Processor For Disk Or Tape Machines.

6) Moving The Disk Directory Off Track 12.

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



ADVENTURES

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

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

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

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

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

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

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

NEW SUPPORT ROMS FOR BASIC IN ROM MACHINES

C1S — for the C1P only, this ROM adds full screen edit functions (insert, delete, change characters in a basic line). Softwave selectable scroll windows, two instant screen clears (scroll window only and full screen), software chose of OSI or standard keyboard format, Bell support, 600 Baud cassette support, and a few other features. It plugs in in place of the OSI ROM. NOTE: this ROM also supports video conversions for 24, 32, 48 or 64 characters per line. Replaces video swap tape on C1P model 2. All that and it sells for a measly $39.95.
C1E/C2E for C1/C2/C4/C8 Basic in ROM machines. This ROM adds full screen editing, software selectable scroll windows, keyboard correction (software selectable), and contains an extended machine code monitor. It has breakpoint utilities, machine code load and save, block memory move and hex dump utilities. A must for the machine code programmer replaces OSI support ROM. Requires installation of additional chip when installed in a C2 or C4. C1 installation requires only a jumper move. Specify system $59.95.

DISK UTILITIES

SUPER COPY — Single Disk Copier
This copy program makes multiple copies, copies track zero, and copies all the tracks that your memory can hold at one time — up to 12 tracks at a pass. It's almost as fast as dual disk copying. — $15.95
MAXIPROSS (WORD PROCESSOR) — 65D polled keyboard only - has global and line edit, right and left margin justification, imbedded margin commands, choice of single, double or triple spacing, file access capabilities and all the features of a major word processor — and it's only $39.95.

P.C. BOARDS

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

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

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

ARCADE AND VIDEO GAMES

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

TIME TREK (8K) — real time Startrek action. See your torpedoes move across the screen! Real graphics — no more scrolling displays. $9.95
INTERCEPTOR C1P ONLY! An all machine code program as fast and smooth as the arcades. You use your interceptor to protect your cities from hordes of enemy invaders. A pair of automatic cannons help out, but the action speeds up with each wave of incoming ships. The fastest and most exciting C1P game yet.
C1P Cassette $19.95
MINOS — A game with amazing 3D graphics. You see a maze from the top, the screen blanks, and then you are in the maze at ground level, finding your way through on foot. Realistic enough to cause claustrophobia. — $12.95

SINGLE STEPPER / MONITOR

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

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

AARDVARK IS NOW AN OSI DEALER!

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

— THIS MONTH'S SPECIALS —

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

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

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

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

VISA

master charge

OSI

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

OSI

OSI-FORTH 2.0 / FIG-FORTH

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

Mr. Clean

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

VIDEO BANNER

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

ADVENTURE-65

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

DISK-PAK 1

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

MANUALS

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

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

CONNECTORS

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

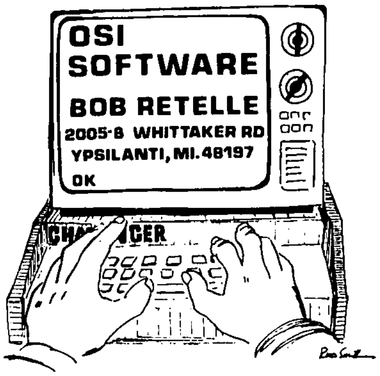
INFO

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

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

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

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



ED:

When I finally got a printer hooked up to my RS232 port on my C1P system, the first thing I did was to get listings of all my programs. Just to be able to see the complete listings on paper was a tremendous relief. Only being able to see 24 lines at a time and trying to debug a program drove me crazy. The next thing I did with my printer was to graph the typical sine wave like everybody else does. After copying a program out of a book and printing the sine wave on paper, I noticed that too much time was taken up printing a character, then carriage returning, tabbing out to the next position, printing, carriage returning, tabbing, etc., etc. And at 300 baud it takes nearly forever to print a decent sine wave. So I wrote a small program to take care of the problem.

