OSITEMS

VOLUME 3, No. 8 NOVEMBER 1981

OSI-tems Volume 3, Number 8

October/November 1981

This month's editor - Mike Bassman

Have you noticed that OSI-tems has A) been getting less frequent, and B) smaller. Not surprisingly, one may cleverly deduce that this comes from a lack of articles. We've been getting more and more members all the time, so this is no excuse. More specifically, all the articles have been coming from a smaller group of members. If you want this valuable magazine to continue (and on a monthly basis) we really do need more output from more people. So please (!) dust off the crayolas, grip firmly between the teeth, and make some marks on the back of your OTB form.

One interesting note: we have many columns of the type 'From the front desk', 'From the workbench', 'From the keyboard', etc., etc. While we use and appreciate all articles, contributions from items of funiture and machinery are somewhat disconcerting.

CONTENTS

Editor for the December issue: Ugo Re
Articles for the December issue should be in by Thanksgiving.
Drop off articles at the store or mail to:

Polk's Computer Dept.
314 5th Ave.
NY, NY 10001

OSI-tems is published by OSUNY (Ohio Scientific Users of New York), a non profit group. Membership (including 12 issues of OSI-tems) is $10 locally, $15 for members whose issues must be mailed.

From the Front Desk

by Ugo V. Re'

I would like to take this opportunity to welcome all our new members. As many of you do not live in the New York City area, OSI-tems will be our only means of communications.

I hope that all of you have had a chance to read through the back issues of OSI-tems. While reading OSI-tems you may have noticed that each month one of our members volunteers to be the editor for an issue of OSI-tems.

The function of the editor is to collect all of the material that has been submitted during the month by our members, and binds it into an issue. The editor should not have to write any of the articles for that issue, as ideally, our members will have submitted enough to put together an issue.

As a new member you can help in the sharing of information and in issuing OSI-tems. Put down your ideas, product reviews, findings, programs, etc. on a piece of paper and send it to me at the following address and I will see that it appears in an issue of OSI-tems.

Ugo V. Re'
Room 2510
32 6 Ave.
New York, N.Y. 10013

OSUNY
OSI-tems
314 5 Ave.
New York, N.Y. 10001

TO ALL MEMBERS

What's happening? Have all our ideas dried up ? Has everyone stopped using their computers?

In the last few months we have had problems issuing a monthly OSI-tems due to a lack of articles. We have a large membership, however some members have never submitted any articles.

How about you? Have you purchased a program or piece of hardware recently? Write a small article explaining how it works and how you like it, and get it into OSI-tems. Everyone is interested in your article and your ideas, even if it is a revision of a previous article or experiment.

Write that article and get into OSI-tems !!

** FLASH **
****
A new member, Kerry Lourash has submitted an article.
See "ERR PRINT" elsewere in this issue.

EPSON MX-80 SCREEN DUMP

By Mike Cohen

For anyone who bought ***GRAFTRAX-80*** for their **EPSON**, here is a program to dump a screen full of OSI graphics. Although it is written in FORTH, the routine is fairly simple. In addition, the hex dump of the GRAFTRAX format character codes should be useful to anyone who wishes to write a similar program.

The hardest part of the screen dump was making up the table of character values. To print each character, 8 bytes must be sent to the printer containing the bits that represent the dot pattern of that character. In the case of GRAFTRAX, the top dot represents the '128' bit, the next one '64' and so on, down to the '1' bit which is 1 row from the bottom wire of the print head.

Since only 8 bits can be output, the bottom wire isn't directly accessible. Therefore, in the graphics mode the line spacing must be set to 8/72", which is equivalent to no space between lines. Unfortunately, when the line spacing is changed, the **EPSON** forgets the top of form position. Therefore, it is necessary to count lines carefully when finished and reset the number of lines per page when it reaches the top of form position.

In the FORTH programs, the word (SD) does the actual screen dump. SDUMP enables the printer output, sets the line spacing for graphics, calls (SD) to do the dump, and finally resets the printer to the proper top of form (SDUMP expects the printer to be at top of form when executed). CHR$ returns the address of the character data when given a character. PCHR uses CHR$ to get the address and sends 8 bytes of character data to the printer.

The remaining words in screens #35 and #36 create graphics patterns and demonstrate the use of SDUMP. The words in screen #33 are used to enter the character data, and assume disk blocks 25 and 26 are used to store the patterns.

```
76FA  0 E7 42 FF FF 42 E7  0  0 E7 42 7E 7E 42 E7  0
770A 81 42 3C  0  0 3C 42 81 81 42 24 24 24 24 42 81
771A  A  6 12  E  E 12  6  A  0  2  7  7  7  7 1F 9F
772A FF 1F  F  7  7  7  F  E  E  F  7  7  7  F 1F FF
773A 9F 1F  7  7  7  7  2  0  0 20 20 24 24 3E 26 26
774A  E 1E 1E 36 36 10 10 10 10 10 10 36 36 1E 1E  E
775A 26 26 3E 24 24 20 20  0  0 38 7C FC FF 7C 38 10
776A  0  F 1C 3F 3D 1F  F  0 1F 34 7F F5 FF 7F 35 7F
777A  0 20 41 FF FF 41 20  0  8  4  E 1D B8 F0 E0 F0
778A 3C 18 18 18 18 5A 3C 18 10 20 70 B8 1D  F  7  F
779A  0  4 82 FF FF 82  4  0  F  7  F 1D B8 70 20 10
77AA 18 3C 5A 18 18 18 18 3C F0 E0 F0 B8 1D  E  4  8
77BA  0 20 50 50 7E 50 7E  0  0  C  8  8  8  8 18  0
77CA  0  8 7E 4A 7E  8  0  0  0  0 7E 42 7E  0  0  0
77DA  0 44 4E 5F 44 44 7C  0 11 22 44 88 88 44 22 11
77EA  0 60 50 4F 4F 50 60  0 88 44 22 11 11 22 44 88
77FA  0  0  0  0  0  0  0  0  0  0  0 FA  0  0  0  0
780A  0  0 E0  0 E0  0  0  0  0 28 FE 28 FE 28  0  0
781A  0 24 54 FE 54 48  0  0  0 C4 C8 10 26 46  0  0
782A  0 6C 92 6A  4  A  0  0  0  0  0 E0  0  0  0  0
783A  0 38 44 82  0  0  0  0  0  0  0 82 44 38  0  0
784A  0 44 28 FE 28 44  0  0  0 10 10 7C 10 10  0  0
785A  0  0  2  C  0  0  0  0  0 10 10 10 10 10  0  0
786A  0  0  0  2  0  0  0  0  0  4  8 10 20 40  0  0
787A  0 7C 8A 92 A2 7C  0  0  0  0 42 FE  2  0  0  0
788A  0 46 8A 92 92 62  0  0  0 84 82 92 B2 CC  0  0
789A  0 18 28 48 FE  8  0  0  0 E4 A2 A2 A2 9C  0  0
78AA  0 3C 52 92 92 8C  0  0  0 80 8E 90 A0 C0  0  0
78BA  0 6C 92 92 92 6C  0  0  0 62 92 92 94 78  0  0
78CA  0  0  0 28  0  0  0  0  0  0  2 2C  0  0  0  0
78DA  0 10 28 44 82  0  0  0  0 28 28 28 28 28  0  0
78EA  0  0 82 44 28 10  0  0  0 40 80 9A A0 40  0  0
78FA  0 7C 82 BA 9A 72  0  0  0 3E 48 88 48 3E  0  0
790A  0 FE 92 92 92 6C  0  0  0 7C 82 82 82 44  0  0
791A  0 FE 82 82 82 7C  0  0  0 FE 92 92 92 82  0  0
792A  0 FE 90 90 90 80  0  0  0 7C 82 82 8A 8E  0  0
793A  0 FE 10 10 10 FE  0  0  0  0 82 FE 82  0  0  0
794A  0  4  2  2  2 FC  0  0  0 FE 10 28 44 82  0  0
795A  0 FE  2  2  2  2  0  0  0 FE 40 30 40 FE  0  0
796A  0 FE 20 10  8 FE  0  0  0 7C 82 82 82 7C  0  0
797A  0 FE 90 90 90 60  0  0  0 7C 82 8A 84 7A  0  0
798A  0 FE 90 98 94 62  0  0  0 64 92 92 92 4C  0  0
799A  0 80 80 FE 80 80  0  0  0 FC  2  2  2 FC  0  0
79AA  0 F8  4  2  4 F8  0  0  0 FE  4 18  4 FE  0  0
79BA  0 C6 28 10 28 C6  0  0  0 C0 20 1E 20 C0  0  0
79CA  0 86 8A 92 A2 C2  0  0  0 FE FE 82 82 82  0  0
79DA  0 40 20 10  8  4  0  0  0 82 82 82 FE FE  0  0
79EA  0  8 10 20 10  8  0  0  0  2  2  2  2  2  0  0
79FA  0  0  0  0  0  0  0  0  0 1C 22 22 14 3E  0  0
7A0A  0 FE 14 22 22 1C  0  0  0 1C 22 22 22 22  0  0
7A1A  0 1C 22 22 14 FE  0  0  0 1C 2A 2A 2A 18  0  0
7A2A  0  0 10 7E 90  0  0  0  0 38 45 45 29 7E  0  0
7A3A  0 FE 20 20 20 1E  0  0  0  0 12 5E  2  0  0  0
7A4A  0  0  1  1 5E  0  0  0  0 FE  8 18 24 42  0  0
7A5A  0  0 82 FE  2  0  0  0  0 3E 20 1E 20 1E  0  0
7A6A  0 3E 20 20 20 1E  0  0  0 1C 22 22 22 1C  0  0
7A7A  0 7F 28 44 44 38  0  0  0 38 44 44 28 7F  0  0
7A8A  0 3E 10 20 20  0  0  0  0 12 2A 2A 2A 24  0  0
7A9A  0 20 20 7E 20 20  0  0  0 3C  2  2  2 3E  0  0
7AAA  0 30  C  2  C 30  0  0  0 3C  2  4  2 3C  0  0
7ABA  0 22 14  8 14 22  0  0  0  0 3D  5  5 3E  0  0
7ACA  0 22 26 2A 32 22  0  0  0  0 10 6C 82 82  0  0
7ADA  0 82 82 6C 10  0  0  0  0  0  0 EE  0  0  0  0
7AEA  0 10 10 54 10 10  0  0  0  8 10 10 10 20  0  0
```

