OSI-TEMS

Volume 2, No 6. July, 1980

THE CHALLENGER JOURNAL

2

OSI - TEMS

Editor:
Mike Bassman

July, 1980 Volume 2, Issue 6

OSI-tems is a non-profit publication devoted to the Ohio Scientific (OSI) Challenger series of computers. Distributed monthly by the Ohio Scientific user's group of New York.

Membership fees (includes subscription) are $10 yearly, $5 for students. Meetings are held monthly (first thursday) at 7:00 P.M. in the basement of
Aristocraft
314 5th Avenue
N.Y., N.Y. 10001

Back issues of OSI-tems are now available. Order them at a meeting, and they will be ready for the next. Charge is $.50 per back issue.

This month's cover has the new Challenger 1P series 2, available from Ohio Scientific in late August.

July 4th Special - Danny Schwartz

```
10 FOR K=1 TO 30:PRINT:NEXT
20 FOR Y=1 TO 18
30 IFY= 12 THEN RESTORE
40 READ I
50 FOR X=1 TO 23
60 PRINT CHR$(I);
70 NEXT X
80 IF Y<10 AND Y/2<>INT(Y/2)THENPRINT CHR$(13);"* * * * * *";
90 IF Y<10 AND Y/2= INT(Y/2)THENPRINT CHR$(13);" * * * * * * ";
100 PRINT:NEXT Y:PRINT:PRINT
110 DATA 161,151,144,161,135,154,164,32,159,162,32
```

Table of Contents

LETTERS

This month we are starting a new column. Warren Modell suggested it. ("I think we should have a 'Letters to the Editor' column. That sounds like a great idea Warren - so why don't you write the first letter. Who me? But it was my idea for someone else.) Warren was finally persuaded (with thumbscrews) to start off this column. In any case, if you have something to say, get out your crayons, scrawl it down, and we will be glad to print it.

Dear Editors,

"OSI-tems" is great! I've enjoyed each issue. You're all to be congratulated and thanked many times.

The best thanks and proof of our appreciation is for all of us to contribute something, to our "OSI-tems" publication. Actually, it's as easy as A,B,C.....here are some topics you may like to write about:

(a) Discuss meetings, ours or others you may have attended.
(b) Computer shows/Flea Markets.
(c) Book(s) Reviews.
(d) Magazine Articles Reviews and Results.
(e) Business Applications.
(f) Games you and your family, friends have enjoyed.
(g) Questions/Help!
(h) Computer News Items/General, not just OSI.
(i) Bargains you've gotten and were happy with.
(j) Bargains you've gotten and wished you hadn't.
(k) Group purchasing possibilities.
(l) Careers/Job Opportunities.
(m) Want ads.
(n) Sell ads.
(o) Ideas to be developed, business, games, education, hardware, etc.
(p) Possible group trips - related to computers or "Porno" type places.
(q) Computer courses.
(r) How to write a "good" program.
(s) How you use your computer to make your home more comfortable, efficient (other than games) i.e.,"Would you like to come up to my apartment to see my Computer? "ETCH!!"
(t) How you used your computer to steal successfully.....or how they mistreated you at police headquarters.

(u) You tell us about yourself and/or your computer system.

(v) How you explained successfully to your mate that you love her more than your computer.

(w) How you explained unsuccessfully to your ex-mate that you love her more than your computer.

(x) Computer communications.

(y). Computer graphics.

(z) How your pet zebra learned Basic on your computer which is now being repaired at Polk's... "Thanks for the pet zebra Mr. Polk."

(Ed note: Anyone caught mentioning a joke involving poke,Polk, or POKE is subject to capital punishment...)

See how easy it is! Twenty six fantastic ideas for you all to choose from. And if we each contribute at least one article every two months it would really help our friends that are doing all the work getting "OSI-tems" together for us!

I think "OSI-tems" should be distributed near the end of the meeting.

Ed: Why?

Perhaps "OSI-tems" should have Department Editors, such as:

(1) Hardware

(2) Software

(3) Resources

(4) General Computer News

(5) Book/Magazine Reviews.

Ed:If we had enough material to split up into departments, we would have department editors.

Cordially,

Warren Modell
3133 Rochambeau Avenue
Bronx, N.Y. 10467

6

Updates and Corrections

Danny Schwartz
2160 Matthews Avenue
Bronx, N.Y. 10462

Here are some enhancements for some programs of mine which appeared in previous issues of OSI-tems.

1) For the "fifteen puzzle" which appeared in the April issue - a "win" routine. Add these two lines to the program and it will stop when you have successfully solved the puzzle, reporting how many moves it took you to win. A win is considered to have occured when all 15 hex characters '1' through "f" are in their corresponding squares on the board; or, if this is impossible to obtain from your position, when the first 13 of them ('1' through 'd') are in the correct positions and the positions of 'E' and 'F' are reversed. The lines to add are:

```
215 M=M+1: FOR X=1 TO 13: IF C(X)=X THEN NEXT: IF C(16)=0 THEN 300
300 PRINT "YOU DID IT IN"M"MOVES."
```

You may be surprised at how many moves you have used! Incedentally, the "impossible" positions don't seem to arise too often. I can't really explain this, since mathematically 50% of all starting configurations should be "impossible". My experience has been though, that about 80% of the time you will get a starting position which is soluble.

