OHIO SCIENTIFIC TECH NEWSLETTER #16
AUGUST 17, 1979

This weeks Tech Letter contains a couple of important items. Enclosed you will find a copy of the letter we are sending to Small System Journal subscribers. The letter explains the change over to KILOBAUD and a second letter provides a coupon for a discount subscription to KILOBAUD. Take a look at the CD 2 + 2 sale, it's a pretty good deal! This week's Tech Letter also includes a fix for the 530 dynamic memory boards, a OS-65U CHANGE to set up the CA-10-X board as device #3, and the source for an OS-65U "SWAP" command. There is also a color flyer of a few new products that should be released in late fall.

OHIO SCIENTIFIC
1333 S. Chillicothe Road • Aurora, Ohio 44202 • (216) 562-3101

Dear 1979 OSI JOURNAL Subscriber:

Regarding your free subscription to our 1979 SMALL SYSTEMS JOURNAL -- the first issues of the JOURNAL were published in 1977. After releasing two 1978 JOURNALS on time, we ran into problems publishing our JOURNALS on a regular schedule.

1979 JOURNALS are now (it started with the May 1979 issue) a regular paid editorial feature of KILOBAUD MICROCOMPUTING -- leading publication in the microcomputer user field.

This move was made in your interest and in the interest of publishing OSI SMALL SYSTEMS JOURNALS on a regular basis. And in your interest -- we are pleased to enclose a letter which entitles you to a subscription to KILOBAUD MICROCOMPUTING at fantastic savings.

The enclosed letter is now included with the warranty card and shipped with every microcomputer that leaves our factory.

Our people are dedicated to producing microcomputer systems with unique capabilities, supported by effective software programs. Our technical people are sound writers. We do our "thing" and do it well. But publishing is not our "thing".

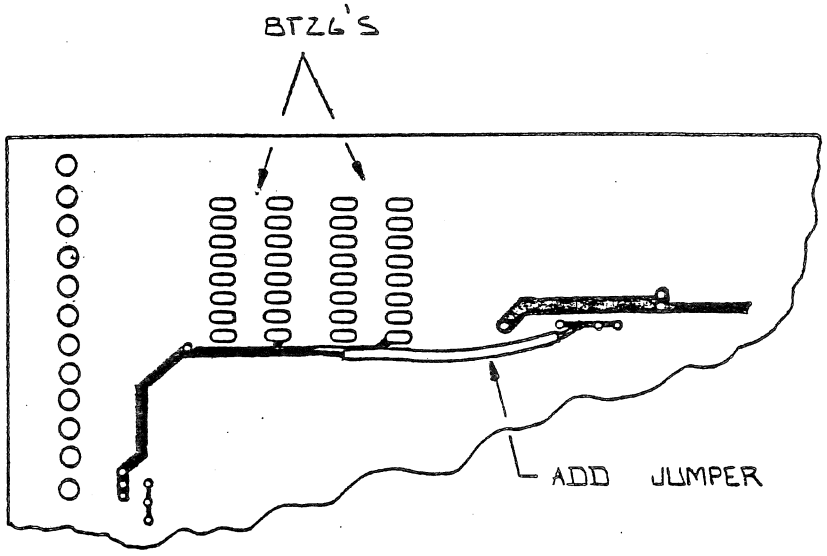
We treasure you as an OSI customer and apologize for the delay in solving our publishing problem. We hope you understand we thank you for your patience. Happy computing!

Sincerely yours,

OHIO SCIENTIFIC, INC.

W. Paul Warren
Coordinator,
Marketing Communications

CM-4 (530) DYNAMIC MEMORY UPDATE



REAR VIEW OF
530 REV C MEMORY BOARD

We have received reports of trouble with the 530 dynamic memory boards operating spuratically. We are currently looking at the trouble. The change shown above decreases ground noise between the 8T26's and the memory array. The jumper should be 18 gauge or heavier wire. The jumper should go from the tape that runs between pin 8 of the two 8T26's to the ground pad near the memory as shown above.

