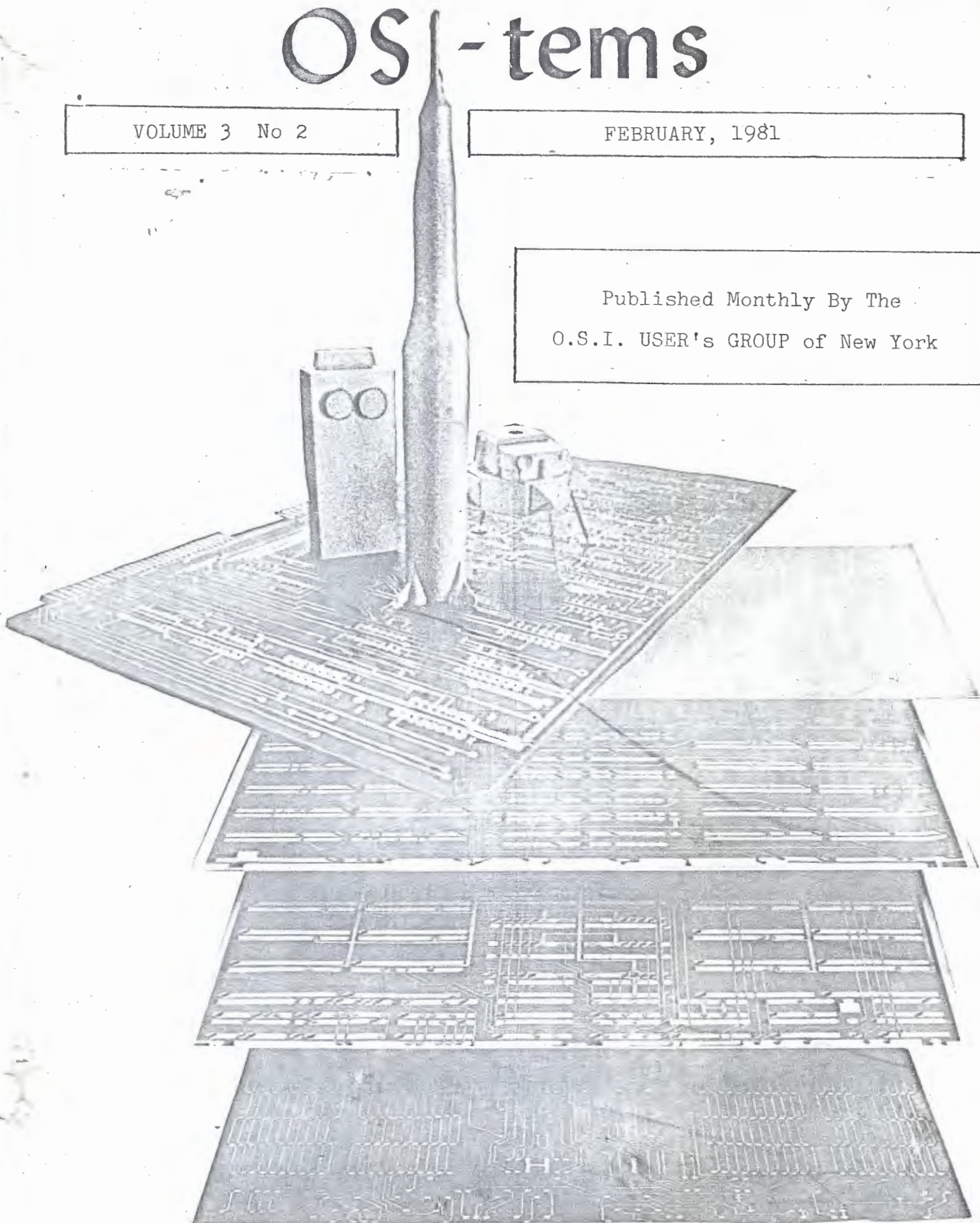


OS - tems

VOLUME 3 No 2

FEBRUARY, 1981

Published Monthly By The
O.S.I. USER's GROUP of New York



dy

OSI-tems is distributed by the Challenger User's Group of New York (CHUGNY).

Club meetings are at 7:00 PM the first thursday of every month at:

Polk's/Aristo
314 5th Avenue
New York, NY 10001

Dues are \$10 yearly, and include 1 year of OSI-tems. Student memberships are \$5.

Table of Contents

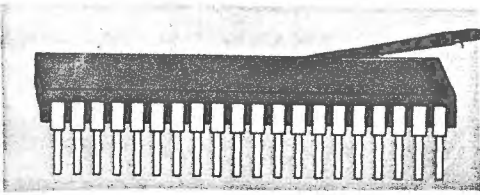
3	Beyond the Keyboard...The Unbreak Mod	Larry Thaler
8	Software Reviews	Mike Bassman
12	Extensions for OS-65D Basic	Danny Schwartz
14	Quick Printer	Klaus Ernst
16	C1/C2/C4/C8 Conversions	Mike Bassman
18	SEB Review	Salomon Lederman
20	Source Update:CEGMON	Thomas Cheng
23	Shape Plotter - Version 2	Mike Cohen
26	Program Reviews	Warren Modell
27	Reagan vs. Re	Warren Modell
28	Advertising and Public Relations	Warren Modell
29	Book Review	Warren Modell
30	Factorials	Thomas Cheng
33	Angry Letter	Terry Terrance

This months issue co-edited by Larry Thaler and Mike Bassman

OSI-tems staff: Salomon Lederman
Thomas Cheng
TerryTerrance
Larry Thaler
Mike Bassman
Danny Schwartz
Mike Cohen
Ugo Re

All material in OSI-tems is the © copyrighted material of the original authors. Reprints forbidden without the express permission of the original authors.

Special thanks this month goes to the St. Louis OSI User's Group, who have given us a set of their newsletters, and also are letting us reprint their material. They have been sent a number of our issues, and will contact the original authors for reprint permission.



BEYOND THE KEYBOARD

by Larry Thaler

Whenever I read O.S.I.tems, I think to myself, "Why don't we have monthly columns like all the other magazines?" Well, they say if you want something done, you should do it yourself, so, here we are; A bimonthly column dedicated solely to hardware and modifications.

Before I go any further, I guess I should list my credentials. I've owned my Challenger 1P for a little over a year now, and have been a member of the user's group ever since that time. Also, I've been editor for two O.S.I.tems, and I've written many programs and articles.

My computer on the other hand is much less qualified. It's front panel has more switches than the cockpit of a seven-forty-seven, and the inside of my C1P has a whole mess of assorted garbage lying across it (it's all neat mind you, there's just a lot of it). But unbelievably so, it still works !!! This alone should tell you how qualified I am.

