

OHIO SCIENTIFIC TECH NEWSLETTER #17

August 24, 1979

Copyright 1979
OHIO SCIENTIFIC, INC.
All Rights Reserved.

TECH NEWSLETTERS

OSI Sales Staff is currently supplying one copy of each newsletter to every authorized dealer and one copy for each of their associates. If extra copies are needed for customers, etc., the cost is 25¢ for each copy. This is to defray costs of copying and postage.

ORDER CANCELLATION POLICY

Ohio Scientific Inc. instituted an order cancellation policy in November of 1978. This policy was noted in the November 1978 Price List and also discussed at the November dealer meeting.

The cancellation policy is as follows:

- 1) If an order is cancelled within 90 days, the total of the order is doubled and then deducted from the current monthly total.
- 2) If an order is cancelled after 90 days there is no penalty.
- 3) If an order, after being shipped, is refused, the dealer is billed for a 10% restocking charge and item #1 is followed (this applies to valid back-orders less than 90 days old, UPS COD only).
- 4) If an order, after being shipped, is refused the dealer is billed for a 10% restocking fee (this applies to orders older than 90 days, UPS COD).

Order cancellations are acceptable only in writing. The cancellation should include the dealers purchase order number, our invoice number if available, the dealer's name and a list of items to be cancelled. This should be sent by the dealer to their respective Sales Coordinators.

Cancellations are accepted the date they are received at our office. This means if we receive a cancellation the day after the shipment is made, the dealer will still be held responsible.

This is a clarification of our policies as stated on page 4 under the topic of "Discount Levels" in the November 1978 price list and page 3 of the January 1979 Dealership Policy & Application.

SALES NOTES

ON DOUBLE SIDED DRIVES

Single sided drives have a felt pad to counter the tape head pressure. Double sided drives lack this felt pad and instead use the opposing tape head. Due to this, the double sided drives may "wear" standard diskettes out rapidly. However, the use of a properly lubricated diskette that is certified for double sided operation should greatly increase the life of the diskette. Even if one plans to use only one side, the excessive wear may occur on a standard diskette. This is due to the fact that the second head is still being used to support the diskette. IBM or DYSAN diskettes certified for double sided operation are considered in the industry to be of high quality. The storage media for a system in a business environment should be of the highest possible quality to insure long term reliability instead of short term savings.

ON OS-CP/M

LIFEBOAT, a software house, has entered into contract with Ohio Scientific, Inc. They will be updating OS-CP/M's FORTRAN, COBALT and MBASIC. It is also our understanding that many other programs will be available from them on OSI compatible diskettes. Further details will be announced when available.

A note from one of our dealers, that we would like to share with you.

TO: CANDY
FROM: KATHY *Kathy*
RE: FEISTY LITTLE 4P

10 AUG 79

CANDY, THIS STORY SHOULD MAKE SOMEBODY THERE HAPPY.

ONE OF OUR CUSTOMERS HAD A SEVERE HOUSE FIRE. EVERYTHING EITHER BURNED OR SUFFERED SMOKE DAMAGE. THE FIRE WAS HOT ENOUGH TO MELT THE KEYS OF THE ENTIRE KEYBOARD OF HIS C2-4P. THINKING IT WAS KAPUT, HE GAVE IT TO ONE OF THE CLEAN-UP CREW CLEARING OUT HIS HOUSE. THE CLEANUP MAN BROUGHT IT IN TO US TO SEE WHAT COULD BE DONE WITH IT, AND IT WORKS PERFECTLY! NO DAMAGE EXCEPT A COMPLETELY MELTED KEYBOARD.

THE GENTLEMAN WOULD LIKE TO ORDER A NEW SET OF KEY TOPS FOR HIS KEYBOARD. I DO NOT KNOW THE PRICES BUT HAVE INCLUDED THEM ON THE ATTACHED PO.

SUPERBOARD II / C-IP MONITOR ROUTINES

BASIC SUPPORT ROUTINES

ADDRESS	FUNCTION	USE
\$FFFE	IRQ VECTOR	6502 IRQ TO \$01C0
\$FFFC	RESET VECTOR	6502 RESET TO \$FF00
\$FFFA	NMI VECTOR	6502 NMI TO \$0130
\$FFF7	SAVE CMD JMP (\$0220)	SETS I/O TO CASSETTE OUTPUT VIA A JMP (\$0220)
\$FFF4	LOAD CMD JMP (\$021E)	SETS I/O TO CASSETTE INPUT VIA A JMP (\$021E)
\$FFF1	CNTRL-C CHECK JMP (\$021C)	RETURNS \$03 IF CNTRL-C IS TYPED
\$FFEE	BASIC OUTPUT JMP (\$021A)	OUTPUT CHR IN A TO SCREEN, OR SCREEN & CASSETTE
\$FFEB	BASIC INPUT JMP (\$0218)	INPUT CHR INTO ACCUMULATOR FROM KEYBOARD OR CASSETTE
\$FF00	COLD START "D/C/W/M"	RESET ("BREAK") ENTRY POINT