OS-65U And The CA-10-X

The following pages show both the "CHANGE" conversation and assembly listing of a I/O driver for the CA-10-X board. The driver anticipates that the CA-10-X board is strapped for $FBØØ. Once the "CHANGE" has been inserted, one may INPUT or PRINT to the CA-10-X board as device number 3. It is recommended that one use port number 15 (Numbering from 1 to 16) on a fully populated CA-10-16 board. The driver is currently set to deal with port one. However, the following POKE may be used to change ports:

POKE 15610, (Port Number -1) *2

For example, if the 15th port is to be used, then the POKE would be:

POKE 15610, (15-1) *2

OR

POKE 15610, 28

Note that the CA-10-X board must be restrapped @ $FBØØ. The address jumpering would be:

A15 Common To	A15 TRUE
A14 Common To	A14 TRUE
A13 Common To	A13 TRUE
A12 Common To	A12 TRUE
A11 Common To	A11 TRUE
A10 Common To	A10 NOT
A 9 Common To	A 9 TRUE
A 8 Common To	A 8 TRUE

Change for OS-65U to permit the CA-10X board to work as I/O device number 3.

```
RUN "CHANGE", "PASS

DISK CHANGE UTILITY


MODE: HEX(H), DEC(D) ? H
UNIT ? A
ADDRESS OFFSET ? CØØ
ADDRESS ? 3CC3
ØØØØ3CC3   90 ? 90
ØØØØ3CC4   14 ? 1B
ØØØØ3CC5   90 ? .
ADDRESS ? 3CC9
ØØØØ3CC9   AD ? 8A
ØØØØ3CCA   Ø6 ? 48
ØØØØ3CCB   FB ? A2
ØØØØ3CCC   A9 ? 1E
ØØØØ3CCD   FF ? A9
ØØØØ3CCE   8D ? Ø3
ØØØØ3CCF   Ø5 ? 9D
ØØØØ3CDØ   FB ? ØØ
ØØØØ3CD1   A9 ? FB
ØØØØ3CD2   ØØ ? A9
ØØØØ3CD3   8D ? 11
ØØØØ3CD4   98 ? 9D
ØØØØ3CD5 - 2D ? ØØ
ØØØØ3CD6 L 4C ? FB
ØØØØ3CD7   A7 ? CA
ØØØØ3CD8 9 39 ? CA
ØØØØ3CD9   20 ? 10
ØØØØ3CDA   DF ? F2
ØØØØ3CDB < 3C ? 68
ØØØØ3CDC L 4C ? AA
ØØØØ3CDD   1E ? 4C
ØØØØ3CDE 9 39 ? A7
ØØØØ3CDF   A9 ? 39
ØØØØ3CEØ   C8 ? 2Ø
ØØØØ3CE1   8D ? E6
ØØØØ3CE2   B7 ? 3C
ØØØØ3CE3 8 38 ? 4C
ØØØØ3CE4   8E ? 1E
ØØØØ3CE5   B9 ? 39
ØØØØ3CE6 8 38 ? 8E
ØØØØ3CE7   A2 ? F9
ØØØØ3CE8 2 32 ? 3C
ØØØØ3CE9   AD ? AE
ØØØØ3CEA   Ø5 ? FA
ØØØØ3CEB   FB ? 3C
ØØØØ3CEC J 4A ? BD
ØØØØ3CED   BØ ? ØØ
ØØØØ3CEE   16 ? FB
ØØØØ3CEF , 2C ? 4A
ØØØØ3CFØ   B8 ? 9Ø
ØØØØ3CF1 8 38 ? FA
ØØØØ3CF2 P 5Ø ? BD
ØØØØ3CF3   F5 ? Ø1
```