DATA FOR OSI CHARACTER PATTERNS IN GRAFTRAX-80 FORMAT

Characters

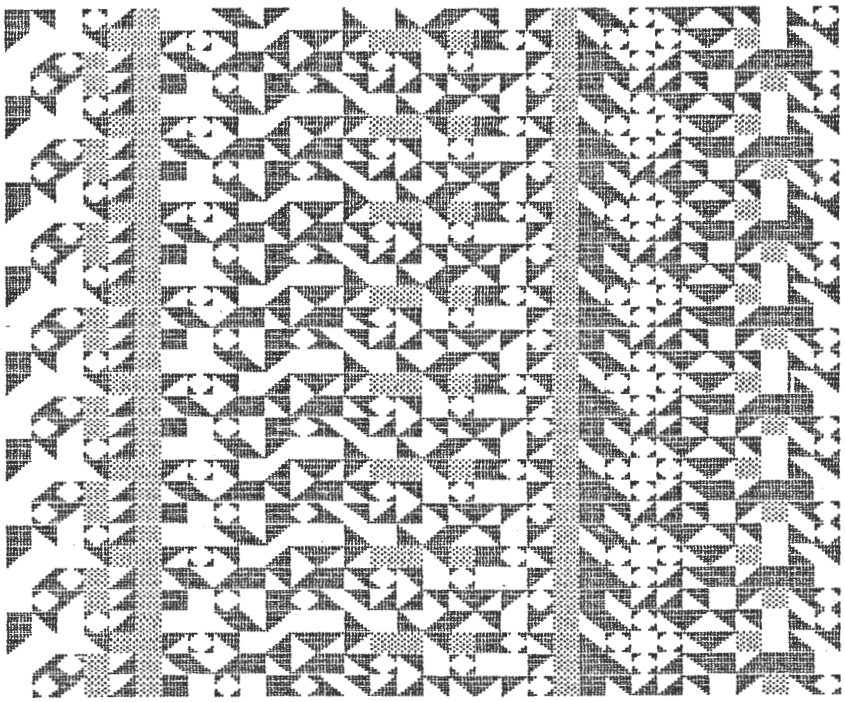
Ø - 127

$00 - $7F

```
7AFE  1  1  1  1  1  1  1  1  2  2  2  2  2  2  2  2
7B0E  4  4  4  4  4  4  4  4  8  8  8  8  8  8  8  8
7B1E 10 10 10 10 10 10 10 10 20 20 20 20 20 20 20 20
7B2E 40 40 40 40 40 40 40 40 80 80 80 80 80 80 80 80
7B3E FF  0  0  0  0  0  0  0  0 FF  0  0  0  0  0  0
7B4E  0  0 FF  0  0  0  0  0  0  0  0 FF  0  0  0  0
7B5E  0  0  0  0 FF  0  0  0  0  0  0  0  0 FF  0  0
7B6E  0  0  0  0  0  0 FF  0  0  0  0  0  0  0  0 FF
7B7E  3  3  3  3  3  3  3  3 C0 C0 C0 C0 C0 C0 C0 C0
7B8E FF FF  0  0  0  0  0  0  0  0  0  0  0  0 FF FF
7B9E 18 18 18 18 18 18 18 18  0  0  0 FF FF  0  0  0
7BAE  7  7  7  7  7  7  7  7 E0 E0 E0 E0 E0 E0 E0 E0
7BBE FF FF FF  0  0  0  0  0  0  0  0  0  0 FF FF FF
7BCE  F  F  F  F  F  F  F  F F0 F0 F0 F0 F0 F0 F0 F0
7BDE FF FF FF FF  0  0  0  0  0  0  0  0 FF FF FF FF
7BEE 1F 1F 1F 1F 1F 1F 1F 1F 3F 3F 3F 3F 3F 3F 3F 3F
7BFE 7F 7F 7F 7F 7F 7F 7F 7F FF FF FF FF FF FF FF FF
7C0E F8 F8 F8 F8 F8 F8 F8 F8 FC FC FC FC FC FC FC FC
7C1E FE FE FE FE FE FE FE FE  0  0  0  0  F  F  F  F
7C2E  0  0  0  0 F0 F0 F0 F0  F  F  F  F  0  0  0  0
7C3E F0 F0 F0 F0  0  0  0  0 F0 F0 F0 F0  F  F  F  F
7C4E  F  F  F  F F0 F0 F0 F0  0  0  F  F  F  F  0  0
7C5E  0  0 F0 F0 F0 F0  0  0  0  0  0  0 3C 3C 3C 3C
7C6E 3C 3C 3C 3C  0  0  0  0 FF FE FC F8 F0 E0 C0 80
7C7E  1  3  7  F 1F 3F 7F FF 80 C0 E0 F0 F8 FC FE FF
7C8E FF 7F 3F 1F  F  7  3  1  7 17  F  F  7  F 1F 3F
7C9E 3F FF 3F  7  F  F 16  4  4 16  F  F  7 3F FF 3F
7CAE 3F 1F  F  7  F  F 17  7 A0 50 A0 50 A0 50 A0 50
7CBE  A  5  A  5  A  5  A  5 AA 55 AA 55  0  0  0  0
7CCE  0  0  0  0 AA 55 AA 55 AA 55 AA 55 AA 55 AA 55
7CDE 81 42 24 18 18 24 42 81  1  2  4  8 10 20 40 80
7CEE 80 40 20 10  8  4  2  1 80 40 20 10 10 20 40 80
7CFE  0  0  0  0 18 24 42 81  1  2  4  8  8  4  2  1
7D0E 81 42 24 18  0  0  0  0 10 10 20 20 40 40 80 80
7D1E  1  1  2  2  4  4  8  8 80 80 40 40 20 20 10 10
7D2E  8  8  4  4  2  2  1  1  0  0  0  0 C0 30  C  3
7D3E C0 30  C  3  0  0  0  0  0  0  0  0  3  C 30 C0
7D4E  3  C 30 C0  0  0  0  0  0  0  0  0 F0 10 10 10
7D5E  0  0  0  0  F  8  8  8  8  8  8  F  0  0  0  0
7D6E 10 10 10 F0  0  0  0  0 80 80 80 80 80 80 80 FF
7D7E  1  1  1  1  1  1  1 FF FF  1  1  1  1  1  1  1
7D8E FF 80 80 80 80 80 80 80 7C F4 74 3C 1C 1C 1C 3C
7D9E 74 74 74 34 1C 1C 1C 7F 7F 1C 1C 1C 34 74 74 74
7DAE 3C 1C 1C 1C 3C 74 F4 3C 18 18 18 F8 F8 18 18 18
7DBE  0  0  0 FF FF 18 18 18 18 18 18 1F 1F 18 18 18
7DCE 18 18 18 FF FF  0  0  0 18 18 18 FF FF 18 18 18
7DDE  0  0  0  0 C0 20 10 10  0  0  0  0  3  4  8  8
7DEE  8  8  4  3  0  0  0  0 10 10 20 C0  0  0  0  0
7DFE  0  0  0  0 3C 42 81 81 81 81 42 3C  0  0  0  0
7E0E 3C 42 81 81 81 81 42 3C 3C 3C 42 42 81 81 81 81