One of the biggest problems with doing any modification to your computer is that of standardization. Everytime you add another device that someone else doesn't have, your machine becomes more and more different. This isn't so bad as it stands; however, every time this happens, you have to modify your old software, so that it still works with your computer. It is for this reason that I have so many switches on the front of my computer, I believe that every modification I do, has to be able to be undone with the flip of a switch. It does no good to speed up the cassette port to 600 Baud, if you have to throw out your old tapes. Right?

THE UNBREAK MOD

O.S.I. computers are really super, but there's one thing they really goofed on. On all O.S.I. keyboards, the break key is in the wrong place. I don't know about you, but I always manage to hit the breakswitch by accident, usually wiping out any program that I have. This month's mod is just a cure to this problem. Instead of using the break key attached to the keyboard, we'll install a new break switch, one you can't hit by accident.

For this mod, the only parts you need are a good sturdy momentary switch, and some short lengths of wire. Radio shack part number 275-1694 (\$1.99) will do fine, although any spst momentary pushbutton switch will suffice. The wire can be any gauge hook up wire as long as it's stranded core. The tools you'll need are: A drill with 1/4" and 1/8" drill bits, a soldering iron, a wire stripper, and an exacto knife or a razor blade.

1) Open computer case. On a C1P, remove six screws on the bottom, and five screws near the keyboard on top. Be sure computer is unplugged

2) Find the break switch. You will note that two wires or leads go to the switch. One of these goes to ground (the big silver foil at the side of the board). The other one is the one we are interested in.

3) Take the exacto knife, and cut the trace going to the break switch.

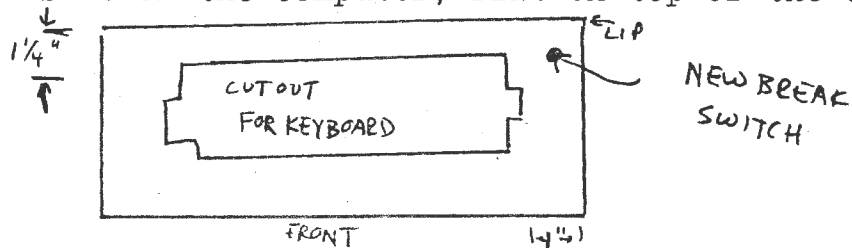
4) Find the nearest hole on the lead that you just cut. Be sure that it is attached to the cut foil. (see diagram)

5) Strip one end of the wire, and attach it to the hole you just found. Solder it on the bottom of the board.

6) Take another piece of wire, and connect it to ground (the other side of the break switch). We can now attach the new switch to these two wires.

7) Before we drill the hole for the new switch, again check to see if the computer is unplugged. If it is, then disconnect the power supply (Red and black cable in rear) and bring the cover of the computer down to the workshop. Be sure it is far away from the computer board, we don't want any filing falling into the board.

8) Find a suitable place to mount your switch. I mounted mine 1" from the side of the computer, flat on top of the case.



9) Mark and drill an $1/8$ " hole into the appropriate spot. Enlarge the hole using a $1/4$ " bit.

10) Scrape off any burrs that may have been on the other side, and shake off all the filings. Bring the case back upstairs.

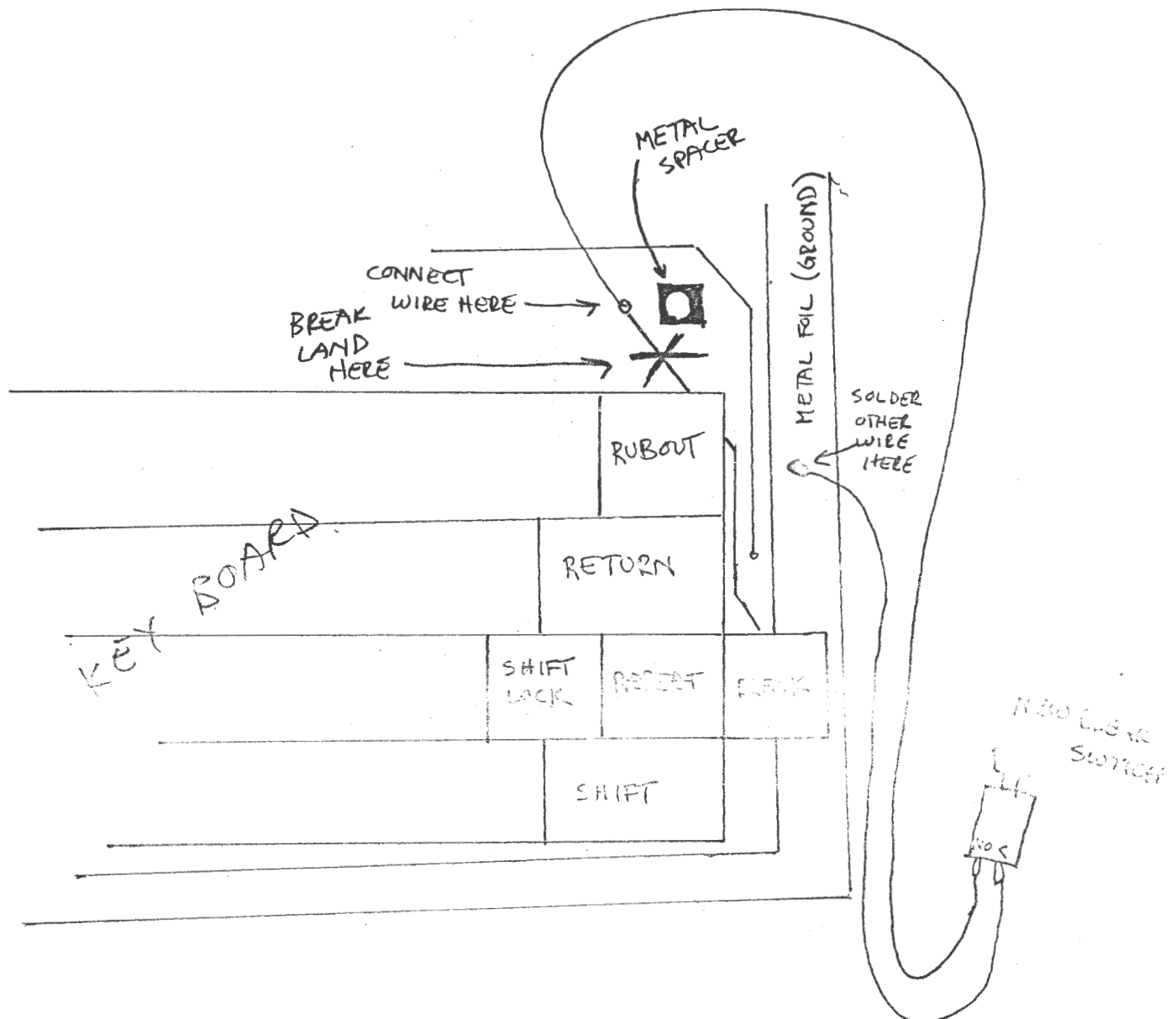
11) Solder the two wires we connected to the Normally open positions on the switch. These are labeled NO and C. Be sure to leave about eight inches of slack so we can take the cover off again without a big hassel.