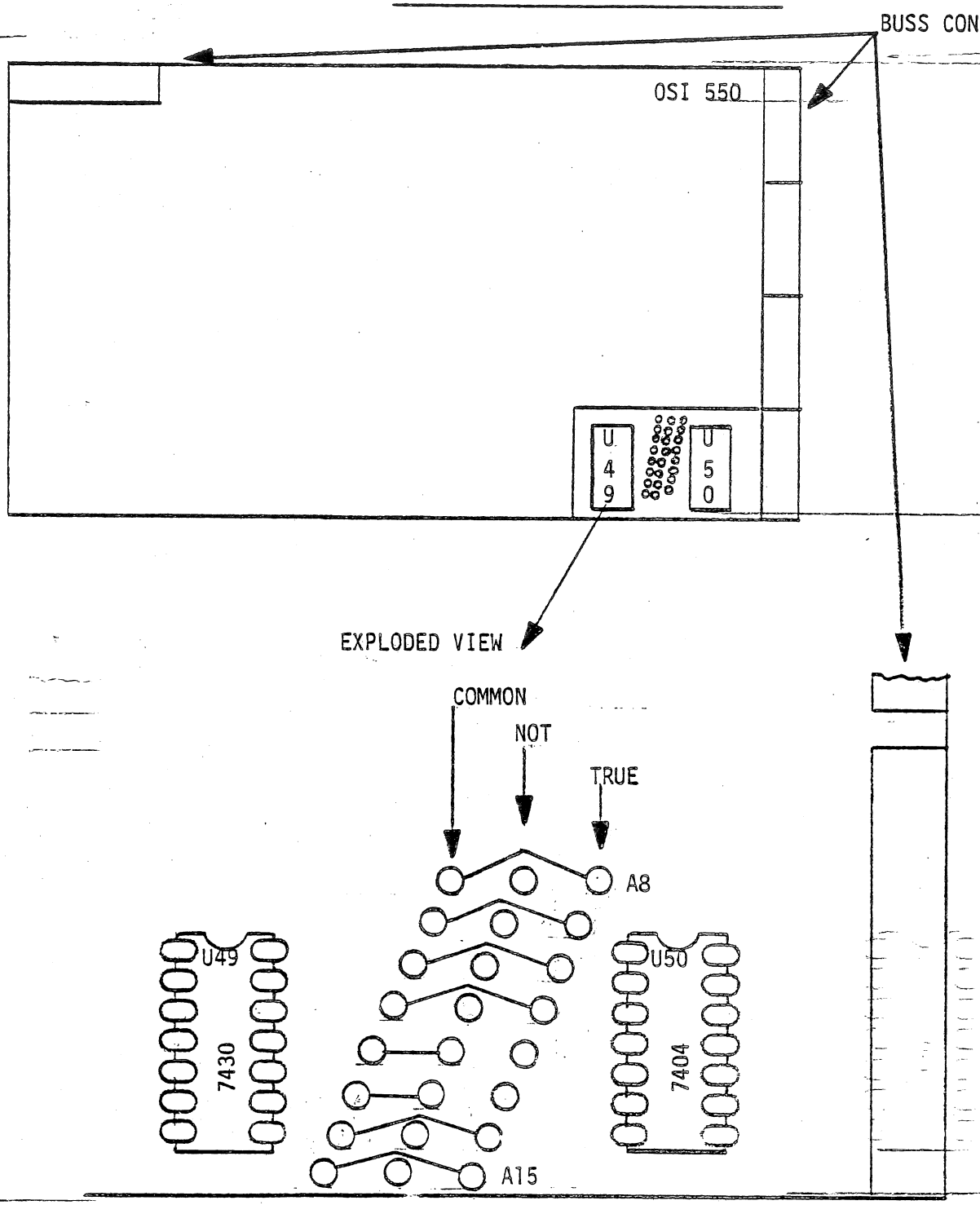
```
0000 3CF4   A9 ? FB
0000 3CF5   01 ? AE
0000 3CF6   20 ? F9


0000 3CF7   B8 ? 3C
0000 3CF8 > 3E ? 60
0000 3CF9   CA ? 00
0000 3CFA   D0 ? 00
0000 3CFB   ED ? .
ADDRESS ? 3D51
0000 3D51   90 ? 90
0000 3D52   02 ? 51
0000 3D53   90 ? 90
0000 3D54   10 ? 00
0000 3D55   AD ? 20
0000 3D56   06 ? 5B
0000 3D57   FB ? 3D
0000 3D58   A9 ? 4C
0000 3D59   FF ? 2D
0000 3D5A   8D ? 3A
0000 3D5B   05 ? 8E
0000 3D5C   FB ? F9
0000 3D5D   A9 ? 3C
0000 3D5E   00 ? AE
0000 3D5F   8D ? EA
0000 3D60   AD ? 3C
0000 3D61 - 2D ? BD
0000 3D62 L 4C ? 00
0000 3D63   A7 ? FB
0000 3D64 9 39 ? 4A
0000 3D65   20 ? 4A
0000 3D66 K 6B ? 90
0000 3D67 = 3D ? F9
0000 3D68 L 4C ? AD
0000 3D69 - 2D ? B6
0000 3D6A : 3A ? 38
0000 3D6B   AD ? 9D
0000 3D6C   05 ? 01
0000 3D6D   FB ? FB
0000 3D6E   10 ? AE
0000 3D6F   FB ? F9
0000 3D70   AD ? 3C
0000 3D71   B6 ? 60
0000 3D72 8 38 ? X

OK
CLOSE

OK
```

```
A*Q01
A*RA

.A

1000 0000          ;         CA-10-X DRIVER ROUTINE
1010 0000          ;       (C) 1979 BY OHIO SCIENTIFIC, INC.
1020 0000          ;        ALL RIGHTS RESERVED
1030 0000          ;        WRITTEN BY R. WHITESEL 8/79
1040 0000          ;
1050 0000          ;
1060 0000          ; CODE LOCATED OVERTOP 430 DRIVER ROUTINES
1070 0000          ; PROVIDES OUTPUT & INPUT TO/FROM THE 550
1080 0000          ; BOARD.  550 BOARD MUST BE ADDRESSED AT
1090 0000          ; $FBXX AND UP.  THE PORT THAT INPUT AND OUTPUT
1100 0000          ; WILL DEAL WITH IS DETERMINED BY THE CONTENTS
1110 0000          ; OF PORTNM.  PORTNM IS SET TO :
1120 0000          ;
1130 0000          ; (DESIRED PORT - 1) * 2
1140 0000          ;
1150 0000          ; E.G.  DESIRED PORT IS PORT NUMBER 3 THEN,
1160 0000          ; (PORTNM) = ( 3 - 1 ) * 2 = 2 * 2 = 4
1170 0000          ;
1180 0000          ;
1190 0000          ; EQUATES :