SUPERBOARD II - C-IP MONITOR ENTRY POINTS

65K POLLED KEYBOARD ROUTINE

ADDRESS	FUNCTION	USE
\$FD00	KEYBOARD INPUT	RETURNS ASCII OF CHR ENTERED AT KEYBOARD. HOLDS UNTIL A KEY IS PRESSED.

SUPERBOARD II / C-IP MONITOR ENTRY POINTS

65VK MONITOR

<u>ADDRESS</u>	<u>FUNCTION</u>	<u>USE</u>
\$FEE9	INPUT ROUTINE	RETURN CHR IN A FROM KEY-BOARD OR TAPE
\$FEDA	ROLL	MOV LSD IN ACC TO LSD IN 2 BYTE NUMBER @\$00FC+X X MUST BE SET BEFORE CALLING
\$FECA	DIGIT	OUTPUT LSD (IN HEX) IN ACC TO SCREEN @ \$D0C6+Y Y MUST BE SET BEFORE CALLING
\$FEB0	OUI	OUTPUT(X) BYTES STARTING @\$00FC+X TO SCREEN STARTING @\$D0C6+Y X AND Y MUST BE SET BEFORE JSR X DECREASES, Y INCREASES
\$FEAC	OUTPUT	OUTPUT ADDRESS DATA IN MONITOR FORMAT ADDRESS = (\$FF,\$FE) DATA = (\$FC)
\$FE93	LEGAL	CONVERT CHR IN ACC FROM ASCII TO HEX IF CHR IS "0" THROUGH "9" OR "A" THROUGH "F". ELSE RETURNS (ACC) = \$80
\$FE80	OTHER	INPUT CHR FROM TAPE AND RETURN CHR IN ACC
\$FE00	START	MONITOR START ENTRY POINT

SUPERBOARD II / C-IP MONITOR ENTRY POINTS

MINI-FLOPPY BOOTSTRAP ROUTINE

ADDRESS	FUNCTION	USE
\$FCCF	KEYBOARD LOAD (LDAKBD)	LOADS COMPLEMENT OF KEYBOARD AND RETURN IT IN ACC
\$FCC6	KEYBOARD LOAD (LDXKBD)	SAME AS ABOVE EXCEPT RETURNED IN X-REG WITH CONDITION CODES SET ON THE CONTENTS OF X-REG. ACC IS PRESERVED
\$FCBE	KEYBOARD WRITE (STAKBD)	WRITE COMPLEMENT OF ACC INTO KEYBOARD
\$FCB1	ACIA OUT (OUTCH)	OUTPUTS CHR IN ACC TO TAPE
\$FCA6	SERIAL INIT (SERINT)	INITIALIZES CASSETTE PORT
\$FC9C	READ BYTE (READ)	READS A BYTE FROM DISC. BYTE RETURNED IN ACC
\$FC91	TIME DELAY (MRT)	TIME DELAY SUBROUTINE Y & X LOST, Y & X RETURNED = 0, TIME DELAY = 1.25 Ms * X
\$FC8B	UNLOAD HEAD (UNLOAD)	UNLOADS FLOPPY HEAD
\$FC06	MANUAL BOOT (LOAD0)	LOADS TRACK ZERO INTO \$2200 UP AND RETURNS TO MONITOR
\$FC00	AUTO BOOT	BOOTS OPERATING SYSTEM IN AND EXECUTES IT

FIX TO PERMIT OS-CP/M TO BE RUN IN A SINGLE USER MODE
ON A SYSTEM MODIFIED FOR OS-65U LEVEL III

- 1.) Boot the CP/M Escort Maintenance Diskette
Insert Fortran or Cobalt Diskette into drive A then
type: CALL 4580=02,1 <CR >
- 2.) Enter the machine language monitor by typing "RE M"
- 3.) Enter the following changes: L4E4A 00 00 00 R
L4E48 00 00R
L4E4E FFR
L4E50 CFR
L4E5B 00 00R
L4E5D 00 00 00R
L4E7A 00 00R
L4E7C 00 00 00R
- 4.) Return to the DOS Kernel by typing:
"L012C 04 AF 2A51R"
"G"
- 5.) Save the changed track on disk by typing:
SAVE 02,1=4580/D <CR >

OS-65U - SETTING THE RETRY COUNT

The POKES below may be used to change the number of times OS-65U performs a retry when a disc error is encountered. This could be useful when attempting to recover data from a "wiped" disc or diskette.