12) Mount the switch, and chack your work.

13) Reconnect the power supply, screw the board back into place, and put everything back the way you found it.

14) Plug the computer in, and see if the new switch works. The screen should read D/C/W/M? when you push the new button, and nothing should happen when you push the old one.

In the works for next month is a circuit board that will let you switch between ROM chips, so you can use either SYSMON or CEGMON, or whatever you want. Also in future months are cassette motor control circuits, 600 baud cassette ports, and other things.



Well then, the first problem facing us is where to put all these switches. The answer to this is easy. O.S.I. was kind enough to provide a little lip on the front of most of its computers (This applies to ALL series I machines, I have no idea where to put switches on series II challengars). If we use this lip to mount all of our switches, we see that not only does it provide easy access to the switches, but it puts them near to critical areas on the computer board, and doesn't interfere with an expansion board (if you choose to buy one). If we drill $\frac{1}{4}$ " holes on the front panel, seperated by 1" spaces, we have room for about twelve micro-switches. On mine, I had to move the "O.S.I. Challangar" nameplate to the top, to make room for more switches. Whenever we do a mod, we'll mount the switches here (with the exception of the new break switch mod - that comes later). See diagram for a better explanation.

Bits....contributed by Salomon Lederman

An easier way to print out a graphics character is to CHIT 513 with the value.

- B385 is an entry point of the coldstart routine which performs the memory initialization.

08....

Software Reviews

Mike Bassman

39-65 52 st.

Woodside, NY 11377

This month has seen the arrival of some outstanding new software. I hope the trend continues.

Galaxia - Hardvark Technical Services

This is an all machine language copy of the arcade game 'Galaxian'. For those of you who know and love the game, this is very good news! The fighter squadrons are fast, bloodthirsty, and capable of performing the same acrobatics as you are used to. The graphics are good, and the play is improved as Galaxia makes use of the OSI half characters, which effectively double the resolution of the playing field. For those of you who have not seen the game before, it is something like this. In the beginning, it looks somewhat like Space Invaders. However, instead of having the aliens move down, a number of them (one, two, or three) swoop down at you (in formation) to attack. They come at you in long winding curves, making them difficult to hit. Sometimes, they will even turn around and go back up for another shot at you. During all this action, they are continuously letting loose a barrage of missiles. This one will keep you and your

09....

family busy for a while. In conjunction with this, I propose that someone design a modification to outfit your Challenger with a coin slot.

Minas - Aardvark Technical Services

Minos is a neat new three dimensional maze game. Basically, it makes a two dimensional maze, which you must solve. However, it puts you right down into the maze, as opposed to the know-all view from above. What this means is that you can see a corridor stretching down a way with a left turn halfway down and ending in a right turn. You 'walk' through the maze by hitting a key, which depicts the walls moving by and the turnoffs and such getting larger until you pass them. You can turn left, right, aboutface and move, as well as drawing a map that shows you where you have been. There are many dead ends and only one possible solution so the game is very tricky. The graphics on this one are, without the best that can be done on an OSI. Walls and perspective are drawn out in simulated high resolution. Once you get the hang of this one, it gets fairly easy, but the incredible graphics make it worthwhile alone.

Video Games 3 - Orion Software Associates

Here is a series of space games that will really knock your socks off. This is a set of three games, so let me detail the features they have in common. First of all, you control the acceleration of your ship, not your velocity.

10....

You move like a real spaceship, not a tank. In other words, to slow your ship down, you would accelerate in the opposite direction that you are moving. Second, each ship possesses a rack (four) of real-time machine gun missiles. Third is machine language. All three are written in it for very fast execution speed. If anything, these are too fast rather than too slow (a refreshing change of pace!). Now let me describe the games.

1 - Meteor Wars

This is a two player game where the goal is to destroy the meteors that are floating around the screen, and try to kill your friend at the same time.

2 - Space Wars

Just like the above, but without meteors, and the added danger of crashing into the sun.

3 - Meteor Wars II

This is a two player game where the goal is to destroy as many meteors as you can without hitting the sun. If you destroy all the meteors, a new set with more appears.

Extended Basic - Software Alchemists

Here is a small machine language program for any cassette based C1 that adds a number of amenities to Basic. First of all, Shift-O now gives a real backspace. RUBOUT provides a machine language language screen clear. Control-S allows you to single step your way through a Basic program.

11....

Control-T will put you in the trace mode, where the line number currently being executed will be printed. The left shift will slow down the printing speed until you restore it with a right shift. A shorthand is built in, where hitting two keys will print an entire word like THEN, NEXT and things like that. In addition, there are some actual additions to Basic. There is a command which will goto a specific line number if the esc key is pressed. Another command will constantly monitor the value of a variable. When the variable equals a specified value, it goes to a defined line number. Also, there is an addition to allow the input of strings with commas. This entire package requires only 1/4 of a K!.

EXTENSIONS FOR OS-65D BASIC

by Dan Schwartz

This program provides four extensions to the standard version of OS-65D U3.X Basic.

1) The command NEW (number of bytes). This command eliminates the need for the CHANGE program by allowing you to change the lower limit of the workspace with a single command. For example, NEW 2048 for a 5-inch disk system or NEW 3072 for 8-inch will reserve room for one disk buffer. NEW 0 will return to a normal workspace. One difference between my implementation and that found in OS-65U is that in this version, the command NEW by itself does not change the workspace limits, but just clears the current workspace as usual. Another is that general expressions are allowed after the word NEW, for example NEW 12*256.

2) RESTORE (line number). Restores the data pointer to the beginning of the specified line. The line number can be any expression, but must evaluate to a line actually present in the program to avoid an error message. The line need not actually be a DATA statement. If it isn't, the next data item will be read from the first DATA line after the specified line.

3) CALL (subroutine address). This one has been done before (see "Machine Language Call" by Mike Cohen in OSI-tams Vol. II, No. 8 (Aug. 1980)). The only difference between my implementation and Mike's is that I combined the two successive subroutine calls needed to evaluate the address into a single subroutine, so that it can be called from other places.