1200 0000          ;
1210 0000          ; GLOBAL :
1220 0000          ;
1230 0000          INPXIT=$391E            ; I/O DIST INPUT EXIT POINT
1240 0000          OUTXIT=$3A2D            ; I/O DIST OUTPUT EXIT POINT
1250 0000          INPRET=$39A7            ; INIT RETURN POINT
1260 0000          RAM000=$3DA4            ; RAM OUTPUT INIT BCC POINT
1270 0000          CHRBYT=$38B6            ; (CHR TO BE OUTPUT)
1280 0000          ;
1290 0000          ; CONSTANTS :
1300 0000          ;
1310 0000          BAS550=$FB00            ; BASE ADDR OF 550 SET IN PLACE OF 430
1320 0000          ;
1330 0000          ;
1340 0000          ;
1350 3CC3                  *=$3CC3
1360 3CC3          ;
1370 3CC3 901B             BCC INP550      ; CHG BCC TO POINT TO NEW CODE
1380 3CC5          ;
1390 3CC5          ;
1400 3CC9                  *=$3CC9
1410 3CC9          ;
1420 3CC9          ; 550 INIT CODE
```

```
1430 3CC9          ;
1440 3CC9 8A       INT550  TXA             ; SAV X-REG
1450 3CCA          ;
1460 3CCA 48               PHA
1470 3CCB A21E             LDX #2*15       ; 15 PORTS MAX * 2
1480 3CCD A903     INIT01  LDA #$03        ; MASTER RESET CODE
1490 3CCF          ;
1500 3CCF 9D00FB           STA BA8550,X    ; MASTER RESET ACIA
1510 3CD2 A911             LDA #$11        ; DEFINE ACIA 2 STOP, 8 DATA, NO PARITY
1520 3CD4 9D00FB           STA BA8550,X
1530 3CD7 CA               DEX             ; POINT TO NXT ACIA CNTRL REG
1540 3CD8 CA               DEX
1550 3CD9 10F2             BPL INIT01      ; LOOP TILL ALL INIT'D
1560 3CDB 68               PLA             ; RESTORE X-REG
1570 3CDC AA               TAX
1580 3CDD 4CA739           JMP INPRET      ; INIT RETURN POINT
1590 3CE0          ;
1600 3CE0          ;
1610 3CE0          ; 550 INPUT ROUTINE
1620 3CE0          ;
1630 3CE0 20E63C INP550    JSR PORTIN      ; GET CHR
1640 3CE3          ;