<u>Function</u>	<u>ADDRESS</u>	<u>CURRENT RETRY COUNT</u>
Seek Retry Count	\$3164 (12644)	2 (0=No Retries)
Read Retry Count	\$31FC (12796)	6 (0=No Retries)
Write Retry Count	\$32CA (13002)	2 (0=No Retries)

OS-DMS REPORT WRITERS

NOT in
V1.3

THE FIX FOR STAT AND STAT 2 CORRECTS A VERY RARE
PROBLEM IN KEY FILE ACCESS.

STAT FIX

- 1) LOAD "STAT" <CR>
- 2) CHANGE LINE 35190 TO:
35190 INDEX <8>=42: INPUT %8, LABEL\$
- 3) SAVE "STAT", "PASS" <CR>

ORIGINAL!

STAT2 FIX

- 1) LOAD "STAT2" <CR>
- 2) CHANGE LINE 1690 TO:
1690 INDEX <8>=42: INPUT %8, LABEL\$
- 3) SAVE "STAT2", "PASS" <CR>

ORIGINAL!

MOVING MACHINE CODE INTO OS-65U

There are two places that a user machine code routine can reside at in OS-65U. The first place is above BASIC'S workspace. The procedure to do this is something like:

- 1) Set BASIC's upper memory limit to just below where the machine code will reside.
- 2) Using POKES, POKE the machine code into place.

Setting the system up as described above has several disadvantages and are listed below:

- 1) Poking machine code into place is a slow process.
- 2) The workspace must be shortened for all programs, even if a particular program does not require the use of the machine code routine.
- 3) If more than one machine code routine is required, a large space must be allocated or the routine must be POKEd.

The second method for using machine code provides a much "cleaner" implementation. This method is described in detail in the following paragraphs.

Before actually moving the machine code into OS-65U, an understanding of where it will be located is essential. The "NEW" command in OS-65U provides a easy means of allocating space for machine code. Diagram 1 shows how OS-65U's workspace is normally set up. Diagram 2 shows OS-65U's workspace after a "NEW 256" has been typed.

As indicated by the two Diagrams (1 & 2), the start of BASIC's workspace has been offset by the number of bytes specified in the NEW statement. This function in itself is not very useful. However, one important fact remains. When a "SAVE" command is executed, memory transfers to disc always start at \$6000. What this means is that the "dead space" allocated by the "NEW XXXX" command is SAVED to disc in front of the BASIC program. So from our example, one can see that when the BASIC program is LOADED, the allocated "dead space" is also brought into memory. In simpler terms, LOADS and SAVES always start transfers to or from memory at \$6000. So what we have is a means of storing machine code in front of a BASIC program such that the machine code "rides" in and out with the BASIC program.

One can see that this method provides a great deal of flexibility. One may create a BASIC program with the desired machine code in front. Then any program needing that particular machine code routine can easily be created. The steps are:

- 1) LOAD the BASIC file containing the machine code.
- 2) Enter The BASIC program.
- 3) SAVE the BASIC program away.

GENERATING MACHINE CODE

To assemble the machine code routines one should use the assembler under OS-65D V3.1 or the WPL-B word processor. The better choice is the WPL-B as it incorporates a full line EDITor, macro CHANGE and FIND commands, plus move and transfer commands. The machine code should be assembled to RUN at \$6000 and up. The assembled machine code must be Saved to the WPL-B or OS-65D V3.1 diskette. Since there exists no assembler under OS-65U and OS-65U can not read WPLB or OS-65D type diskettes, two utility programs have been provided. "LOAD32" and "LOAD48"(for 32K and 48K machines respectively) provide a means of calling machine code into OS-65U. When "RUN" the utilities "come up" in the OS-65D kernel mode i.e. the familiar "A*" is output. At this point the machine code may be "called" into OS-65U. The steps below give the exact sequence.

- 1) Type:
RUN "LOAD32", "PASS" <CR> for a 32K machine or for a 48K machine RUN "LOAD48", "PASS" <CR>
- 2) Call the machine code into place by entering:
A*CXXX=YY,Z

Where XXXX is the address the machine code is to be called into (normally \$6000), YY is the track number and Z is the sector number.