4) GOTO or GOSUB (expression). Once that subroutine is in place, after routines can call it. To replace a call to GOTO with a call to GOSUB, simply change the GOTO to GOSUB and the line number to a line number. For example, GOTO 100 becomes GOSUB 100. The code for GOTO and GOSUB is the same, so you can use GOTO or GOSUB interchangeably. For example, GOTO 100 becomes GOSUB 100 and GOSUB 100 becomes GOTO 100.

Note: the code for NEW XXX as given here also uses this evaluator, so it cannot be used unless CALL is enabled. The code for RESTORE XXX uses it only by way of the GOTO code, so it will work with or without the evaluator in place, but will only be able to use explicit line numbers if it is not.

```

10 REM  EXTENSION PROGRAM FOR OS-650 BASIC
20 REM  DANIEL SCHWARTZ  75 E 190 ST  BRONX NY 10468
30 PRINT
40 PRINT"ENABLING 'NEW XXX', 'RESTORE XXX', CALL AND COMPUTED GOTO/GOSUB"
50 PRINT
60 CC=PEEK(2073):POKE 2073,96:REM  LOCK SYSTEM WHILE CHANGING
70 REM
80 REM  POKE IN CODE FOR 'CALL', WITH EVALUATOR AS SUBROUTINE
90 FOR X=0 TO 11:READ C:POKE 2157+X,C:NEXT
100 REM  POKE IN KEYWORD LETTERS 'C' AND 'A' FOR 'CALL'
110 POKE 709,67:POKE 710,65
120 REM
130 REM  POINT GOTO CODE TO USE NEW EVALUATOR
140 POKE 2215,115:POKE 2216,8
150 REM
160 REM  RESERVE 104 BYTES IN HIGH MEMORY
170 M=PEEK(132)+256*PEEK(133)
180 M=M-104:MH=INT(M/256):ML=M-256*MH
190 POKE 132,ML:POKE 133,MH:POKE 128,ML:POKE 129,MH
200 REM
210 REM  POKE IN CODE FOR 'NEW XXX' AND 'RESTORE XXX'
220 FOR X=0 TO 103:READ C:POKE M+X,C:NEXT
230 REM
240 REM  SET UP NEW DISPATCH ADDRESS FOR 'NEW'
250 ND=M-1:NH=INT(ND/256):NL=ND-256*NH
260 POKE 566,NL:POKE 567,NH
270 REM
280 REM  SET UP NEW DISPATCH ADDRESS FOR 'RESTORE'
290 RD=M+55:RH=INT(RD/256):RL=RD-256*RH
300 POKE 534,RL:POKE 535,RH
310 REM
320 POKE 2073,CC:REM  RESTORE PREVIOUS CONTROL-C STATUS
330 REM
340 PRINT M-12670"BYTES FREE"
345 REM  FOR MINI-FLOPPY USE M-12926 FOR BYTES FREE
350 REM
360 REM  OBJECT CODE FOR 'CALL', WITH EVALUATOR SUBROUTINE
370 DATA 32,115,8,116,33,5,76,105,10,76,114,25
380 REM
390 REM  OBJECT CODE FOR 'NEW XXX'
400 REM  FOR MINI-FLOPPY CHANGE FIRST FORTH DATA ITEM IN LINE 390
410 DATA 76,105,10,76,114,25
420 REM  245,114,25,1,13,10,20,21,105,103,173,150,20,130,19,160,130
430 DATA 106,133,106,2,137,152,141,3,76,76,4,233,3,176,2,176,20,133
440 DATA 120,132,121,233,1,176,1,136,133,25,152,26,160,0,152,145,25,24
450 DATA 76,20,6
460 REM
470 REM  OBJECT CODE FOR 'RESTORE XXX'
480 DATA 208,3,76,10,8,170,165,199,72,165,200,72,165,134,72,165,135,72
490 DATA 138,32,166,8,165,191,133,142,165,200,133,143,104,133,135,104
500 DATA 133,134,124,133,200,134,133,199,32,191,0,208,251,96

```

" Handshake (CTS) of Quick Printer II DOES work with C1P by Klaus Ernst

Attracted by the low price and encouraged by a favorable review in AARDVARK JOURNAL vol.1, no.1 p.7 I bought a Radio Shack QUICK PRINTER II.

I had already populated the RS-232 OUTPUT on my 600 board, so all I had to do was, to double the BAUD rate. I studied Aardvark's "600 BAUD CASSETTE *PRINTER CONVERSION" instructions and decided all this messing with capacitors etc. was not for me. So I did the second version on the update sheet instead. Next I poked the values listed on page 8 of AARDVARK JOURNAL vol.1 no.1. The printer worked fine within the limitations described in the Aardvark article. Instead of a handshake (CTS=clear to send) the printer is slowed down by software to avoid the loss of characters.

Then I came across a Quick Printer II Fix by our own Danny Schwartz (yeah Danny!) in AARDVARK JOURNAL vol.1 no.3 p.5. With this loaded up the handshake should work. First I had to hook up the CTS line. The schematic of the printer shows that CTS swings between +5 VDC and -6.2VDC. I did not want to hook up a negative voltage directly to a TTL chip, instead I populated the RS232 INPUT (Q2 et al.), fed CTS thru there (where it also gets inverted) and connected it to CTS (pin 24) of the ACIA(6850). Pin 24 had to be disconnected from GND first (cut trace at W5). I installed a switch to be able to *CTS either to GND or the CTS line. I loaded the fix, typed SAVE and a PRINT command, hit RETURN and - the system locked up!

I checked with a couple of knowledgeable people, who said: "Forget it, the handshake won't work!"

Then at the last OSI meeting I checked with Danny, who told me to type in Aardvark's list of BASIC commands. It should read: GDB: 6D102 (not 80102)!

Well, I could not wait to try it out. I loaded the corrected fix - and it worked!

I wrote a BASIC loader for it, which will selfdestruct after you run it(220 NEW). The fix suppresses NULLs that confuse the printer. Never use CHR\$(0)(NUL) more than once (in a row) or your machine will lock up, because the CTS line will stay low.