Lines 5 and 7 determine the starting position for a sine wave, about in the middle of a 132 column printer. Same with line #20. (the #60) Lines 10 and 15 determine graph and size or put in your own formula. The program tabs out to the proper position, prints a "*", line feeds, back spaces, prints the next character, line feeds, etc. Line #45 checks on the accumulation of characters and spaces printed, and if it exceeds the terminal size it will carriage return, then tab out to where the graph left off.

```
 1 REM ---FAST   PRINT---
 2 REM SET TERMINAL WIDTH TO
     255
 5 C=60
 6 PRINTTAB(C)
 7 D=60
10 FORA=0T020.0STEP.3
15 X=SIN(A)
20 Y=X*15+60
23 IFINT(Y)>CTHEN28
24 IFINT(Y)=CTHENC=C+1:GOTO40
26 PRINTCHR$(8);:D=D+1:C=C-1:
     IFINT(Y)<>CTHEN26
27 C=C+1:GOTO40
28 PRINT" ";:D=D+1:C=C+1:IFINT
     (Y)<>CTHEN28
29 C=C+1
40 PRINT"*";CHR$(10);:D=D+2
45 IFD>PEEK(15)THEND=C:GOTO47
46 GOTO50
47 PRINTCHR$(13);
48 PRINTTAB(C);
50 NEXTA
60 END
```

Brant L. Baun
Newbury Park, CA

ED:

To start, I would like to say that I really enjoy your magazine and I thought that it was time to add my own tidbits. My system is a C1P MF with a single drive and OS65D3. 1. To update your system from tape to disk I found that by:

```
 POKE 23,0
 DISK!"IO,01"
 - LOAD TAPE -
```

in immediate mode will allow a tape to load without causing the DOS to jump out to error messages. The POKE sets the terminal length to double space. After loading the tape - POKE 23,132 before editing the tape.

2. For people who don't like the cursor character the way it is, it is possible to change it to any character you desire: POKE 9682, X (X = whatever character).

This location for the C1, and I believe for the C4, is 9680.

3. Easy key detection is accomplished by:

```
 XXX DISK!"GO FD00",
```

where XXX is the line #. This detects a key being pressed. To detect a key and get a value:

```
 XXX DISK!"GO 252B":
     P=PEEK (9059).
```

For a string character: P$=CHR$ (PEEK(9059)).

To use the USR(X) routine to do the same trick use:

```
XXX Q1=PEEK(8955):Q2=
    PEEK(8956)
XXX POKE 8955,43:POKE 8956,37
     .
     .
XXX X=USR(X):P=PEEK(9059)
    :P$=CHR$(PEEK(9059))
```

4. If you would like the RUBOUT for the character delete (instead of shift O) add the following to BEXEC*:

```
POKE 1394,127:POKE 1419,127
POKE 2820,127:POKE 1386,128
```

Jim Zajac
Dearborn Hts. MI

* * * * *

Brant:

Thanks for the program. It would certainly be faster than CR's and tabs. Readers should note that this program will only work on printers which recognize CHR$(8),(10) and (13) as backspace, line feed without CR and CR.
AL

* * * * *

ED:

I recently purchased an OSI C4P cassette-based machine and I use it with an RCA portable B/W T.V. which I have had lying around the house for awhile. The trouble is I can't see what characters I key into the television screen until I press the RETURN key and they are shifted up one. In other words, I need some way to lower the voltage to the vertical portion of the tube---shrinking it so that I can see the bottom of the screen where the characters first appear.

I've talked to several local T.V. repair shops by phone and, as far as they are concerned, I'm speaking Chinese. Actually I've been asking them if there is a way to modify a television for direct entry video to alleviate this problem.

Also I have a tendency to set the T.V. on top of the computer since it's small enough to fit and does not appear to receive interference. Can this, in any way, do harm to either the C4P or the T.V.? Your response and suggestions from readers will definitely be appreciated.

Ronald Boyd
Corpus Christi, TX

* * * * *

ED:

I just ordered a Sams book from DBMS and I suspect this note will be redundant. But I was glad to see your good news about OSI's take over by MACOM. Nothing could be worse than OSI factory response. I have a CIII with bad disk problems and they have not fixed it in several trips to the factory since November 1979!