- 3) Warm start OS-65U by entering:
A*GXXX

Where XXXX=7E12 for a 32K machine and BE12 for a 48K machine.

- 4) Now enter:
New XXXX <CR>

Where XXXX stands for the number of bytes to be allocated for the machine code routines plus one.

E.G. machine code runs from \$6000 to \$60FF. The statement then would be "NEW 256", $((\$60FF+1)-\$6000) = \$6100 - \$6000 = \$100 = 256$

- 5) Now enter the BASIC program to be used in conjunction with the machine code.
- 6) SAVE the BASIC program to disc.

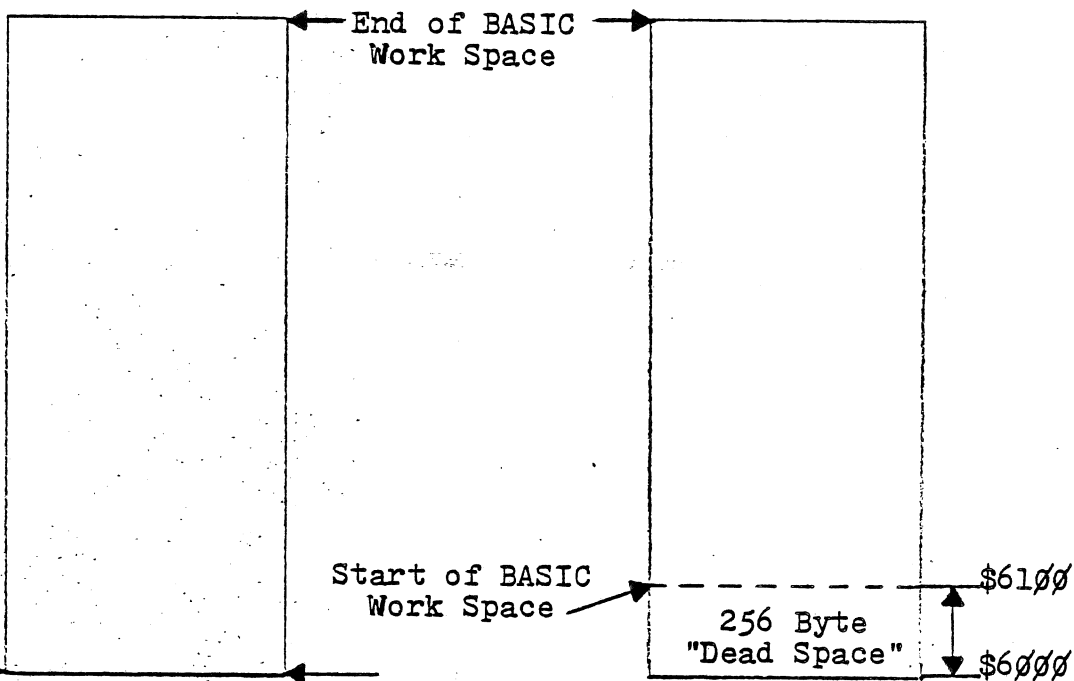
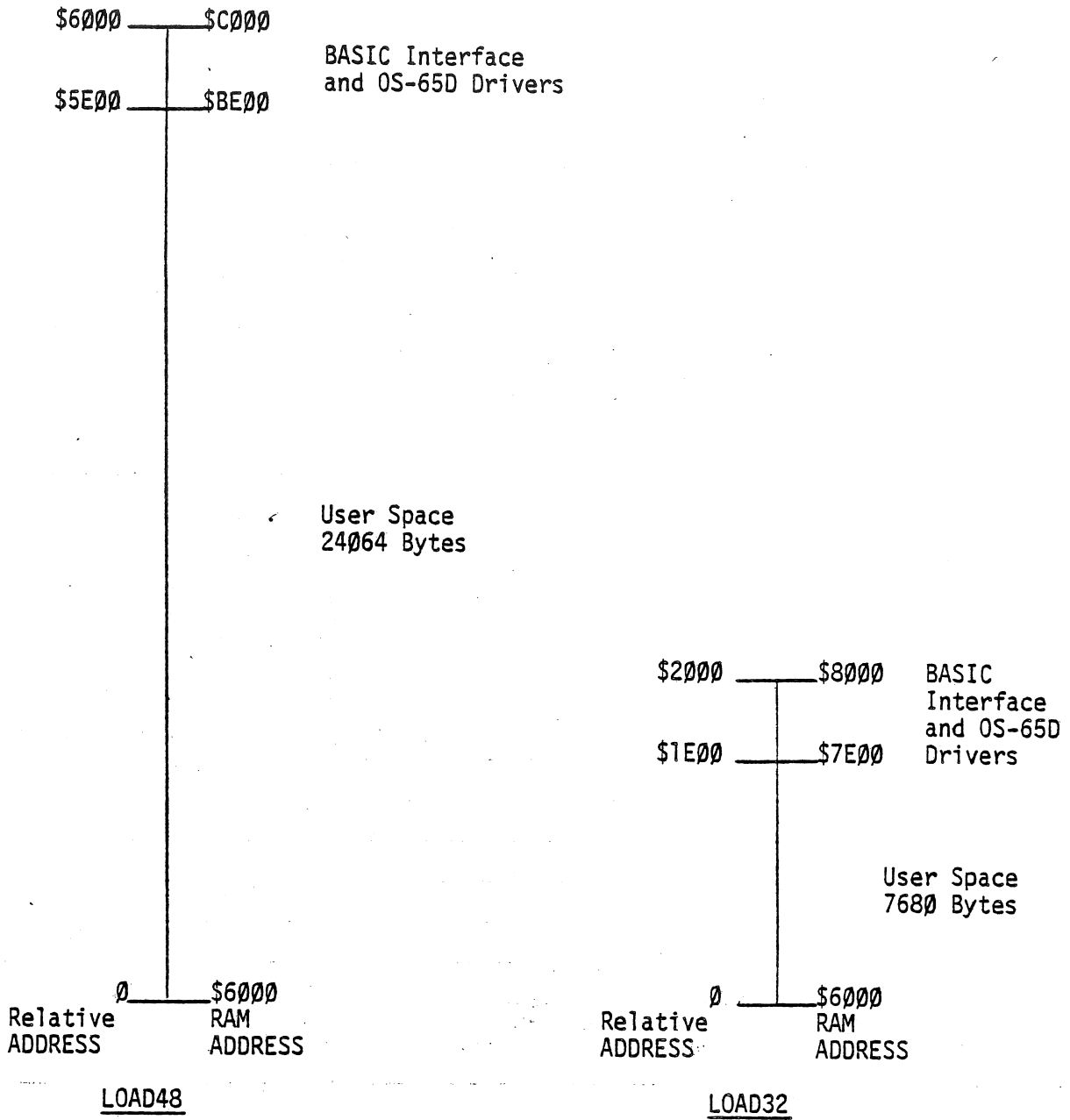