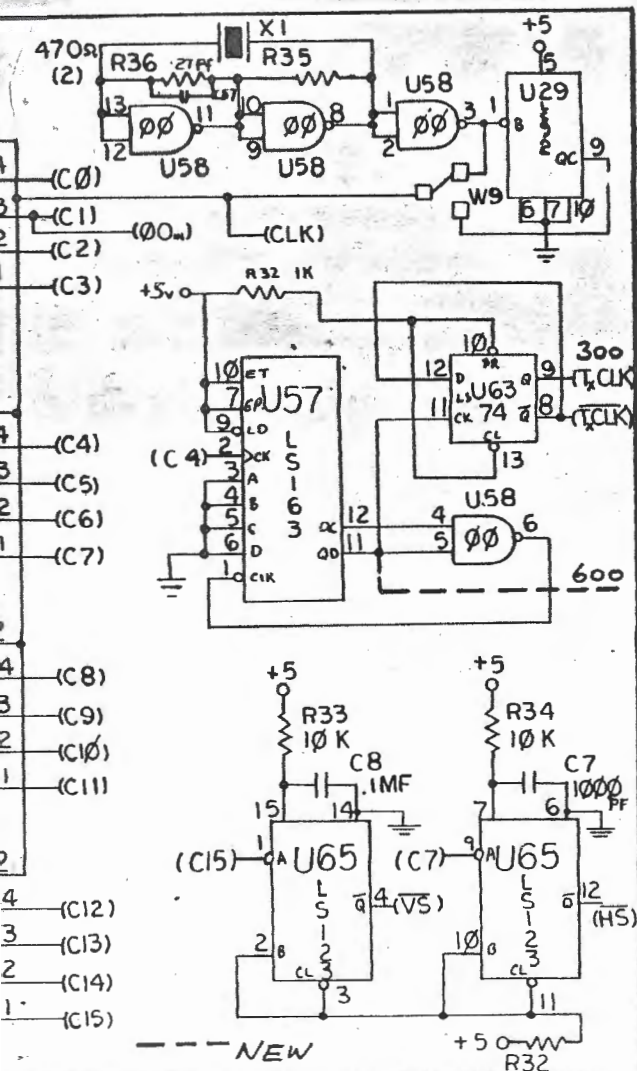
* CONNECT

LIST

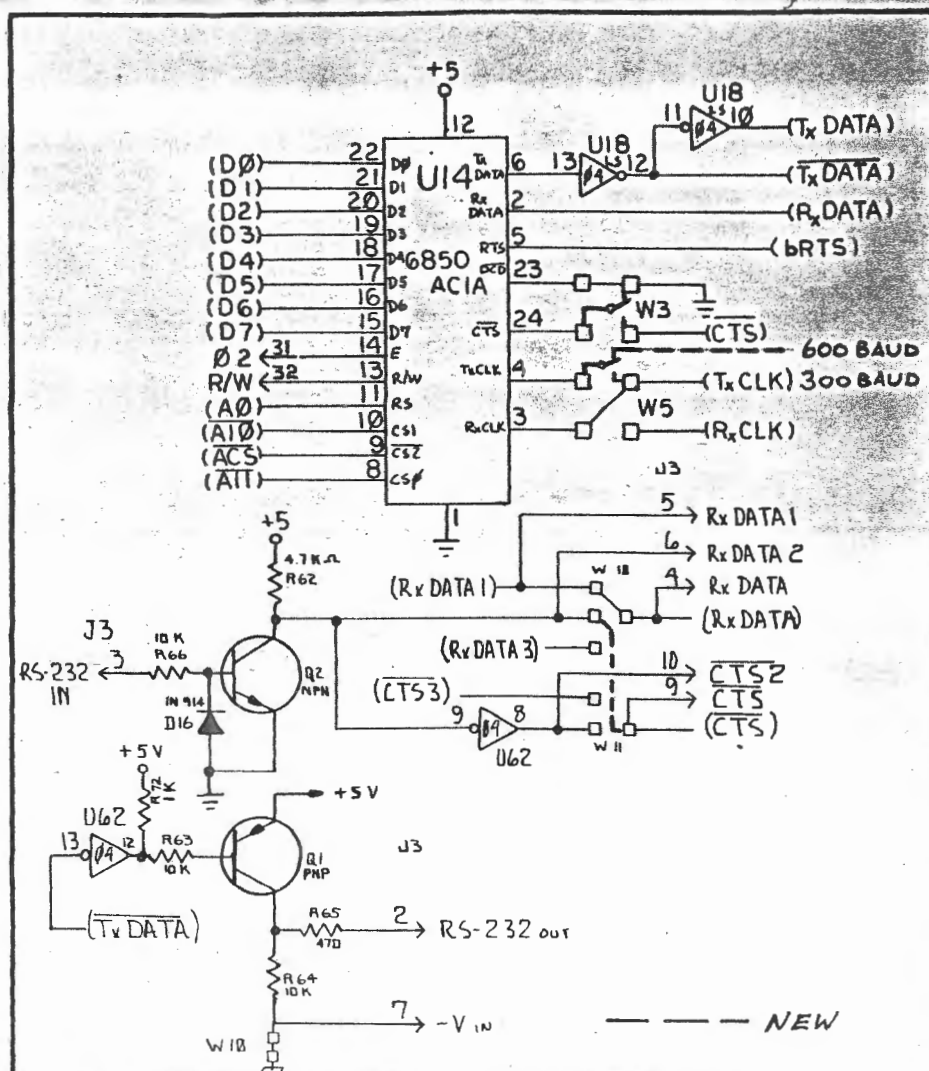
```

REM QUICK PRINTER II FIX
* CTS
* NEW BASIC LOADER BY KLAUS ERNST
GDB: 6D102
GDB: 6D102
200 DATA 32,45,191,72,173,5,2,24
0,4,104,76,177,252,104,96
210 DATA 169,34,141,26,2,169,2,1
41,27,2,76,116,162
220 NEW
OK

```

NOTIFIC	product name/number	
page	status	sheet 9 of 13



OHIO SCIENTIFIC	product name/number	
date 5 SEP 1978	revision B	page
status	sheet 6 of 13	

HARDWARE MODS FOR 600 BOARD FOR USE OF R/S QUICK PRINTER II

REMOVE: CUT JUMPER BETWEEN GND PAD AND CTS PAD (PIN 24) AT W3

INSTALL: 1) SPDT SWITCH AT W3 (NOTE 1)

2) POPULATE RS-232 IN

3) ADD JUMPER FROM Rx DATA 2 PAD OF W10 TO CTS PAD OF W11

4) CONNECT CTS LINE OF PRINTER TO RS-232 IN

BAUD RATE SWITCH:

REMOVE: CUT JUMPER BETWEEN Tx CLK PAD AND PAD CONNECTED TO PIN 4 (Tx CLK)

INSTALL: 1) SPDT SWITCH AT W5 (NOTE 1)

2) RUN NEW WIRE FROM ONE SIDE OF SWITCH TO PIN 11 OF U63. (600 BAUD LINE)

NOTE 1: THIS IS WHERE THE WIRES GO, PUT SWITCH IN CONVENIENT LOCATION.