1650 3CE3 4C1E39           JMP INPXIT      ; RETURN TO CALLER
1660 3CE6 8EF93C PORTIN    STX X:TEMP      ; SAV X-REG
1670 3CE9          ;
1680 3CE9 AEFA3C           LDX PORTNM      ; GET INDEX TO PORT
1690 3CEC BD00FB INWAIT    LDA BA8550,X    ; ANY INPUT ?
1700 3CEF          ;
1710 3CEF 4A               LSR A
1720 3CF0 90FA             BCC INWAIT      ; NO- LOOP TILL THERE IS !
1730 3CF2 BD01FB           LDA BA8550+1,X  ; GET CHR INPUT
1740 3CF5 AEF93C           LDX X:TEMP      ; RESTORE X-REG
1750 3CF8 60               RTS
1760 3CF9          ;
1770 3CF9          ;
1780 3CF9 00       X:TEMP  .BYTE $00       ; TEMP STORAGE FOR X-REG
1790 3CFA          ;
1800 3CFA 00       PORTNM  .BYTE $00       ; (PORT INDEX)
1810 3CFB          ;
1820 3CFB          ;
1830 3CFB          ;
1840 3D51                  *=$3D51
1850 3D51          ;
1860 3D51 9051     UART00  BCC RAM000      ; POINT 430 OUT INIT TO MEM OUT INIT
1870 3D53 9000             BCC OUT550      ; POINT TO NEW I/O ADDR
1880 3D55          ;
1890 3D55          ;
1900 3D55          ;
1910 3D55                  *=$3D55
1920 3D55          ;
1930 3D55 205B3D OUT550    JSR PORTOU      ; OUPUT CHR IN A
1940 3D58          ;
```

```
1950 3D5B 4C2D3A          JMP OUTXIT      ; CARRY ON
1960 3D5B 8EF93C PORTOU   STX X:TEMP      ; SAV X-REG
1970 3D5E                 ;
1980 3D5E AEFA3C          LDX PORTNM      ; GET INDEX TO PORT
1990 3D61 BD00FB OUWAIT   LDA BA8550,X    ; READY TO TAKE THIS CHR ?
2000 3D64                 ;
2010 3D64 4A              LSR A
2020 3D65 4A              LSR A
2030 3D66 90F9            BCC OUWAIT      ; NO- LOOP TILL ACIA READY
2040 3D68 ADB638          LDA CHRBYT      ; GET CHR TO BE OUTPUT
2050 3D6B 9D01FB          STA BA8550+1,X  ; STUFF CHR INTO ACIA
2060 3D6E AEF93C          LDX X:TEMP      ; RESTORE X-REG
2070 3D71 60              RTS

.80
```