I was also intrigued by your interest in CP/M. Frankly if the OSI dealers would get off the 65U and 65D kick (and it would appear to be the push from the factory which kept them there) and get on to a super Operating System in CP/M, I believe OSI CIII owners would be better served. I use WordStar under CP/M and that alone makes it all worth it, even though one has to do some work to get the printer to work on OSI CP/M. I am anxious to see what other software you review and what you say in your CP/M column. I am a learning programmer who spends a great deal of time on APPLE computers and uses OSI in processing the journal, CRYPTOLOGIA, a journal devoted to all aspects of cryptology. I do subscription material and manuscript preparation on the OSI. Let me know if you want some comments about producing technical manuscripts with WordStar on the Spinwriter. I should be glad to submit an example for a camera ready copy publication in the column if you wish.

I am anxious to find a software/hardware solution to the problem of reading and using non OSI CP/M disks, e.g. other 8" CP/M disks. Do you know of any easy solution which can be implemented by a novice? Does anyone have a solution? I would appreciate hearing from anyone with such a solution. Thank you.

Brian J. Winkel
Albion, MI

Brian:

Glad to hear you now have your printer working with WordStar! As I understand the problem of using non-OS CP/M disks in your computer: CP/M is the operating system, the program which tells the disk controller (among other things) what track and section of the disk to read, how many bytes to read and where to put them in RAM. However, the actual disk drive itself determines the precise physical location on the disk which is accessed and exactly how to interpret the magnetic patterns it finds there as I's and O's. A disk recorded on (for) another computer may use any one of a number of physical recording formats, all within CP/M. The disks you use must be not only CP/M, but physically compatible with your computer.

Fortunately, Lifeboat has a pretty good selection of CP/M software at decent prices, all available in OSI Compatible Format. Of course, if you were trying to use CP/M disks belonging to someone else who had a Cromemco... but that would be illegal anyway.

I prefer 65U to CP/M: faster, Level III available, much better disk system, etc. But WordStar is written for CP/M, so I need it, too. Do send printed samples and a note telling us exactly how you got your NEC to work with WordStar.

AL

* * * * *

CONTINUED FROM PAGE 8

```
0000                  ;SAVE,2-18-81
0000                  ;BY DAVID A. JONES
0000                  ADDF=$FE     FROM ADDRESS
0000                  ADDT=$F0     TO ADDRESS
0000                  ACIA=$FCB1   RECORD CHARACTER
0000                  LEGL=$FE93   CHECK FOR LEGAL HEX
0000                  DSPL=$FEAC   DISPLAY OUTPUT
0000                  ROLA=$FEDA   ROLL INPUT INTO ADDRESS REG
0000                  INPT=$FEE9   SERVICE KEYBOARD
0000                  DISP=$BF2D   DISPLAY INPUT
18A0                  *=$18A0
18A0 A90D     STRT    LDA #$0D      CARRIAGE RETURN
18A2 202DBF           JSR DISP
18A5 A953             LDA #'S       PRINT S FOR SAVE
18A7 202DBF           JSR DISP
18AA 20E9FE   FROM    JSR INPT      GET START ADDRESS
18AD 202DBF           JSR DISP      DISPLAY IT
18B0 C92D             CMP #$2D      IF DASH THEN
18B2 F014             BEQ TO        GET END ADDRESS
18B4 2093FE           JSR LEGL      HEX CHARACTER ?
18B7 30F1             BMI FROM      IF NOT GET ANOTHER
18B9 A202             LDX #2        $FC+2=$FE, FROM ADD REG
18BB 20DAFE           JSR ROLA      MOVE IT IN
18BE B1FE             LDA (ADDF),Y GET DATA FOR THIS ADDRESS
18C0 85FC             STA $FC       STORE IT HERE AND
18C2 20ACFE           JSR DSPL      DISPLAY ADDRESS AND DATA
18C5 4CAA18           JMP FROM      ON SCREEN AT $D0C6 ETC
18C8                  ;
18C8 20E9FE   TO      JSR INPT      GET END ADDRESS
18CB 202DBF           JSR DISP      DISPLAY IT
18CE C90D             CMP #$0D      CAR RET
18D0 F00D             BEQ XMIT      END ADDRESS COMPLETE
18D2 2093FE           JSR LEGL
18D5 30F1             BMI TO
18D7 A2F4             LDX #$F4      $FC+F4=$F0, TO ADD REGISTER
18D9 20DAFE           JSR ROLA
18DC 4CC818           JMP TO
18DF                  ;
18DF A92E     XMIT    LDA #'.       SET ADDRESS MODE
18E1 20B1FC - 15BF    JSR ACIA
18E4 A200             LDX #0
18E6 BDC6D0   ADRS    LDA $D0C6,X SEND START ADDRESS
18E9 20B1FC   15BF    JSR ACIA  BF15
18EC E8               INX
18ED E005             CPX #5
18EF D0F5             BNE ADRS
18F1 A92F             LDA #'/       SET DATA MODE
18F3 20B1FC   15BF    JSR ACIA
18F6                  ;
18F6 A000             LDY #0
18F8 A200     DATA    LDX #0
18FA BDCCD0   NIBB    LDA $D0CC,X GRAB CHAR FROM SCREEN
18FD 20B1FC   15BF    JSR ACIA      SEND IT
1900 E8               INX
1901 E002             CPX #2
1903 D0F5             BNE NIBB      GET OTHER HALF OF WORD
1905 A90D             LDA #$0D      CR TO INCREMENT ADDRESS
1907 20B1FC   15BF    JSR ACIA
190A                  ;
190A E6FE             INC ADDF      ADDRESS OF NEXT CHARACTER
190C D002             BNE *+4
190E E6FF             INC ADDF+1    IF NECESSARY
1910 C8               INY
1911 98               TYA
1912 48               PHA
1913 A000             LDY #0
1915 B1FE             LDA (ADDF),Y GET NEXT CHARACTER
1917 85FC             STA $FC       PUT IT IN $FC
1919 20ACFE           JSR DSPL      AND ON SCREEN
191C                  ;
191C A5FE             LDA ADDF      CHECK FOR LAST
191E C5F0             CMP ADDT
1920 D006             BNE MORE
1922 A5FF             LDA ADDF+1
1924 C5F1             CMP ADDT+1
1926 F005             BEQ LAST
1928 68       MORE    PLA
```