7E1E 81 81 81 81 42 42 3C 3C 70 F8 FC 7F 7F FC F8 70
7E2E  C 1C CD E3 E3 CD 1C  C 18 3C 79 FF FF 79 3C 18
7E3E 18 3C 7E FF FF 7E 3C 18 E7 C3 81  0  0 81 C3 E7
7E4E 18 18 3C 3C 7E 7E FF FF FF FF 7E 7E 3C 3C 18 18
7E5E  C 1C 39 FF FF 39 1C  C 3C 18 DB FF 7E 3C 18 18
7E6E 30 38 9C FF FF 9C 38 30 18 18 3C 7E FF DB 18 3C
7E7E  0  4  9 1E  9  4  0  0  0  4  9 1E  B  C 30  0
7E8E  0  0  0  3 FF  F  F 1F  0 40 20 13  F  F 1F 1F
7E9E  8  8  8  B  F  F 1F 1F 1F  F  F FF  3  0  0  0
7EAE 1F 1F  F  F 13 20 40  0 1F 1F  F  F  B  8  8  8
7EBE  0  0 1F  E 7E  E 1F  0  0  4  C 1F 3E 1C 28 40
7ECE 44 7C 7C 7C 54 10 10  0  0 20 30 F8 7C 38 14  2
7EDE  0  0 F8 70 7E 70 F8  0  0  2 14 38 7C F8 30 20
7EEE  0 10 10 54 7C 7C 7C 44 40 28 1C 3E 1F  C  4  0
```

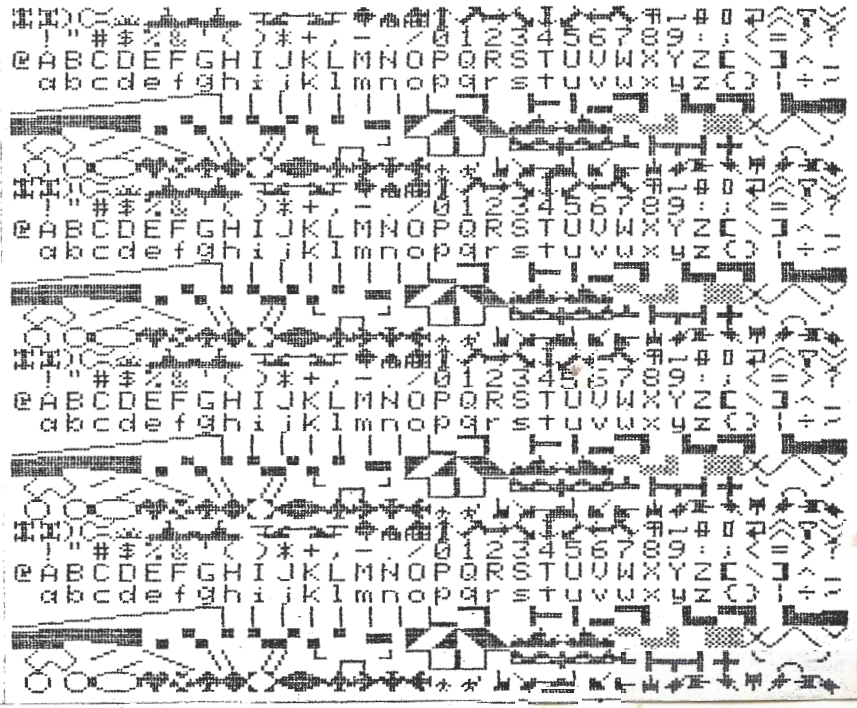
Characters
128 - 255
$80 - FF

```
SCR # 33
  0 ( GRAPHICS ENTRY )
  1 FORTH DEFINITIONS DECIMAL
  2 : CHAR-NUM ( RETURN ADDRESS OF CHARACTER )
  3     8 * 1024 /MOD 25 + BLOCK + ;
  4 : NUM ( READ A NUMBER )
  5     CR ." >" QUERY 1 TEXT PAD NUMBER DROP ;
  6 : CHR ( INPUT A CHARACTER )
  7     CHAR-NUM ( GET ADDRESS )
  8     ." ENTER DATA IN BINARY" 2 BASE !
  9     8 0 DO
 10      DUP I + NUM SWAP C!
 11     LOOP
 12     DECIMAL
 13     DROP UPDATE ;
 14 ;S
 15


SCR # 34
  0 ( SCREEN DUMP )
  1 FORTH DEFINITIONS DECIMAL
  2 25 CONSTANT CHRGEN ( BLOCK CONTAINING CHARACTER DATA )
  3 : CHR$   8 * 1024 /MOD CHRGEN + BLOCK + ;
  4 : PCHR ( PRINT CHARACTER IN GRAPHICS MODE )
  5   CHR$ DUP 8 + SWAP DO I C@ EMIT LOOP ;
  6 : (SD) ( DUMP SCREEN )
  7   54272 53248 ( C1P VALUES ) DO
  8      CR 27 EMIT 75 EMIT 0 EMIT 1 EMIT ( FOR C1P )
  9      I DUP 32 + SWAP ( 32 CHR LINES ) DO
 10       I C@ PCHR
 11      LOOP
 12   32 +LOOP ( FOR 32 CHR LINE )
 13   CR
 14 ; -->
 15


SCR # 35
  0 ( SDUMP CONT. )
  1 FORTH DEFINITIONS DECIMAL
  2 : SDUMP ( SCREEN DUMP )
  3   1 8994 C! ( TURN PRINTER ON )
  4   CR 27 EMIT ASCII A EMIT 8 EMIT ( SET LINE SPACING )
  5   (SD) ( DUMP A SCREEN OF GRAPHICS )
  6   27 EMIT ASCII 2 EMIT ( NORMAL SPACING )
  7   12 EMIT 11 0 DO CR LOOP 27 EMIT 67 EMIT 66 EMIT
  8  2 8994 C! ; ( TURN PRINTER OFF )
  9 : FILL-SCREEN ( FILL THE SCREEN WITH ALL GRAPHICS CHARACTERS )
 10    54272 53248 DO
 11       I 255 AND I C!
 12    LOOP ;
 13 : DEMO FILL-SCREEN SDUMP ;
 14 ;S
 15

Z-FORTH    07 01 1981
```



```
SCR # 36
  0 ( GRAPHICS DEMO ) FORTH DEFINITIONS DECIMAL
  1 175 VARIABLE C#  32 1 C# + C! ( INITIALIZE ARRAY )
  2  176 C, 177 C, 178 C, 187 C, 161 C, 233 C,
  3 : R% ( GENERATE PSEUDO-RANDOM NUMBER )
  4      8996 C@ 9 * 7 + 256 MOD DUP 8996 C! ;
  5 : RND R% 29 * R% / 7 AND ; ( MOD 7 RND NUMBER )
  6 : PICK-CHR C# RND + C@ ; ( GET RANDOM CHARACTER )
  7
  8 : FILL.SCREEN
  9   54272 53248 DO
 10     PICK-CHR I C!
 11   LOOP ;
 12
 13 : SD FILL.SCREEN SDUMP ;
 14 ;S
 15
```