CA-10-X (550) ADDRESS STRAPPING



To move CA-10-X board to reside @ the 430 board address:

Cut Foil Between	A13 Common & A13 Not
Cut Foil Between	A12 Common & A12 Not
Cut Foil Between	A10 Common & A10 True
Jumper Between	A13 Common & A13 True
Jumper Between	A12 Common & A12 True
Jumper Between	A10 Common & A10 Not

OS-65U SWAP COMMAND

The "SWAP" command code provides OS-65U with the ability to "SWAP" variable values. This command is very useful in sort programs. For example, if a BASIC program must compare to variables (Strings in this case) and then exchange their values, the folling steps are taken:

1) If comparision of A$(X) and B$(X)

2) Perform Swap by:
   T$=B$(X): B$(X)=A$(X): A$(X)=T$

It is easy to see that this can really slow things up when one starts performing this test on a 1000 string variables. However, with the "SWAP" command the sequence would be:

1) If comparision of A$(X) and B$(X)

2) Perform SWAP by:
   SWAP A$(X);B$(X)

The "SWAP" command code resides in front of the BASIC program and uses the subroutine shown to the left to enable and disable the command.

```
100 REM                OS-65U SWAP COMMAND
110 REM             (C) 1979 BY OHIO SCIENTIFIC, INC.
120 REM                ALL RIGHTS RESERVED
130 REM             WRITTEN BY R. WHITESEL 8/79
140 REM
150 REM
160 REM        COMMAND SYNTAX :
170 REM
180 REM        SWAP VARIABLE NAME ; VARIABLE NAME <CR>
190 REM                          OR
200 REM        SWAP VARIABLE NAME ; VARIABLE NAME ; ETC; ETC <CR>
210 REM
220 REM
230 REM
240 REM TO USE THIS ROUTINE THE MACHINE CODE FOR THE SWAP
250 REM COMMAND MUST BE PLACED IN FRONT OF THE BASIC PROGRAM.
260 REM THIS CAN BE DONE BY ENTERING A "NEW 100 <CR>". FOLLOWING
270 REM THIS, ONE SHOULD HIT THE RESET BUTTON AND ENTER THE MONITOR.
280 REM PROCEDE BY ENTERING THE MACHINE CODE INTO $6000
290 REM UP. WHEN THE MACHINE CODE HAS BEEN ENTERED, TYPE :
300 REM L 012E 0000 R
310 REM G
320 REM NEW 100 <CR>
330 REM THIS WILL RESTART BASIC AND PERMIT ONE TO ENTER THIS BASIC
340 REM PROGRAM.  NOW WHEN ONE DESIRES THE SWAP COMMAND, ONE
350 REM NEED ONLY LOAD THIS PROGRAM INTO MEMORY AND ENTER THE
360 REM PROGRAM WHICH WILL USE THE SWAP COMMAND IN FRONT OF
370 REM THE SUBROUTINE AT 63700 AND UP.  FOR EXAMPLE
380 REM
390 GOSUB 63700: REM ENABLE SWAP COMMAND
400 FOR X=0 TO 9: A(X)=X-9: B(X)=X
410 PRINT "BEFORE SWAP A(";X;")=";A(X),"B(";X;")=";B(X)
420 SWAP A(X);B(X)
430 PRINT "AFTER SWAP A(";X;")=";A(X),"B(";X;")=";B(X)
440 PRINT: NEXT
450 GOSUB 63900: REM DISABLE SWAP COMMAND
460 END
470 REM
480 REM
490 REM
498 REM
499 REM
500 REM
63700 REM ENABLE "SWAP" COMMAND ENTRY POINT
63710 REM
63720 REM SAVE RESERVED WORD "NULL"
63730 REM
63740 FOR X=9025 TO 9028: RS(X-9025)=PEEK(X): NEXT
63750 REM
63760 REM SAVE "NULL" DISPATCH ADDRESS
63770 REM
63780 DISP(0)=PEEK(8738): DISP(1)=PEEK(8739)
63790 REM
63800 REM POKE RESERVED WORD "SWAP" INTO PLACE