LISTING CONTINUED ON PAGE 22

CONTINUED FROM PAGE 21.

```
780 1929 A8              TAY
790 192A 4CF818          JMP DATA
800 192D                 ;
810 192D 68     LAST     PLA          CLEAN UP STACK
820 192E A200            LDX #0
830 1930 BD4319 END      LDA REST,X   RESET FLAG
840 1933 F006             BEQ LINE
850 1935 E8              INX
860 1936 20B1FC  15BF    JSR ACIA
870 1939 D0F5            BNE END
880 193B A90A   LINE     LDA #$0A     LINE FEED
890 193D 202DBF          JSR DISP
900 1940 4CA018          JMP STRT
910 1943 2E     REST     .BYTE '.FE00G'
910 1944 46
910 1945 45
910 1946 30
910 1947 30
910 1948 47
920 1949 00              .BYTE $0
```

USER GROUP NOTES:

SOUTHERN ONTARIO

McMaster University
General Sciences Building
Room 312

Schedule for 1981: June 6th, September 5th, December 5th. For more information call Dr. N. Solntseff or Mr. C. Bryce

Unit for Computer Science
McMaster University
Hamilton, Ontario L8S 4K1
(416) 525-9140 Ext. 4689 or 2065

NEW OSI DISCUSSION GROUPS

2nd Tuesday of Month

For those interested in discussions of cassette-based systems, hardware, C-1P,C-4P, and Basic-in-ROM, and Polled Keyboard systems. Topic for first meeting on April 14th: PIA/VIA hardware (input/output - interfaces). Experiences in selling software.

3rd Tuesday of Month

For those interested in OS-65D, Polled keyboard,5 1/4" and 8" disks, home systems, control features, and MDMS.

4th Tuesday of Month

For those interested in OS-65 U, Serial terminals, single and multi-user, DMS, 8" Floppy and hard disk, and business applications.

All meetings at 7.30 p.m. at Community Computers, 2704 N. Pershing Dr., Arlington, VA (near intersection with Washington Blvd.)