OSI GRAPHICS
CHARACTER SET →
BY EPSON MX-80



HARDWARE REVIEW

Lewis Computer Systems Light Pen

A light pen is one of those computer accessories that is often very impressive to the uninitiate in computer hardware as well as an input device that can solve some thorny system problems.

A couple of articles on light pens have appeared in the popular journals within the last year, but for those of you who have not had the time (or the inclination) to read them, I will give a synopsis of the principles involved.

Computer displays fall into two catagories, those where the processor knows which raster is being addressed, and those which don't. A raster is the individual dot on the CRT (or TV) display and is not to be confused with the dots that are generated by the video controller chips in some 'hi-res' machines like the Apple, Atari, VIC, or SEB boards. The dots that these machines generate, even in their highest resolution modes, are made up of several rasters. A standard-aspect TV has about 4000 rasters horizontally by about 3000 rasters vertically.

Most of the very early TV displays and some of the very simple ones like Don Lancaster's "Cheap Video" require that the processor generate the raster scan and therefore fall into the first catagory of systems which know which raster is being addressed. Many of the most sophisticated vector graphic displays (like the Tektronics) also fall into this catagory. Most of the current microcomputer displays fall into the second catagory. In our Challengers, for instance, we place codes for characters into a section of memory, then circuitry consisting of a ROM, dot clock, and various logic gates takes over and, independent of the processor, generates the TV signal.

All of this background is necessary if one is to understand how a light pen operates. The operation of a light pen is important if one is to decide if it will fill his need and if he can program it. In the case of the first type of display, the output of the light pen can be tied to an interrupt line. Since the processor always knows which part of the screen is being scanned, when the interrupt is generated, the processor immediately knows which part of the screen was pointed to. This type of light pen requires little software support. In the case of our type of display, the hardware can be somewhat similar, but some action on the part of the processor will be necessary to find out where on the screen the pen is pointing to.

The usual method is for the processor to flash points or squares on the screen. As an individual square is alternately turned on and off, the output of the pen is sampled. If the pen indicates presence or absence of light in step with the flashing square, then it is pointed at that square. If not, then the next square on the screen is flashed until a correspondence is found. This method is limited in two ways. First the resolution is limited to the smallest dot or square that can be flashed, in our case the quarter block is probably the smallest character usable for this purpose. Second, since many on/off cycles are necessary to make an accurate identification and since the squares must be flashed at a rate considerably slower than the TV scan rate, if

quick responce to the light pen is important, relatively few places on the screen can be checked.

None of the above limitations is so severe as to preclude many useful applications for light pens on our Challengers. Chess/Checker like gaming displays, and even serious "applications" type programs can be enhanced by light pen input. Non-computerist computer users who may be intimidated by a keyboard might respond more positively to light-pen input if for no other reason than the novelty of it.

The only OSI-commpatible, ready-made light pen that I know of is made by Lewis Computer Systems and marketed by Farrager and Associates. While this unit has some drawbacks, if you don't care to 'roll your own', then it will have to do. Unless I miss my guess, the heart of the Lewis unit is an Light Activated Silicon Controlled Rectifier (LASCR). This is a device which, essentially, is a light-activated electronic switch. The LASCR is housed in what appears to be an empty, unmarked BIC pen body (yech!). This is in turn connected to a nice coiled cord, which terminates in a standard 1/4 in. three conductor plug and jack set. Only three wires must be hooked up inside your computer. Clear instructions and a set of diagrams should make installation possible by almost anyone, but soldering is required. The three connections are to +5V, ground and the shift-lock key. The pen works by faking the closure of the shift-lock key, and therein lies one of its problems.

A tic-tac-toe demo program is included with the pen. The program, which is supplied on both tape and hard copy, loaded with no problems, provided that the light pen is not inserted into its jack. Once the pen is inserted in the jack, it inhibits the function of the shift-lock key making keyboard entry impossible. A program has to be loaded, RUN and all non-light pen input accomplished before the pen can even be plugged in. This is not too much of a problem if all of a program's input can be by light pen, but intermixing keyboard and light pen input becomes a drudge when the pen must be unplugged repeatedly. This can probably be eliminated by using the pen in conjunction with some little used key rather than the shift lock.

Instructions are included dealing with how to adjust the TV or monitor that you are using to get the best performance from your pen. Reading between the lines in these instructions gives insight into how the pen works. On running the demo program, one sees the limitations of a software-scanned screen. The nine 'target' squares on the tic-tac-toe board flash for several seconds before the program zeros-in on one square. Then a couple of extra flashes are made just to assure accuracy before that square is replaced by the player's marker. The screen set-up/scan routine is written as a general purpose subroutine which with little or no modification could be used in your own programs.

The price for the Lewis pen was about $29.95 when I got mine several months ago. While on first glance you may look at the hardware, software, and documentation that you get and feel that the price is a little excessive, I guess from a business standpoint it really isn't. If you made your own similar unit, it would probably cost you one-third as much and then you'd still have to write some software.