63810 REM
63820 POKE 9025,ASC("S"): POKE 9026,ASC("W")
63830 POKE 9027,ASC("A"): POKE 9028,ASC("P")+128
63840 REM
63850 REM POKE "SWAP" DISPATCH ADDRESS INTO PLACE
```

```
63860 REM
63870 POKE 8738,00: POKE 8739,96
63880 RETURN
63890 REM
63900 REM DISABLE "SWAP" COMMAND ENTRY POINT
63910 REM
63920 REM RESTORE RESRVED WORD "NULL"
63930 REM
63940 FOR X=9025 TO 9028: POKE X,RS(X-9025): NEXT
63950 REM
63960 REM RESTORE "NULL" COMMAND DISPATCH ADDRESS
63970 REM
63980 POKE 8738,DISP(0): POKE 8739,DISP(1)
63990 RETURN
```

```
A*RA

.A

1000 0000          ;          SWAP OVERLAY CODE FOR OS-65U V1.1
1010 0000          ;          COPYRIGHT 1979 BY OHIO SCIENTIFIC, INC.
1020 0000          ;          ALL RIGHTS RESERVED
1030 0000          ;          IMPLEMENTED BY R. WHITESEL 8/79
1040 0000          ;
1050 0000          ;          THIS VERSION RESIDES IN FRONT OF
1060 0000          ;          A BASIC PROGRAM @ $6000...
1070 0000          ;
1080 0000          ; SWAP PERMITS THE VALUE OF TWO VARIABLES TO BE
1090 0000          ; SWAPPED WITHOUT BUILDING ANY TEMPS.
1100 0000          ;
1110 0000          ;
1120 0000          ; SYNTAX :
1130 0000          ;
1140 0000          ;        SWAP VAR EXP, VAR EXP, ETC, ETC <CR>
1150 0000          ;
1160 0000          ;
1170 0000          ; EQUATES :
1180 0000          ;
1190 0000          ; BASIC ROUTINES :
1200 0000          ;
1210 0000          PTRGET=@7456     ; RETURNS PNTR TO VAR
1220 0000          CHRGOT=@0306     ; RTS'S W CHR TYPE @ (TXTPTR)
1230 0000          SNERR:=@7036     ; SYNTAX ERROR OUTPUT ROUTINE
1240 0000          TMERR:=@6310     ; TYPE MISMATCH ERROUT ROUTINE
1250 0000          CHRCHK=@7025     ; CHKS CHR @ (TXTPTR) SAME AS (A). IF / SN ERR
1260 0000          ;
1270 0000          ;
1280 0000          ; BASIC POINTERS :
1290 0000          ;
1300 0000          VARPNT=@0224     ; (PNTS TO VAR VAL AFTER CALLING PTRGET)
1310 0000          VARNAM=@0222     ; (VARIABLE NAME AFTER CALL TO PTRGET)
1320 0000          ;
1330 0000          ;
1340 0000          ; LOCAL POINTERS :
1350 0000          ;
1360 0000          FRMPNT=$00ED     ; (TMP PNTR TO ONE OF VARS TO BE SWAPPED)
1370 0000          VARTYP=$00EF     ; (TMP FOR VARIABLE TYPE / 5=FLT'N, 3=$, 2=INT)
1380 0000          ;
1390 0000          ;
1400 0000          ;
1410 0000          ;
1420 6000                  *=$6000      ; RESIDES IN FRONT OF BASIC PROGRAM
```

```
1430 6000          ;
1440 6000          ;
1450 6000 F03A    SWAPER  BEQ SN:ERR        ; NO ARG FOLLOWING IS AN SN ERROR
1460 6002          ;
1470 6002 202E0F  SWAP01  JSR PTRGET        ; GET PTR TO FIRST VAR VAL
1480 6005          ;
1490 6005 204560          JSR GETTYP        ; GET TYPE OF 1 ST VAR
1500 6008 85EF            STA VARTYP        ; SAV IT FOR CMP LATER
1510 600A A594            LDA VARPNT        ; SAV PTR TO FIRST VAR VAL
1520 600C 85ED            STA FRMPNT