Anybody interested in a user group in Munich or West - Germany, please contact: Christoph Krinninger, Friedensstr. 3, 8016 Feldkirchen, West - Germany

Ohio Scientific Users Group of Northern California forming: Write Rod Freeland, c/o Public Interest Computer Services, P.O. Box 1061, Berkeley, CA 94701, or call (415) 654-9880 after 1:00 P.M.

AD$

BIG BAG $25 (BIGGEST BAG (5 1/4" DISK) of software you can buy for the bucks) 1) Single Disk Copier. 2) Terminal Emulator makes a terminal out of your computer. 3) High Memory Disk Buffers. 4) Data Management System (not OSI). 5) Disk Head Load-Unload for all disk I/O. 6) Spool to Disk (better than indirect memory). 7) VTOC Fixer. 8) Menu of Files at Boot Time. 9) File Transmission Program via phone lines with error checking and recovery. New PERSWP personal word processor for OS65D C1P/4P. PERSWP is $15 with BIGBAG, $20 sold separately. For OS65D C1P or C4P. Detailed documentation is available for $5, refundable when software is purchased.
Computer Power
3223 Suffolk Lane
Fallston, MD 21047
Phone 301-692-6538

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

C4PMF ACCOUNTING SYSTEM

A double entry accounting system for small businesses based on the IDEAL (TM) Professional Bookkeeping and OSI MDMS structure. Prints Income Statement, Trial Balance, Chart of Accounts, and more. System is configurable by user and requires OS65Dv3 and MDMS capability with minimum 8K workspace. Program and data disks plus User guide $100.00. Program listings $20.00 ea. VIDEO VENTURES, 1708 Beechwood, Fullerton, CA 92635

BASXR OS65U compatible. Helps with debugging and modification. Lists all variables and/or commands and their line number locations. Locates specific lines on entry of decimal value of Basic commands. Complete program listing, documentation, and program disk, $50 No COD'S. EIS, Inc., P.O. Box 5893, Athens, GA 30604, 404-353-2858.

Yes. There is an alternative to time consuming software that ends up costing you more in backtracking and worry than it's worth...

It's the quality OSI software available from SOFTWARE CONSULTANTS. We're an independent software vendor dedicated to providing you with the kind of quality programs you need and want. In fact, we've centered our business around meeting the needs of the OSI user.

Most of our products are modifications and/or expansions of existing OSI packages, but some are complete replacements of them. We also have other things you'd never get from OSI, like a completely commented Disassembly Manual for 65D and a video system enhancement routine for better looking graphics and games.

If you're in the market for cost-effective, reliable, and above all, complete software, you're in the market for SOFTWARE CONSULTANTS. Take a look at the list below and see for yourself...

1. OS-65D V3.2 DISASSEMBLY MANUAL With CROSS REFERENCE LISTINGS...

- 50 pages of Disassembly listings in standard Assembler format.
- Fully commented code--not a useless program listing.
- 10 pages of Cross Reference listings--a complete, computer-generated concordance.
- Full-sized (8½" x 11") book lies flat on your desk for convenient use.

2. REF COMMAND UNDER BASIC

A Complete Cross Reference Utility...

- High speed, memory resident utility available on OS-65D or OS-65U.
- Lists any or all occurrences of an individual Basic variable, line number, or numeric constant.
- Generates a sorted listing of all occurrences of any variable, and sends it either to printer or screen.
- Available either on 5¼" or 8" floppy disks.
- Requires 1K of memory. Can either be automatically loaded into the high area of memory (48, 32, or 24K), or under the OS-65U, can be hidden away in the OS.

3. SPOOLER/DESPOOLER UTILITY

A real time-saver for busy systems...

- Designed for hard-disk based systems under OS-65U. Can be used with all 3 printer interfaces now supported by OSI.
- Spooler task intercepts data bound for the printer and writes it out on hard disk for temporary storage.
- While your normal Basic program is still running, the despooler task relays information stored on hard disk to your printer for normal, low-speed printout. You don't have to sit around and wait for the printer to finish so you can get back to work.
- Utility is written in highly efficient, machine language that saves hours of computer time. Requires no rewriting of your Basic programs and causes only a minimal decrease in the thruput of the concurrently running Basic program.
- Makes multiple copies of any output by repeating the process as many times as necessary. Produces sharp, clear, originals everytime.