Lewis Computer offers an installation service ($15.00) and can be reached at:

Lewis Computer Systems
804 W. North
Moravia, IA 52571

As I mentioned earlier, the pen can also be had from:

Farrager & Associates
7635 W. Bluemound Rd.
Milwaukee, WI 53213

Terry Terrance

Kerry Lourash
1220 North Dennis
Decatur, Ill.
62522

Sept. 8

Ugo;

Glad to hear from you. Please enroll me as a member of OSUNY. You sent me several partial articles that were most interesting. One of them, a program designed to print the ClP error messages correctly, was so intriguing that I wrote my own version. It's patched into the output routine via the output vector. This version checks the stack to see if the calling routine is at $A252. The error message is printed without the normal CR/LF, which will save some space on the small ClP display.

```
10 *=$1500
20      PHA   SAVE A
30      TXA   SAVE X
40      PHA
50      TSX   STACK TO X
60      LDA $107,X  SEE IF
70      CMP #$52    CALLING
80      BNE A1      ROUTINE
90      LDA $108,X  IS $A252
100     CMP #$A2
110     BEQ A2      YES, BRANCH
120 A1  PLA   RESTORE X
130     TAX
140     PLA   RESTORE A
150     JMP $FF69
160 A2  PLA   RESTORE X
170     TAX
180     PLA   RESTORE A
190     JSR $A8E3  PRINT '?'
200     LDA $A164,X  1ST CHAR
210     JSR $A8E5  PRINT IT
220     LDA $A165,X  2ND CHAR
230     AND #$7F   ZERO HI BIT
240     JMP $A25F  TO ERROR
                   ROUTINE
```

BBS DIRECTORY

By Mike Cohen

Here is a list of some of the free BBS numbers in the New York City area. These numbers can be used by anyone with a 300 baud modem and any form of terminal software. To log on to any of these systems, just dial the number and when you get the modem tone, hit <RETURN> 1 or 2 times. The systems will ask you for your name , terminal characteristics, and sometimes your location and phone number. None of these systems require accounts or passwords, but some have an option of assigning a password to regular users for private messages.

All of these are single user systems, so don't be discouraged if the number is busy - just keep trying. Most of these systems are running on either Apples or TRS-80s and use similar commands. The most common commands are 'S' for scan message summaries, 'R' for retreive full messages, 'B' for bulletins, and 'I' for system information.

Here is the list by area code:

212

ABBS: The French Connection.....799-9577
ABBS: The Big Apple.....362-1040

These are both very similar (the same?) systems, being run by Roger Kaplan and Adrian Steckel on APPLE II's. The software is still being perfected, so the systems tend to crash pretty often. Right now, it's sort of flakey, but shows potential.

Comm-80, Queens.....897-3392

A fairly new system running on a TRS-80. This one has all the standard features including bulletin, information, and What's New sections. Continuously being improved, and pretty reliable.

BBS DIRECTORY

Connection-80, Bronx.....933-9459
Connection-80, Woodhaven.....441-3755
Connection-80, Staten Island.....442-3874

These are all very similar systems running on TRS-80s. Connection-80 is the most popular BBS program and has many additional features including a Downloading section, which can transmit ASCII files and BASIC programs to any smart terminal equiped with memory buffering. The Woodhaven system has unusually large Bulletin and Downloading sections, including many BASIC programs for non-TRS-80 systems.

Modem Over Manhattan (*MOM*).....245-4363

This is one of the most popular bulletin boards and the busiest one in the area, so just keep trying. Run by Larry Kelly, this system has a very large following and has an excellent magazine section with articles on Phone Phreeking and many other topics plus sports predictions by Cazzie Mack. A downloading section was recently added, although I haven't seen it yet.

PMS McGraw-Hill.....997-2488

Another Apple, but with much more reliable software. This is a very nice system with all the usual features plus a feature section, which allows browsing and purchasing Mcgraw-Hill books. This system is the NYC branch of People's Message System, a popular APPLE Bulletin Board with systems throughout the country.

RCP/M AABB.....787-5520

This is a remote CP/M system with an emphasis on Astronomy. It features an Astro-bulletin with moon, sun, and planet positions, plus interesting and unusual sightings. Unlike most Remote CP/M systems, however, this one doesn't allow users access to CP/M directly.

914

RCP/M RBBS, Brewster.....279-5693
RCP/M SJBBS, Bearsville.....679-6559 (rb)

Two more Remote CP/M systems, although I've only tried the Brewster, NY system. This one allows access to the CP/M operating system, although with certain security features, for downloading programs and viewing .DOC files. The Bearsville, NY number uses a ring-back system - Dial the number, allow it to ring once, hang up and dial again.

516

CBBS, Long Island.....334-3134

Another Connection-80 system, similar to the others.

CBBS, LICA LIMBS.....561-6590
RCP/M RBBS.....698-8619 (rb)
RCP/M RBBS.....791-1767

I haven't tried these numbers.

From the workbench

by
Claudio Caballero

I haven't done a real project this month, so I've just put together a few useless hardware goodies that everybody probably knows about anyway (Mike said something about saving my kneecaps this month?). They're pretty much in random order, and none require more than a few dollars worth of parts.

1) Cheap inverted video.(Fig.1)
This one is for all you deranged people like me who don't want to bother with fancy hardware mods, but just think that tank games do look much nicer in inverted video. All it does is switch between the regular output of the video shift register and an inverted one. About 2 bucks for the switch. The inverter is U18(9&8) on the 600 series 1;U31(5&6) ,which you might have to install, on the 502; and on the 600 series 2, there's a 7404 (eitherU16orU18) with(1&2) and(3&4) unused somewhere near the left side of the keyboard.

2)Mic. to Aux.
If you've ever come across a tape recorder that has the Mic. input screwed up, but the Aux. input works fine,and wished you could use it with you're OSI, here's how. On C1's, move the wire from P7/J2 to P9/J2. On C2/4's, move the wire from P8/J2 (on the 502) to P10/J2.

3)Rom disable for C2/4's.(Fig.2)
As soon as I throw in my SEB-2 (look for my review this issue), I'd like to be able to get a full 48K of ram for FORTRAN and other neat stuff. This requires that I either pull out my preciuos ROMs, or, disable them. I prefer the latter route since it's easily reversible. Since I can't open my machine while typing this article, I can't show you where the trace is. But, there is a trace that goes to pins 21&18 of all the BASIC ROMs. At one end of this trace there is one of OSI's world renowned(or is it denounced) jumper pads. All ya gotta do is attach the center contact of an SPDT switch to the trace, and either side to either side of the 5v power line (don't forget to cut the jumper!), and voila!!!, you can disable you ROMs in favor of some RAM which you can switch in via a switch connected across the 8K block select jumpers on a 527 board. That's all for this month. Again, if you have questions, see me at the meeting. I may not answer them, but it makes me look important!!!!