Diagram 1

Diagram 2



Diagrams 3 and 4 show how memory is allocated using "LOAD32" and "LOAD48". For further details refer to page 45 in the OS-65U Level I manual.

FLAG 13 AND FLAG 14 UNDER OS-65U

Flag 13 and 14 are very usefull but are unknown FLAG Statements in OS-65U. These FLAGS permit BASIC programs to be merged as well as provides a method for bringing a BASIC program into the work space without the destructive effect of a "LOAD". To merge two BASIC programs, the following steps must be executed.

Program "TEST2" contains a subroutine between lines 30000 and 31999 that is to be merged into the program called "TEST1".

- 1) Create a scratch data file called, e.g. "Scrap0" with access rights of R/W (Read/Write).
- 2) LOAD "TEST2" i.e. LOAD "TEST2" <CR>
- 3) OPEN "SCRAP0" i.e. OPEN "SCRAP0", 1
- 4) LIST the subroutine into the file i.e. LIST %1, 30000 - 31999 <CR>
- 5) PRINT an "OK" to the file i.e. PRINT %1, "OK" CR
- 6) CLOSE the file i.e. CLOSE <CR>
- 7) LOAD "TEST1" i.e. LOAD "TEST1" <CR>
- 8) Enter: FLAG 13 <CR>
- 9) Enter: OPEN "SCRAP0", 1 <CR>
- 10) In the immediate mode enter:
INPUT %1, <CR>
- 11) A "SN ERROR" will appear on the screen, after this enter:
FLAG 14 : CLOSE <CR>
- 12) Now SAVE "TEST1" to disc. The subroutine has now been merged.

If one desires to place a machine code routine in front of an existing BASIC program.

- 1) LIST the BASIC program to a data file as in steps 1 through 6.
- 2) Using "LOAD32" or "LOAD48" Call the machine code into OS-65U.

- 3) Type a "NEW XXXX" (where XXXX equals the site of the machine code program plus one).
- 4) Proceed with steps 8 through 12.
- 5) The BASIC program will now be placed above the machine code routine.