4. FIG FORTH UNDER OS-65U

With the "extras" that make a difference!

- Supports multi-user systems.
- Supports hard-disk systems.
- Includes a number of helpful utilities and applications.
  - Terminal oriented editor
  - Printer and terminal tools
  - Case command
- With a multi-user system, you have the freedom to use both Fig Forth, a totally extensible language that runs better and faster than basic, and basic, at the same time.
- Compare this with other versions of Fig Forth that run only on floppy-disk, single-user systems. Ours is the only one that runs under OS-65U and gives you the multi-user, hard-disk capabilities.

5. VIDEO ROUTINE

To give your video-based system added dimensions...

- Software that adds valuable capabilities normally found only on separate terminals...and even some nice "extras" you won't normally find.
  - Full color controls for super-effective graphics.
  - User controlled cursor positioning.
  - 24 special control codes.
  - Extensive control of screen parameters.
  - Horizontal and vertical plotting.
  - Easily extensible for your own applications.
- Special "window" feature for multiple imaging, so you can print to separate parts of the screen at different times. Routine will save positions of windows already defined and allows up to 6 windows on the screen at the same time.
- Prints any graphic character you need to create complete charts, tables, diagrams, games, etc.
- Occupies 1¼K of memory under OS-65D.

Unlike the majority of other software vendors, we offer our customers copies of source code (on floppies) for any of our products they've purchased. For a nominal fee of $10, covering our material, postage and handling cost, we'll send you the source code you choose. Simply write in what you want as you fill out the order coupon. If you'd like a catalog of our products, please indicate that on the coupon, too, and we'll be happy to send you one. Also, please include any ideas you may have for future products you'd like to see available from SOFTWARE CONSULTANTS.

SOFTWARE
CONSULTANTS
HARDWORKING *software*

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

	Totals
______DISASSEMBLY MANUAL(S) @ $24.95 ea.	______
______CROSS REFERENCE UTILITY(S) @ $29.95 ea. Disk size 5¼" ______ or 8" ______ OS-65D ______ or OS-65U ______	______
______SPOOLER/DESPOOLER UTILITY(S) @ $59.95 ea. Std. parallel interface ______ Serial interface w/430 board or equivalent ______ Serial interface w/CA10X board ______	______
______VIDEO ROUTINE(S) @ $24.95 ea.	______
______FIG FORTH(S) @ $79.95	______
______SOURCE CODE @ $10.00 ea. For ______	______
______SOFTWARE CONSULTANTS Product Catalog	Free

Don't put it off. ORDER NOW. You'll receive your order postpaid from Memphis in 2-3 wks. Dealer inquiries invited for all products. Please specify disk size and OS for Cross Reference Utility and printer interface for SPOOLER/DESPOOLER Utility.

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.

NAME ______
ADDRESS ______
FIRM ______ TITLE ______
CITY/STATE ______ POSTAL CODE ______

Several other exciting products will be available soon too, so keep us in mind as you plan for your upcoming projects.

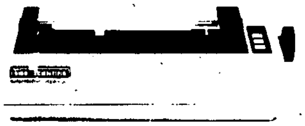
PEEK (65)
The Unofficial OSI Users Journal

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

DELIVER TO:

DBMS, INC.

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



VISA

ACT-5A TERMINAL	Upper lower case, 110-19,200 baud, unique split screen 39x48 character display, pass thru and buffered printer port, full cursor control and edit operations. 8 special function keys, reverse video, blinks and underlines fields, plus many other features	GRAY SCREEN $829.00
		GREEN SCREEN $889.00
MIME-2A TERMINAL	Same features as ACT-5A but EMULATES HAZELTINE 1500 DEC-VT52...SOROC IQ-120	GRAY SCREEN $829.00
		GREEN SCREEN $889.00
CAT MODEM	0-300 Baud, Acoustic, Answer/Originate	$175.00
EPSOM MX-80	Matrix, Upper/Lower case printer	$575.00

master charge
THE INTERBANK CARD

Order yours today and we will ship tomorrow freight collect.