Fig. 1

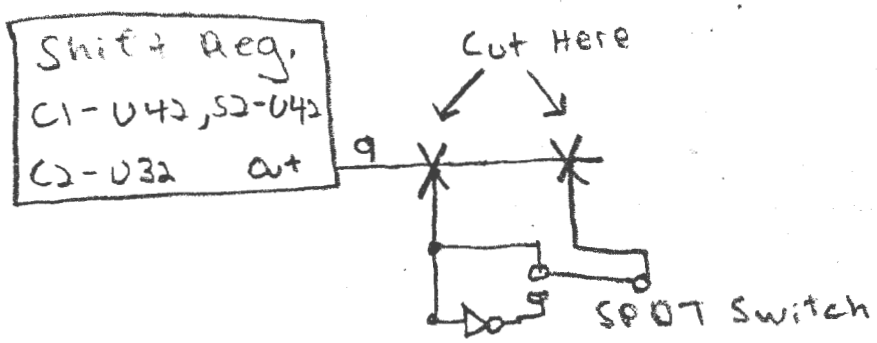
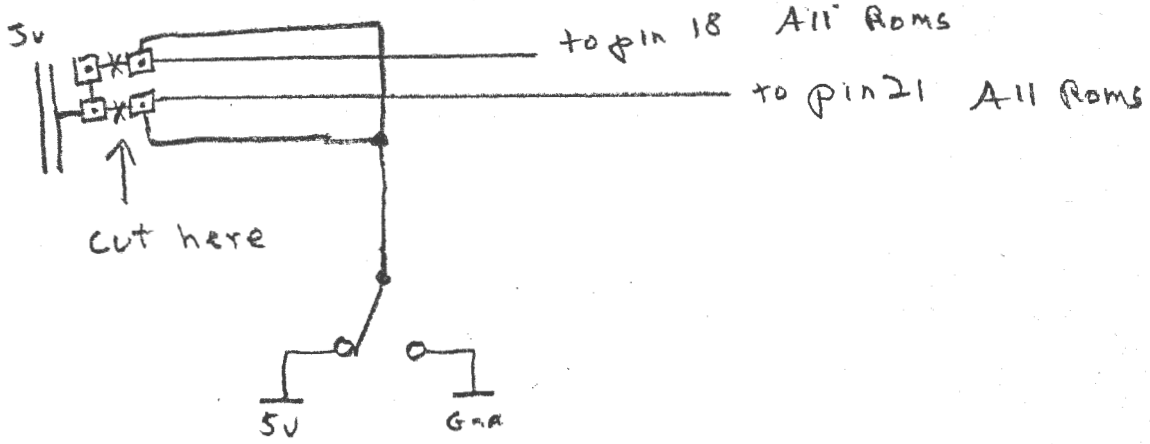


Fig 2.

I think it looks something like this.



OS65D'S VIDEO ROUTINE

by Mike Cohen

Here is a small utility for C1P disk users. This program will adapt the video driver for any non-standard line lengths. The routine is based on the C1P video update given in Elcomp's First Book of OSI.

All constants for the video driver are calculated from the actual line length entered. The variable OF, in line 20 is an offset used to center the display which may need to be changed for some systems. In general, the value of 1 for line lengths less than 32 characters and 3 for more than 32 character lines seem ok.

Line 220 determines if 2K of memory is present and lines 230-270 set the cursor address accordingly. Line 300 sets the control register on Series 2 systems for the proper video mode.

```
10 INPUT"LINE WIDTH";LN
15 LW=32:P=0:IFLN>32THENLW=64:P=1
20 OF=1:IFLW>32THENOF=3
30 REM LN IS LENGTH
40 REM OF CENTERS LINE
50 EX=LW-LN
60 B=INT(EX/2)
70 L=B+OF
80 R=EX-L
90 ST=64+L:EN=ST+LN
100 POKE 9636,ST
110 POKE 9670,EN
120 POKE 9723,LW-1
130 POKE 9725,L-1
140 POKE 9730,EX
150 POKE 9738,LN+L
155 Q=EX:IFQ>0THENQ=Q-1
160 POKE 9743,Q
170 POKE 9766,ST
180 POKE 9770,ST
190 POKE 9783,EN
200 POKE 9736,LW-1
210 POKE 9800,LW
220 POKE 55000,32:IFPEEK(55000)=214THEN300
230 POKE 9667,215: POKE 9677,215
240 POKE 9685,215: POKE 9750,215
250 POKE 9757,215: POKE 9764,215
260 POKE 9776,215: POKE 9786,215
270 POKE 9801,215: POKE 9811,215
300 POKE 55296,P: END
```

TWO SCREENS IN ONE

by WILLIAM J. BALABAN

Is the Superboard/C1P screen size too small for the amount of information that you want to display? With four new chips and a handfull of diodes, I've found a reasonable compromise. It is a second page of screen memory which can be displayed simply by writing to or reading from it. It can be used without destroying the contents of the original page contents. The pages can be flipped back and forth as needed. System messages and the results of print statements will appear on the original page (this is a function of the screen driver routine) and therefore will not destroy anything you were doing on page 2. The added video memory takes advantage of as much of the existing circuity as possible. It was most convenient to locate it in the next 1K block above the existing memory(54272-55295). If you are ever at a loss for memory, the video memory can be used for machine language routines (the screen will display randomly changing characters anytime the computer accesses page 2.

Be aware (!):the modification will require that several existing lands be (shudder!) cut. The new video ram chips will be piggybacked on top of the old video ram chips.

Piggybacking is a cost-effective way of adding memory or any other chips where most of the connections will be in parallel. The new 2114 memory chips are placed on top of the old video rams (align them so that they are in the same direction) so that their pins overlay the pins of the old chips. Pin 8 of the upper chips must be bent at a 45-90 degree angle (it will be the only top pin not in contact with a lower pin. All other pins must be soldered to the lower corresponding pins. With the double decker chips in their sockets, solder a wire between the two lifted pins.

The two control chips may be located in the unused experimental area of the Superboard/C1P. The wire should be thin and insulated (wire-wrap wire is a good choice). The wires should be kept short and neat; a few appropriately placed drops of glue will keep them in their place.

The circuit works by setting the 74112 flip-flop whenever the original 1K video memory is selected (through a read or write to one of its' locations) and by resetting it when the new video memory is "or'd" with the two flip-flop outputs in order to provide two signals (one goes to pin 8 of the upper set, the other goes to pin 8 of the lower set) to select the required 1K block of memory. The diodes D5 through D8 are used to cause the data direction bus (U24 and U25) to respond properly for both pages of memory; without them the computer could not read or write to the new memory.

OS65D POKES

By Mike Cohen

There are several little-known and very useful pokes which make 65D Basic's input statement much easier to use. It is possible to change the question mark prompt, accept null input, and even accept strings containing leading blanks, commas, and colons.A few of these are documented in the 65D manual. Here are some of the locations and their purpose:

2888, 8922	Poke 0 into both of these to allow null input, which will return 0 or a null string if you just hit return. Poke a 27 for normal input.
2972	Poke 13 to allow colons on input otherwise poke 58.
2976	Poke 13 to allow comma on input otherwise poke 44.
2793	This location contains the question mark prompt. Poke it with the ASCII code for the character of your choice.
2683	Poke a zero here to prevent BASIC from echoing a linefeed after each input line. However if the input line is longer than the screen width, it will still force a linefeed. Poke 10 to restore linefeeds.
207	Poke a 36 here to allow leading blanks on input otherwise it must contain 240. CAUTION: This is a part of the CHRGET routine in page zero and when it's poked to allow leading zeros, will cause a syntax error if the program line contains any blanks outside of quotes.

HARDWARE REVIEW

Zenith Data Systems 12" 'Green Screen' Monitor

Here's an item for those of you who may not yet have a monitor or who are considering upgrading your present monitor. Since Zenith now owns Heathkit, they have started the Zenith Data Systems division which sells Heath computer products under the Zenith name. The ZVM-121 monitor, however, seems to be Zenith's contribution to the "Data Systems" line.

The ZVM-121 is housed in a beige plastic, TV-style case (complete with blank area where tuner controls would have been mounted) which, apparently, is color coordinated to match the Apple, but which blends well with the color of C1, C4, and C8 tan cases. To my eye, the styling of this case is better than the other monitors in this price range. Front mounted controls consist of "Black Level" (brightness), Contrast and On/Off. Controls on the rear panel include Horizontal Hold, Vertical Hold, 40/80 character switch, Character Width control, and Vertical Size (the latter adjustable by alignment tool only). Input is accomplished via a standard RCA plug.