1530 600E A595            LDA VARPNT+1
1540 6010 85EE            STA FRMPNT+1
1550 6012 205660          JSR ISSEMI        ; MUST HAV A ';' OR SN ERR
1560 6015 F025            BEQ SN:ERR        ; ';' <CR>' IS SN ERROR
1570 6017 202E0F          JSR PTRGET        ; GET PTR TO 2 ND VAR VAL
1580 601A 204560          JSR GETTYP        ; GET TYPE OF 2 ND VAR
1590 601D C5EF            CMP VARTYP        ; SAME TYPE OF VAR AS FIRST ?
1600 601F D01E            BNE TM:ERR        ; NO- TYPE MISMATCH ERROR
1610 6021 AA              TAX               ; VAR TYPE IS ALSO '# BYTES TO MOV' CNT
1620 6022 A000            LDY #$00
1630 6024 B194    SWAPEM  LDA (VARPNT),Y    ; SWAP VAR VALS
1640 6026          ;
1650 6026 48              PHA               ; SAV ON STK
1660 6027 B1ED            LDA (FRMPNT),Y    ; GET CHR FRM SEC VAR
1670 6029 9194            STA (VARPNT),Y    ; STUFF IT
1680 602B 68              PLA               ; GET CHR FROM OTHER VAR BAC
1690 602C 91ED            STA (FRMPNT),Y    ; STUFF IT
1700 602E C8              INY               ; PNT TO NXT
1710 602F CA              DEX               ; MOV ALL OF THE VAR VALS ?
1720 6030 D0F2            BNE SWAPEM        ; NO- CONT SWAP
1730 6032 20C600          JSR CHRGOT        ; FIND OUT IF MORE TO DO
1740 6035 F00B            BEQ OKEXIT        ; NO- RTS W/O ERROR
1750 6037 205660          JSR ISSEMI        ; A ';' MUST FOLLOW OR SN ERROR
1760 603A D0C6            BNE SWAP01        ; MUST BE MORE TO DO
1770 603C 4C1E0E  SN:ERR  JMP SNERR:        ; SN ERROR EXIT
1780 603F          ;
1790 603F 4CC80C  TM:ERR  JMP TMERR:        ; TYPE MISMATCH ERROR EXIT
1800 6042          ;
1810 6042 A900    OKEXIT  LDA #$00          ; EXIT W/O ERROR
1820 6044          ;
1830 6044 60              RTS
1840 6045          ;
1850 6045          ;
1860 6045          ; GETTYP RETURNS (A) = VAR TYPE
1870 6045          ; (A) = 5 = FLOATING POINT VARIABLE
1880 6045          ; (A) = 3 = STRING VARIABLE
1890 6045          ; (A) = 2 = INTEGER VARIABLE
1900 6045          ;
1910 6045          ;
1920 6045 A905    GETTYP  LDA #$05          ; FLT'N PNT DEFAULT
1930 6047          ;
1940 6047 2492            BIT VARNAM        ; AN INT?
```

```
1950 6049 1004           BPL CHKNXT      ; NO- CONT
1960 604B A902           LDA ##02        ; YES- AN INT RTS W (A) = 2
1970 604D D006           BNE TYPRTS
1980 604F 2493   CHKNXT  BIT VARNAM+1    ; STRING OR FLOATING POINT ?
1990 6051        ;
2000 6051 1002           BPL TYPRTS      ; FLOATING POINT SO RTS W (A) = 5
2010 6053 A903           LDA ##03        ; STRING SO RTS W (A) = 3
2020 6055 60     TYPRTS  RTS
2030 6056        ;
2040 6056        ;
2050 6056        ; ISSEMI CHKS CHR @ (TXTPTR) IS A ';' IF / SN ERR
2060 6056        ; ELSE JMP CHRGET
2070 6056        ;
2080 6056 A93B   ISSEMI  LDA #';         ; CHR @ (TXTPTR) A ';' ?
2090 6058        ;
2100 6058 4C150E         JMP CHRCHK      ; GO FIND OUT
```

.SO