C1/C2/C4/C8 Conversions

Mike Bassman
39-65 52 st.
Woodside, NY 11377

Have you written a neat program that you think you could sell? If you have ever tried to sell your program, the firm you are selling to will probably ask you to convert your program for use by other OSI computers. Is this difficult? Not really, it's just a matter of knowing the right tricks. First of all, OSI computers are divided into to sections. These are C1P, and then there is everything else. This is because the C2P, the C4P, and the C8P work similarly. The differences between the C2/4/8 series and the C1P series are the screen size, and the keyboard. The C1P has a 24x24 video display, and the C2/4/8 has a 64x32 display.

The first step in doing program conversions is to have your program detect which machine it is on. It can be done by PEEKing location 57088 (the keyboard). If the contents are greater then 127 then it is a C1. If they are smaller than 127, its a C2/4/8. So, the first line of your program could be:

```
10 IF PEEK(57088) >127 THEN 50:REM 50 SETS C1 VALUES
```

The next, and probably most important part is converting the screen size and addressing. As an example lets say a C1P program is being converted to work on a C4P. To make life a lot easier, we will bring the C4P screen down to 32x32 by saying "POKE 56832,0." Now both screens are more or less the same size, but the addressing of it is still different. The C1P has a line length of 32, while the others have a line

17....

length of 64. You should set a variable representing the upper left hand corner of the screen. On a C1, this would be 53380, on the others, a 53442. Now everything can be referenced indirectly. Instead of POKEing some number with a value, POKE the upper left hand corner + the number of characters across + the number of lines down * line length will give a value that will work on both machines, provided the corner and line length variables have been set correctly.

The last thing necessary to make conversions work are the differences in the polled keyboard. The C1P values for both the keyboard row select and key detect are 127, 191, 223, 239, 247, 251, and 253, in that order. On the other machines, the corresponding values are 128, 64, 32, 16, 8, 4, and 2.

Happy converting!

This article is intended as an objective hardware review of SEB, the Super Expansion Board for your C1P. SEB is manufactured by Grafix, 911 Columbia Av. N. Bergen, N.J.

SEB is a graphics board that turns your C1P display into a hi-resolution pixel driven video display. Effective resolution is 256x192 individually programmable dots. This is nearly as good as ATARI(320x192), or APPLE(280x192). SEB uses a Motorola MC6847 video display generator chip (the same one as the APF Imagination Machine). Some of the hardware features are as follows:

- 1) True hi-res in 12 modes
- 2) Directly supports up to 16K of user RAM
- 3) 8 colors (less in the higher res modes)
- 4) All of video RAM needed for hi-res (SEB pro (64K 5K))
- 5) On board RF modulator

As for the installation, SEB is connected to a not a hardware person. I had a friend put it in for me. There were no problems to cut. However, the polarization is reversed on the power supply connector, no big deal to fix but annoying until you notice it. SEB fits nicely into your C1 case. My SEB runs on one power supply, you may need another one if you don't have low power memory chips.

19....

I've had SEB for nearly a month now. I've been working on software for SEB graphics for a while now. Objectively I must say that it is quite difficult to do hi-res software. The resolution is very nice but the effort is quite great as compared to normal CI graphics. If you are impressed by graphics on the ATARI, APPLE, SORCERER, etc... then you will like SEB.

SEB has its own character generator. However characters cannot be mixed with graphics. In the character mode you get 32 characters by 16 lines. In the hi-res mode you can theoretically create your own characters but this must be done by defining the bits for each character. The character mode is good for text. A friend and I have written a fairly simple machine language routine that will drive BASIC display on SEB. Thus I can run hi-res graphics programs and when a system error message comes up the display will switch to character mode.

SEB is nice price for hardware. However, I must say not that good at programming then you will have some trouble working with the document A100, which is hardware oriented. You should be familiar with hardware if you want to greatly benefit from the hi-res graphics which is highly bit oriented. All in all SEB provides a great challenge to he who wants to design the ultimate graphics programs. Good luck!

SOURCE UPDATE: CEGMON

CEGMON is the name of a new monitor chip put out by a British group. Cegmon extends and improves upon the old SYMMON chip, which is supplied with your original computer. Among its features is the elimination of OSI's cursed cursor backspace with a true backspace/delete. This alone will probably prompt some of you to plunk down the \$60 for the chip; but wait: there's much, much more!

CEGMON IN BASIC:

The keyboard under CEGMON is now a "real" typewriter. This means that the only adventure you can play on same is one purchased from a software firm, (try Marooned in Space or Ghost Town by Orion. Trust me.) not the one supplied by OSI. Rather than guessing which shift key does what, when and where, the keyboard is now "normal". Shift P is not a line delete anymore, but shift zero is. (This is a mistake. I always hit that thing when I'm trying for a right parentheses;) Shift-O with the shift lock down is still a backspace. The rubout key is now decoded as a backspace/delete; the one the CEGMON writers advise that you use, because the shift-O only works with the shift lock down. With shift lock up, you will get a capital-O when you hit shift-O.

The so called "screen handler" uses part of page two to define a software "screen" on the true, physical screen. With their new screen handler, you can define windows, bottom of the screen, etc. In reference to the screen handler, several new "commands" are supported. Some of these are:

```
PRINT CHR$(10).....moves cursor down one line (cursor is now free).
PRINT CHR$(11).....moves cursor right one space.
PRINT CHR$(12).....homes cursor to top left of screen.
PRINT CHR$(13).....CR-moves cursor to beginning of same line.
PRINT CHR$(26).....clears entire screen.
PRINT CHR$(0).....clears the display.
```

(Note: The software supplied does not have a screen memory window. If(26) will clear the entire screen from 000 to 0400. (For CII.)

CEGMON documentation states that you can access these commands by holding down a control-something. Don't believe it. Those same control keys are masked out by BASIC. They do, however, work perfectly fine in the machine language monitor.

Best of all is a screen editor. With it, you can edit in mid-line. While not like the PET or ATAKI screen editors, this system is workable and convenient, and has peculiar strengths of its' own. Pressing control-E will turn on a block at your current cursor position. Using other control keys, you can maneuver this block about the screen, to the point at which you want to copy from the screen into memory. When you press escape, the letter beneath the block is copied into the BASIC line buffer, and the current cursor location; this echo effect keeps you updated. From there, it is treated

as if you had entered it from the keyboard. There you can leave it, or erase it with a backspace. You can also keep moving the special block/edit cursor about, to copy different section of a program or line. With this, you can combine several lines into one by copying into one BASIC line. As an added bonus, OSI's assembler/editor will also possess real backspace and line editing, since it also uses the monitor ROM. (The extended monitor also, but I haven't had a chance to try that out yet.)

CEGMON IN MACHINE:

After working with OSI's six digit display in the machine language monitor, CEGMON is a true pleasure to use.