Using this monitor is fairly standard, plug it in, connect your computer, adjust the hold controls and you're in business. The only functional difference is connected with the 40/80 character switch. This control seems to be another concession to Apple users. When used with a C1, the 40 character position gives you a display similar (save for the fact that your screen is green) to an OSI monitor or a Leedex. When switched to the 80 character position, the screen scan is compressed so that the screen is not overscanned, and unscanned portions of the screen are now visable on the top, bottom and sides of the display. The net effect of these phony "guard bands" on the C1 display is quite nice, text is nicely centered on the screen; and it seems that the 80 character switch also changes the screen's aspect ratio so that the C1 characters actually come out square. In the 80 character position, 31 lines and 28 columns of the C1's display are fully visable, much better than any other monitor I have ever seen. The green screen seems to be easier on the eyes, as many ads have claimed.

This monitor does have some minor faults. When the video input to the screen is shut off, say by turning off your computer but not the monitor, the screen looks just as it does when the monitor is off. Since the ZVM-121 lacks a pilot light, you could conceivably leave the monitor on without knowing it. I tried this monitor with a C1, C4, and C8. The C1 and C4 produced bright displays, the C8, however, produced a very dim display. I have been having trouble with this C8's video board *and* the problem may well be in the computer. The character width control on the back of my monitor does not seem to function. Like many other monitors made in the Orient (the ZVM-121 is made in Taiwan), quality control is a little lax. My monitor does not have all of the contrast that I would like, but I have seen other ZVM-121's that have plenty of contrast.

The big advantage to buying a Zenith product as opposed to other brands

is the service. The monitor was accompanied by a six page list of supposed "Zenith Data Systems" service centers. Since the first monitor that I was shipped was "dead on arrival", I had a chance to try out the service. I first called the local TV service center listed as a data service center, and upon explaining my problem with this product got the response, "What's a monitor?" Not wanting to give the service man an education in TV (after all, monitors are used in applications other than computers), I called the local Heathkit service center which was also on the list. Here at least the technician knew what I was talking about, but they did not yet have the schematics for this unit as it was so new, and they would not touch it until they did. I'm sure Zenith will soon realize that they cannot list every one of their TV repair shops as data service centers and the service thereafter should be top-notch.

At a retail price of $149.00 the ZVM-121 is less than other popular b/w monitors and much less than similar "green screen" monitors. It certainly would be a good choice for anyone looking for a monitor.

Terry Terrance

HARDWARE REVIEW: D&N's Disk Switch

---

by Mike Cohen

This $29 kit, which was described in last month's MICRO, is the one modification all OSI 5" disk users should make. It could prevent a lot of disk drive problems and worn out disks.

One of the biggest problems with OSI's 5" floppy systems is that the disk drive's motor runs continuously and the head remains pressed against the disk most of the time. This causes much unnecessary wear both on the disk drive and on the disk. OSI, in an attempt to cut cost by using the same circuitry in both 5" and 8" disk systems, totally eliminated the motor control both in their hardware and software.

The Disk Switch solves the problem by turning off the disk motor and lifting the head when the disk isn't being accessed. In addition, it can also prevent non-existing drives C and D from being selected in a 2 drive system and causing it to hang. The operation of the Switch is invisible to the operating system and it requires no software changes.

However there is one minor drawback to the Disk Switch-speed. It adds about one second to disk access time while it waits for the disk to come up to full speed. After the disk is accessed, it will remain on for about 2 seconds allowing further access with no delay.

There are 2 additional cautions if you're using it on a C1P. First, pin 14 of the A13 board isn't connected to +5 volts, which is what the Disk Switch uses for power. This is easily fixed by jumpering the pin, which is unconnected, to the nearest power trace on the 610 board. A second problem is that it sometimes causes HEXDOS to hang up if the real-time clock is used, since it seems to interfere with HEXDOS's software motor control.

In all, the Disk Switch is a useful addition which will extend the life of disks and of the disk drive itself. The disk drive will run cooler and the motor is less likely to wear out if it's kept on for long periods.

Source Update

Mike Bassman

Lately OSI has gone into spastic fits. No one is sure what they are going to do, least of all themselves. Oddly enough, there has been a sizable increase in new products. Listed below are some.

Systems International Incorporated 15920 Luanne Drive Gaithersburg, Maryland 20760	These people make systems that are software compatible with the OSI.
Interesting Software Consultants 15217 Campillos Road La Mirada, CA 90638	Interesting personal software for Challengers. Write for catalog.
Darby Software 692 Cordelia Drive Galloway, Ohio 43119	Disk cataloger.
Ron Lashley Software 2934 W. Missionwood Circle Miramar, FL 33025	Games, etc.
Northern Micro 29 Moorcroft Park New Mill Huddersfield, England	Selling many additional ROMs for OSI. They sound nice... Also a C1 RAM+ROM board.
Dee Products 150 Birchwood Road Lake Marian, IL 60110	RS232 kit & case kit for Superboard.
Universal Systems 2020 W County Rd. B. Roseville, MN 55113	Good utilities for disk systems, but expensive.
Micro-Interface 3111 So. Valley View Blvd. Las Vegas, Nevada 89102	Roms and boards for OSI. IEEE488 interface & ROM terminal are original.
Dragon Byte Software 4066 Polaris Avenue Joshua Tree, CA 92252	Educational software and floppy disk maintenance.
Responsive Computer Technology PO Box 719 Silver Spring, MD 20901	Disk utilities for c2/c3.

Source Update continued on next page.

...Source Update continued

D&N Micro Products 3684 N. Wells st. Ft. Wayne, Indiana 46808	OSI compatible hardware. I have heard many nice things about their products.
Modular Systems P.O. Box 16A Oradell, NJ 07649	They make a doubler for OSI drives, among other things.
Pretzelland Software 2005 D Whittaker Rd. Ypsilanti, MI 48197	Games for OSI, icluding sound on C1 Series 2. See review next issue.
PETTED P.O. Box 21851 Milwaukee, Wisconsin 53221	Games.
Computer Science Engineering 57 Beals St., Rm 57-12 Brookline, MA 02145	Named file system for SAVE?LOAD commands, includes token loading for speed.
Software Plus 1818 Ridge Ave. Florence, Alabama 35630	Games.
Teaco, Inc. P.O. Box E 2117 Ohio st. Michigan City, IN 46360	Hard disk interface (Shugart SA4008) for C3-C or comparable OSI system.
Logical Software P.O. Box 354 Farmington, MI 48024	DBMS for OS65U systems.
Bustek P.O. Box A St. Charles, Missouri 63301	Roms for C1/2. Usual editing plus extended machine language monitor.
BC Software 9425 Victoria Dr. Upper Marlboro, Maryland 20870	Games for the C1.
Computer Business Service P.O. Box 20384 San Jose, CA 95160	Software plus OSI T-shirt.
Vega Enterprises 1564 Locust Ave. Long Beach, CA 90813	Maze games for OSI.

A HARDWARE REAL-TIME CLOCK

by KLAUS ERNST

In the last issue of OSI-tems (Sept. '81 Vol.4 No. 7) Mike Cohen showed how to add a software controlled real-time clock to your C1P. It requires very little additional hardware (a few pieces of wire) - if you have a 61e board. I don't - so here is a hardware real-time clock which runs independent from your micro and is easy to hook up to a port if you have one.

HARDWARE

The 60 Hz time base consists mainly of a MM 5369 Oscillator/Divider and a color burst crystal. A line frequency time base (from 60 Hz household current) could also be used.

