Newton Software Exchange

INDEPENDENT
OSI USERS
NEWSLETTER

VOL. 1, No. 4 APRIL, 1979

MAILING ADDRESS: P.O. Box 518, Newton Corner, MA 02158

FIXING THE USR BUG IN THE C2-4P

Herb Purcell, 1634 Stanford Dr., Anchorage, AK 99504, has figured out why the USR call does not work properly. He writes, "I found a clue in the original Mits Microsoft manual. An FC error message will result from 'a call to USR before the address of the machine language subroutine has been patched in.' I was putting my vector in hex 23E and 23F as the OSI Basic manual says to do. So I suspected the vector address was wrong.

"The suspicion was confirmed by Bruce Thomson's program in (Feb. 1979 issue of) the newsletter. Note that on lines 8000 and 8010 he puts the vector in decimal 11 and 12. I tried it and it works!."

HELP NEEDED BY CPM USER

Joanne B. Rudnytsky of Quality Educational Designs, 2924 N.E. Stanton, Portland, OR 97212 is trying to use an OSI C3-S1, running under CPM, as a developmental system for mathematics education programs. She needs to know how to define the CA-10X (550 board) as an output device. Any suggestions.

STATISTICAL SOFTWARE WANTED

Martha McNutt, product evaluation department, Anchor Hocking, 109 North Broad St., Lancaster, Ohio 43130 needs statistical programs for a Challenger III. She specifically is looking for "routines for ANOVA's, different types of regression, transformation of variables, and other simple statistical calculations."

We suggested the Osborne book, "Some Common Basic Programs", have any of you other ideas or progams?

TOUCHING UP CABINETS AND MORE ON CHARATER SETS

Philip R. Moscinski, N2EU, P.O. Box 54, Harrison, N.J. 07029 offers a few suggestions on touching up scratched metal cabinets. He fixed his C2-4P using Testor's 'flat black' hobby paint, and the blue area by a mixture of two Testor's hobby paints: three parts No.#1111 Blue and one part No.#1108 Blue.

Philip is very interested in graphics for the 2-4P. He notes that OSI sent him literature, free on request, which contained a book on the character graphics which listed the whole graphics set of the 2-4P. Also included were complete grids for the graphic experimenter, giving both hex and decimal locations.

MYSTERY PROGRAMMER REVEALED

Richard A. Lary, P.O. Box 234 Kilauea, HI 96754, who supplied the listing of BASIC error codes on systems using the 540 video board, which is on page 4 of this issue, says, "Ever wonder who wrote OSI Basic in ROM? Type "A" in response to MEMORY SIZE? and you will find out." Richard invites a call if you are visiting KAUAI. His home phone is 828-1319.

--
This letter was so good and so complete, we are running it as is. Thanks, Ken.

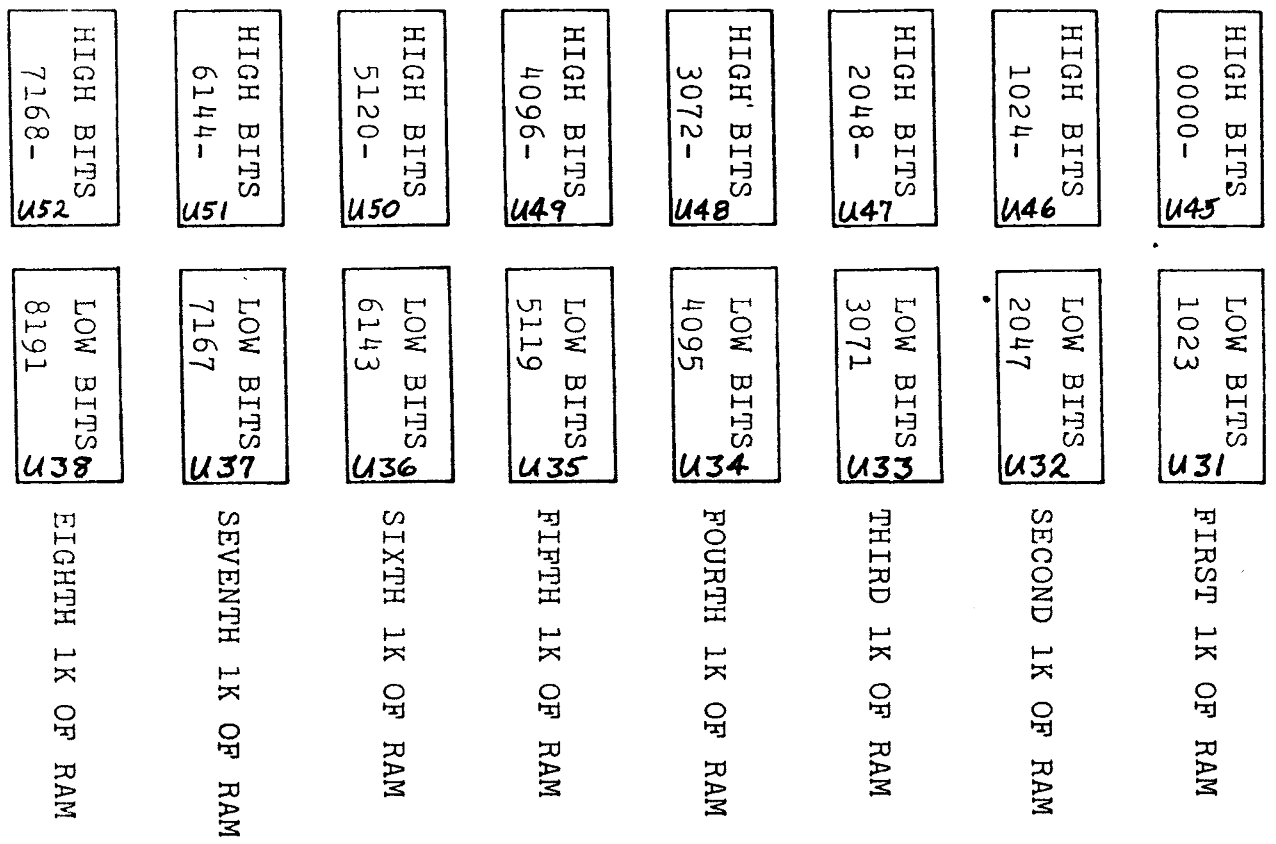
Kenneth F. Elliott
123 Raleigh St.
Rochester, New York
14620