CEGMON retains the OSI ML Monitor command format, but little else is the same. Entry into data mode is still with a slash, exit with a period, and incrementing the memory location in data mode with a return. In addition, you will have:

- 1) Scrolling
- 2) Memory Move
- 3) Monitor format save and autostart
- 4) Load (with space bar used to exit the mode!)
- 5) Tabular display (hoorah!)
- 6) Breakpoints and break table, restart.
- 7) unlisted others.

These are the major commands. The general ease of use is enough to set you drooling with pleasure. If you are a machine language programmer, get this at once. It is more valuable in machine language than in BASIC.

Memory move is a simple move. It does not recompute JMP's and JSR's for you. Tabular display is an eight byte per line hex display of specified memory locations. The cursor and scroll will continue to act in CEGMON's fashion, that is, the cursor is free over the screen, not fixed at the bottom line. It is a matter of setting the option of no/scroll in monitor at some time. This is obtained to obtain speed in loading in from tape. (The screen doesn't have to scroll) A line-feed in data mode will scroll down, opening a new memory location, and keeping the old one and it's contents still on the screen.

RANDOM THOUGHTS AND WARNINGS:

CEGMON is great for me, but perhaps not for you. I have an 8K C1P, and already have some problems with software compatibility. CEGMON does use various bits of page zero and page two (the use of page two is only to keep tabs on the new defined screen. POKE 538;149 will bring back the -gack- SYNMON monitor, along with attending fake backspace and fixed cursor. But it will free page two from \$0222 to \$02FA.) Any heavily based ROM program will probably have a bit of difficulty in operating, neither will a program which uses location 538 or various bits of page

zero. For example: WP6502 will not load in (tape version); Aardvark's Maze program will half-work. Earthship's Lunar Lander will not work at all. (It think it uses location 538 for something.) HEXDOS for the C1 with disk will not work. However, some compatability has been maintained, but by no means as high as you would wish.

AHEM. **ORION software runs.** Aardvark's cursed cursor control burns. \$FEED still returns a character at location 531. The Cegmon manual tells you how to construct a non-halting GET statement, how to construct a BASIC trace routine, how to simulate PRINT AT.

Documentation is professional.

I've only run into one minor bug in CEGMON, in the Machine language monitor (BASIC PROGRAMMERS heave a sigh of relief); When you are typing in data mode-ASCII mode, the backspace will not work correctly. To see the bug, enter this mode, then type directly to screen memory. You will see the ASCII characters being directly entered into memory, but if you try to backspace, you will see that it will be entered in the next free or consecutive memory location as the old BASIC underscore symbol, an ASCII 96. (96? I'm working on this at 12:30 AM, the night after a double test, so don't expect me to be perfect. If I'm right, so much the better.)

Hopefully, I have gien you an overview of what CEGMON is and what it does. I feel that it is well worth the \$60 price tag, even though it does have a few drawbacks. (Someone is trying to construct a CEGMON/SYMON switch, so this too, may come to pass.)

Installation for me was fairly painless; a cut trace, and one soldered jumper (DO NOT LET MICHAEL BASS CUT OR SOLDER FOR YOU.) as I understand it, installation of the device is not complicated - some things to do in a laboratory.

THOMAS CHENG
26 MADISON STREET
APARTMENT 4J
NEW YORK, NEW YORK 10036

SHAPE PLOTTER - Version 2

by Mike Cohen

Here is an update on Sol's shape plotter from November 1980 OSItems. For this one, I have added multiple shape tables and adapted it to a disk system.

Many disk users wonder where to put machine language routines, since page 2 isn't free. Often, the solution is to put it in high memory, but how many times have you forgotten to set top of memory and had your machine code wiped out by strings???

Here's the solution - put the machine code under the basic workspace. Not only doesn't the memory size have to be set, but the machine code rides in and out with the basic program so there's no DATA statements and POKes to set it up.

To set up a program this way, it must be assembled to reside at \$327E, the start of BASIC's workspace. The program must end with 3 bytes of zero and the locations 3279 and 327B must both point to the address of the second zero byte (low byte first as usual). In \$327D you should put the number of tracks the machine language section would take up (remember, on a 5" disk each track is 8 pages).

After the program is properly set up, it should be saved on disk using a PUT command. Then, load up BASIC and call in the machine language program that was saved and type NEW.

The address that you put in \$3279 and \$327B become BASIC's new start of workspace and the zero byte at that location allows basic to use it. If that first byte isn't

zero, BASIC will give an error message. The second zero byte is also important. If it isn't zero, BASIC will give an error message.

That was for the table by the extended memory.

The BASIC program is similar to the BrainCrash program which appeared in C.O.C.E. II several months ago. The machine language routine will work on all systems, but the BASIC demo is for Commodore 64. In the routine, under the address of \$11877, put the address of the first shape table. Then, call it with X=USR(X), as it is done in the subroutines at 1000 and 2000, which select alternate shapes.