The 60 Hz pulses are further divided by the next stages: 10 Hz, 1 Hz, Seconds (units 0-9), Seconds (Tens 0-5), Minutes (units 0-9), Minutes (tens 0-5), Sorry no Hour stage - requires 1 more 7492. Since ripple(asynchronous) counters are being used, where a pulse (high to low transition) at the input of the first stage of the chain "trickles" through the stages in domino fashion, precautions must be taken to get error-free (time) readouts. To allow the outputs to settle or to prevent a new pulse from disturbing the outputs, a latch (7475) will hold the output of the 1 Hz stage until a reading through port A has been taken. The HOLD line is held LOW during this time by a POKE to the PA7 line of port A (made an OUTput when initializing the VIA).

The 8 bit latch (74LS 373) is really wasted here because Latch Enable and Output Enable are always on. It is left over from an earlier version. It could be used for multi-plexing inputs since the outputs are TRIstate This is of course not a complete clock. Missing are: manual set buttons, hour display. The reason for all these ommisions: somebody came up with a better mousetrap! If you really need a time clock you are much better off using the clock chip (MSM 5832) diseribed in an article by Randy Sebra in MICRO (Feb'81) Less work and more features for less money (thanks to LSI).

THE PROGRAM

Line 10: make PA0-6 inputs, PA7 output

Line 20: make HOLD line high

Lines 200-230: HOLD line low, readout, HOLD high If there is a change from the previous reading sub routine 1000 will convert decimal value into HEX (a string) and display the time at the bottom of the screen (where it will not be scrolled). The printing of A and H$ is only for test purposes and can be omitted if you add this routine to one of your programs that require a real-time clock.

For that little extra touch the timing loop is split in half and makes a colon flash in front of the time display.

```
OK
LIST

 10 POKE40947,128
 20 POKE40945,128
 200 POKE40945,0
 210 A=PEEK(40945)
 220 POKE40945,128
 230 IFA<>T THEN PRINTA;T=A:GOS
UB1000
 300 FORX=1TO1000:NEXTX
 310 POKE54151,32
 320 FORX=1TO1000:NEXTX
 330 POKE54151,58
 340 GOTO200
 1000 HX$="0123456789ABCDEF"
 1110 B=A
 1120 H$="":FORI=1TO4:K=INT(B/16
):H$=MID$(HX$,B-K*16+1,1)+H$:B=K
 1130 NEXTI
 1135 PRINTH$;
 1140 POKE54152,ASC(MID$(H$,3,1
))
 1150 POKE54153,ASC(RIGHT$(H$,1)
)
 1160 RETURN
OK
```

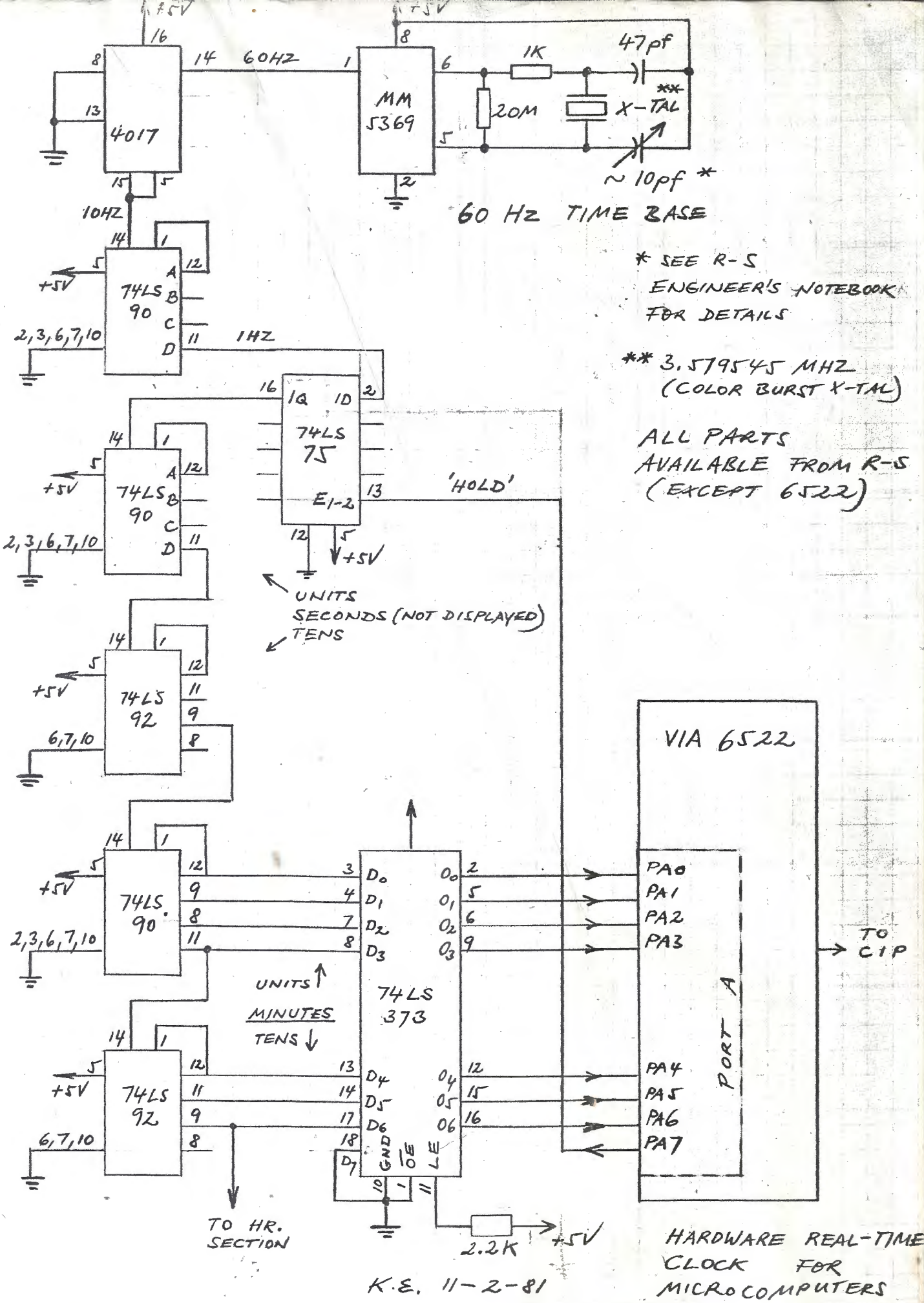
references:

Real Time Clock for 610 Board by Mike Cohen OSI-tems9/81

Does Anyone Really Know What Time It Is?
by Randy Sebra MICRO 2/81

Engineer!s Notebook by Forrest M. Mims,III Radio Shack

CALECTRO digital handbook


+5V
16
8
14 60HZ
1
4017
13
15
5
10HZ
MM 5369
6
1K
47pf
20M
X-TAL
**
5
~ 10pf *
2
60 Hz TIME BASE
* SEE R-S ENGINEER'S NOTEBOOK FOR DETAILS
** 3.579545 MHZ (COLOR BURST X-TAL)
ALL PARTS AVAILABLE FROM R-S (EXCEPT 6522)
14
1
74LS 90
A
B
C
D
12
11
2,3,6,7,10
1HZ
74LS 75
1Q
1D
2
E1-2
13
12
'HOLD'
UNITS SECONDS (NOT DISPLAYED)
TENS
74LS 92
9
8
6,7,10
UNITS
MINUTES
TENS
74LS 373
D0
D1
D2
D3
D4
D5
D6
D7
O0
O1
O2
O3
O4
O5
O6
GND
OE
LE
18
17
VIA 6522
PA0
PA1
PA2
PA3
PA4
PA5
PA6
PA7
PORT A
TO C1P
TO HR. SECTION
2.2K
K.E. 11-2-81
HARDWARE REAL-TIME CLOCK FOR MICROCOMPUTERS