Dear Editor,

Herein is a small bit of information for Superboard II and IP users. This was the result of a hardware problem I experienced a few weeks ago on my CPU 600 board. After six (expensive) calls to the factory, and no contact with the technician on any of the calls, I decided to look into the matter myself.

The problem was a flaky RAM chip (2114) in my 8K memory, but the question was, which one of the sixteen was doing it to me. On an intermittant basis, it was returning something other than what was put into it.

The first step was to convert the memory map of the RAMs from Hex to Decimal. It worked out as follows:



Each two chip pair is responsible for a total of 1000 Bytes of memory. One of the pair is the storage for the lower four Bits, and the other stores the higher four Bits of each eight bit word (or Byte).

I then wrote a simple program to "POKE" into each Byte a value that would then be "PEEKed" out again and checked against what was put into the location. The value "POKED" into each location was 255, which represents an eight Bit Binary word of 1111 1111 . Or really two four Bit words, each of which is stored in a separate chip of the pair.

The program follows:

```
10  FOR X = 1024 TO 8191
20  POKE X, 255 : Y = PEEK (X)
30  IF Y<>255 THEN PRINT X ; Y
40  NEXT
```

There are 58 Bytes of program.
Run time= 1.5 min.

If there is a Bit failure, the program will print out X (the decimal location of the failure), and Y (the value that is there currently).

The value that comes back as Y is then converted to its binary value, and can then be compared with the binary value of the inputed 255.

AN EXAMPLE:

Let's say you got a printout of 5021 191.

This indicates a failure in the fifth 1K of RAM. Note that the fifth K stores from 4096 to 5119 on those two chips.

Okay, which of these two are bad? To determine this, we have to find out whether the failure is in the lower four bits, or the higher four bits. A comparison of the binary values is made.

255 (dec) equals 1111 1111 (binary)

191 (dec) equals 1011 1111 (binary)

It is now obvious that a zero has been returned in place of the one that was put into that location on the 64th Bit spot. So then, it is the chip that stores the higher four bits that is to blame.

If you have trouble with decimal to binary conversion, look at your OSI IP User's Manual. They give a decent program for this conversion.

At this point, mark the suspected bad chip and swap it with one from another 1K section and run the program again to see if the problem follows the chip to the new location. If it does, then then chip is indeed bad. If it doesn't, then look to the possibility that the socket might be bad, or a bad solder connection near that chip is causing the failure.

WARNING: TURN OFF POWER WHEN EXCHANGING CHIPS !!!!!!

Be absolutely sure you mark the suspected chip and its location on the board. And when you swap chips, remember where you got the "good one" from the board.

Kenneth Elliott
Kenneth F. Elliott

Additional:

The program doesn't check below 1024, or the first 1K block. Basic in ROM uses about 0000 to 769 for system overhead. You don't need to POKE into this area, but there is no reason you can't start POKEing and PEEKing from 900 on.

Change line 10 to FOR X = 900 TO 8191

The process of elimination of the first 1K block was easy enough for me using the original program.

OUR THANKS
To Richard Lary, P.O. Box 234, Kilauea, HI 96754 for taking the time and trouble to produce the following list of error codes for C2-4P using the 540 video board.

BASIC ERROR CODES

CODE	DEFINITION
D/	Double Dimension: Variable dimensioned twice.
F/	Function Call Error: Parameter passed to function out of range.
I/	Illegal Direct: Input or DEF FN statements cannot be used in direct mode.
N\	NEXT without FOR
O/	Out of Data: More READs than DATA.
O┐	Out of Memory: Program too large or too many GOSUBs,FOR-NEXT loops or variables.
O▟	Overflow: Result of calculation too large for basic.
S┘	Syntax Error: Typo, ect.
R\	RETURN without GOSUB.
U▟	Undefined Statement: Attempt to jump to nonexistent line number.
/◢	Division by Zero
C┘	Continue Error: Attempt to inappropriately continue from BREAK or STOP.
L▟	Long String: String longer than 255 characters.
O▟	Out of String Space: Same as out of memory
S▟	String Temporaries: String expression too complex.
T┐	Type Mismatch: String variable mismatched to numeric variable.
U─	Undefined Function
B	Bad Subscript: Improper subscript following DIM statement.

TIRED OF RENUMBERING? TRY THIS---
Earl Morris, 3200 Washington, Midland MI 48640 writes that the BASIC line renumbering program in the March 1979 issue of Personal Computing will run on OSI if the following changes are made:

line #	change from	change to
63010	AH=4	AH=3
63510	L=1024	L=768
63540	<>137	<>136
	<>141	<>140
	<>167	<>160