5 TIME=150

7 FORM=1T030:PRINT:NEXT:POKE54117,32

10 POKE8955,126:POKE8956,50

15 FORM=53954T053980:POKEM,128:POKEM+32,209:NEXT

17 FORM=53922T053948STEP3:POKEM,INT(RND(3)*3+13):NEXTM

20 DATA 2,2,167,1,157,1,161,3,167,24

30 DATA 165,1,161,1,161,1,161,1,161,1,155,1,176,1,161,1,161,24

40 DATA 166,1,161,1,161,1,161,1,161,1,128,1,161,1,161,1,161,24

50 DATA 176,1,224,1,225,1,226,3,226,2,226,0

60 DATA 165,3,161,1,156,1,165,2,2,25

70 DATA 161,1,161,1,178,1,155,1,161,1,161,1,161,1,161,1,167,24

80 DATA 161,1,161,1,161,1,128,1,161,1,161,1,161,1,161,1,168,24

90 DATA 226,2,226,3,226,1,224,1,225,1,178,0

100 AD=11897:PT=11899

110 P=11901:P1=P

120 DEFFNH(X)=INT(X/256):DEFFNL(X)=X-FNH(X)*256

130 READ Q:POKE P,Q:P=P+1:IF Q>0 THEN 130

140 P2=P

150 READ Q:POKE P,Q:P=P+1:IF Q>0 THEN 150

160 S1=53910:S2=53891

170 GOSUB1000:GOSUB2000

175 FORT=1T0TIME:NEXTT

180 GOSUB1000:GOSUB2000

190 S1=S1-1:S2=S2+1:IFS1-S2>9THEN170

195 GOSUB1000:GOSUB2000

200 FORM=1T06:POKES1+M,42:POKES2+M,42:POKES1-M,42:POKES2-M,42

210 POKES1+32*M,42:POKES2+32*M,42:POKES1-32*M,42:POKES2-32*M,42:NEXTM

220 FORM=1T06:POKES1+M,32:POKES2+M,32:POKES1-M,32:POKES2-M,32

240 POKES1+32*M,32:POKES2+32*M,32:POKES1-32*M,32:POKES2-32*M,32:NEXTM

300 RUN

END

1000 POKEAD,FNL(S1):POKEAD+1,FNH(S1):POKEPT,FNL(P1):POKEPT+1,FNH(P1)

1010 X=USR(X):RETURN

2000 POKEAD,FNL(S2):POKEAD+1,FNH(S2):POKEPT,FNL(P2):POKEPT+1,FNH(P2)

2010 X=USR(X):RETURN

```

3293 00702E LDA #F7C
3294 00FD ST, #0
3295 AD792E LDA $2E79
3296 00FE STA #FE
3297 AD7A2E LDA $2E7A
3298 00FF STA #FF
3299 0000 LDA #0
3294 B1FE LDA ($FE),Y
3296 AA TAX
3297 B1FC LDA ($FC),Y
3299 91FE STA ($FE),Y
329B 8A TXA
329C 91FC STA ($FC),Y
329E E6FC INC #FC
32A0 D002 BNE $32A4
32A2 E6FD INC #FD
32A4 B1FC LDA ($FC),Y
32A6 D001 BNE $32A9

```

32A8	60	RTS
32A9	18	CLC
32AA	65FE	ADC \$FE
32AC	85FE	STA \$FE
32AE	A5FF	LDA \$FF
32B0	6900	ADC #\$00
32B2	85FF	STA \$FF
32B4	E6FC	INC \$FC
32B6	D0DC	BNE \$3294
32B8	E6FD	INC \$FD
32BA	4C9432	JMP \$3294
32BD	00	BRK
32BE	CB	???
32BF	32	???
32C0	0500	DRA \$00
32C2	54	???
32C3	494D	EDR ##4D
32C5	45AB	EOR \$AB
32C7	3135	AND (\$35),Y
32C9	3000	BMI \$32CB
32CB	E532	SBC \$32
32CD	07	???
32CE	00	BRK
32CF	814D	STA (\$4D,X)
32D1	AB	???
32D1	319D	AND (\$9D),Y

PROGRAM REVIEWSScale 1 to 10

Author: Robert J. Retelle
2005 Whittaker Rd.
Ypsilanti, Michigan, 48197

- 1) "Rebel Gunner" Rev. 12-A \$9.95
Rating: 8.5 Three Levels
You're Flying An Xwing Fighter.
You have to destroy TIE Fighters which maneuver very quickly.
Once you lock on to them, in the Crosshairs, your ship automatically fires four phasers, blowing up the enemy fighter. It's not easy!!
- 2) "AAARRRGGG!!!" with Sound \$7.95
Rating: 8.0 Five Levels
Fast Paced. You chase around the screen trying to hit many elusive targets. It's a race against time and you can get a bonus for high score.
- 3) Mike Base Rev. 3 with sound
Rating 6 \$4.50
Good Game. Displays Good Graphics.
- 4) "AAARRRGGG!!!" Rev. 12-A \$9.95
Rating: 7.5 Two Levels

The enemy tank places barriers in various parts of the screen. You have to catch the enemy tank and destroy it in order to keep the area clear. This is difficult as the spaces keep filling up.

Warren Modell, V. P.

I sent for five programs:

#1001 Buller \$5.00

#1004 List \$7.00

#1006 McClintock \$5.00

#10015 Checking & Disbursements \$7.25

Not one of them worked. I sent them all back requesting a replacement. It's just about one week now. Hopefully I'll have it by the next meeting in February and I'll let you know if it worked out. Bills Mico service sends a contract stating you will not duplicate except for a backup copy.

Warren Modell, V. P. 3133 Rochambeau Avenue, Bx. N.Y.N.Y.10467
OSUNY

Reagan Vs. Re

UGO Re our new President promised to do more for us the first Thursday evening of each month, than President Reagan. On that promise the Ohio Scientific Users Group of New York overwhelmingly voted UGO in as Prez.

Of course, whenever a politician makes promises you tend to take it with a big grain of salt. The salt has been dissolved. This politician is actually keeping his word!! It's too bad we can't make him President of OSI!!

Our founding president, David Gillette, certainly deserves a big round of applause! Starting from scratch, and he did have to scratch for our benefit, was no easy job. His hard work will go down in the history of Anglo-American Diplomacy. Dave - we all thank you very, very much.

Some Some of our new activities are really packing USERS into a standing room crowd!!

- 1) Assembly Language Class
- 2) Speakers, Demonstrations and Films (Hardware and software).
- 3) Nifty Tricks (not from 42nd Street) "How to flash with your challenger."
- 4) Basic Electronics Class - "How to add to or Repair your challenger without the use of a hammer"

re first... to the... we've been able to... more things we can do. Let our... to... Remember, we're a... group.

Information is continually... OSI Corporation... will all...

their... articles... with some articles.

one of our members... writing...

Warren Modell, V.P. OSUNY
3133 Rochambeau Avenue, Ex. 10467

ADVERTISING AND PUBLIC RELATIONS

Every once in a while OSI gets some coverage. Have you heard!

China may soon announce a production agreement for the "THREE PROCESSOR" C3 System. Negotiations are now going on between OSI and the Yangtze River Industries who also manufactures computers. The Chinese are interested in Banking, industrial and process control applications.

OSI has always lacked good advertising and Public Relations. We can learn from their poor record. They advertise in the wrong place with the wrong message and don't respond to their customers or potential customers with either warmth or affection. -

So let's:

1) Send letters to every computer magazine and newspaper at least once every three months making them aware of OSUNV. Tell them where and when we meet. Include a short synopsis of our interests.

2) Authors - mention you're a reviewer

3) Professors - state you're

4) Students - mention you're

Meeting

5) School Students

BOOK REVIEWMicrocomputers - A Technology Forecast and Assessment to the Year 2000

by Kensall D. Wise
 Kan Chen
 Ronald E. Yokely

Published by John Wiley & Sons, N. Y.
 Available through the New York City Public Library
 Mid-Manhattan Library
 8 East 40th Street, N. Y. N. Y. 10016
 Library Catalogue #001.6404W

If you're engineering oriented you will find the entire book loaded with resource information.

If you're computer software oriented you will find some specific chapters to be interesting and gives you directions and resources to seriously consider for your future.

If you're interest in computers is general - the conclusions or summaries of each chapter and the entire last chapter "Policy Implications" which includes monitoring microcomputer developments, computer system planning and manpower training and employment would be of interest.

available from the public library, it could be worthwhile looking at it to help determine the direction you may want to go in and help in deciding what to do in the future.