2) For the dumb terminal program in the last issue - If there are any members who have a C4P or a C2-4P and have an RS-232C and use of a modem, the following three lines added to the program will make it run correctly on either a C1 or a C2 or C4.

```
5 C4=PEEK(65506)
25 IF C4 THEN POKE 756,252:POKE 762,252
75 IF C4 THEN POKE 977,21:POKE 978,191
```

If you are making a tape of the program, type in:

```
110 SAVE:PRINT
120 PRINT"CLEAR:POKE 121,1:POKE 122,4:POKE 1024,0:NEW":LIST-100
```

and then type RUN 110 to make the tape. Then the necessary POKEs will will be entered into the machine whenever the program is LOADed from the tape.

continued...

3) There were two typing errors in my "Beat the String Bug" article two issues back. The first is the calculation of how many bytes a string forming routine uses. It's best if you just ignore the "*10" which appears before the answer, 210. The second is in the program listing on the next page. Line 40 should read NEXT X. It should definately not read NEXT Y.

Number Formatter

Mike Cohen
2255 Barker Avenue
Bronx, N.Y. 10467

This program will input a number and then output it in any format you wish. You are asked the number of integer digits to be used, and the number of floating point digits to be used (F1 and F2, respectively). Then, you are asked if you wish to supress leading zeros, and finally asked for the actual number, which is outputted in your customized format.

```
 1 REM**NUMBER FORMATTER
 2 REM**MIKE COHEN
 10 INPUT"F1,F2";F1,F2
 20 INPUT"ZERO SUPPRESS";Z$:F0=(LEFT$(Z$,1)="Y")
 30 PRINT:PRINT
 40 INPUT X
 50 GOSUB1000
 60 PRINT X$
 70 GOTO 30
 1000 DEF FNL(X)=LEN(STR$(X))
 1003 X$=LEFT$(STR$(X),1):X=ABS(X)
 1004 X=X+.5*10↑(-F2)
 1005 L1=FNL(INT(X)):L2=FNL(X)-L1
 1010 X1$=MID$(LEFT$(STR$(X),L1),2)
 1020 X2$=RIGHT$(STR$(X),L2-1)
 1030 IFLEN(X1$)<F1THENX1$="0"+X1$:GOTO1030
 1040 IFLEN(X2$)<F2THENX2$=X2$+"0":GOTO1040
 1050 IFNOTF0THEN1090
 1060 M=2:X1$=" "+X1$
 1070 IFMID$(X1$,M,1)>"0"THEN1090
 1080 X1$=LEFT$(X1$,M-1)+" "+MID$(X1$,M+1):M=M+1:GOTO1070
 1090 X$=X$+RIGHT$(X1$,F1)+"."+LEFT$(X2$,F2)
 1095 RETURN
OK
```

Ugo Re
167 Sprucewood Drive
Levittown, N.Y. 11756

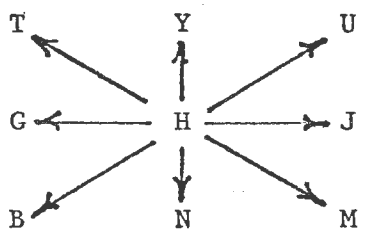
8

Etch-A-Sketch

PROGRAM FUNCTIONS

The program is located from Ø22Ø (hex) to Ø2F8 (hex) and uses zero page locations F6 to F9.

When the program is initialized, the screen is cleared and a ? is set at the lower center of the screen. Typing any ASCII character will replace the ? with that character. That character may then be moved in any direction on the screen by pressing any of the following keys: T, Y, U, G, J, B, N, M. Consider "H" as the present location of the character, the control vector keys will move the character as follows:



The following command keys are used:

Line Feed	clear screen and reinitialize
ESC	replace character at present location with a blank
Return	get a new character (?) at present location
Backspace	change size of screen, alternate between 64 x 32 and 32 x 32

If this program is to be called as a subroutine from another program, make the following changes:

Location	Current	New
Ø2C1	A9	EA
Ø2C2	2Ø	EA
Ø2C3	1Ø	EA
Ø2C4	86	6Ø

ESC will now cause a return from subroutine. To blank a character, hit return and then replace ? with a space.

The program resides from Ø22Ø (hex) to Ø2F8 (hex). The program may be repositioned by changing the 9 subroutine calls and 4 jump absolute at the following locations:

Ø235, Ø262, Ø28B, Ø292, Ø299, Q2AØ, Ø2A7, Ø2AE, Ø2B5, Ø2BC, Ø2D2, Ø2EB, Ø2F5

continued...

PROGRAM LISTING

	Ø	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Ø22Ø	D8	A9	Ø1	85	F6	EA	A9	ØØ	AA	85	F8	A9	DØ	85	F9	A9
Ø23Ø	2Ø	85	F7	A9	Ø1	2Ø	C5	Ø2	A9	D8	C5	F9	DØ	F5	C6	F9
Ø24Ø	A9	9F	85	F8	A9	3F	81	F8	2Ø	ED	FE	85	F7	81	F8	2Ø
Ø25Ø	ED	FE	C9	ØA	FØ	DØ	C9	1B	FØ	67	C9	ØD	FØ	E6	C9	5F
Ø26Ø	DØ	Ø3	4C	DE	Ø2	29	7F	C9	54	FØ	1E	C9	59	FØ	21	C9
Ø27Ø	55	FØ	24	C9	47	FØ	27	C9	4A	FØ	2A	C9	42	FØ	2D	C9
Ø28Ø	4E	FØ	3Ø	C9	4D	FØ	33	DØ	C6	A9	BF	2Ø	C5	Ø2	1Ø	BF
Ø29Ø	A9	CØ	2Ø	C5	Ø2	1Ø	B8	A9	C1	2Ø	C5	Ø2	1Ø	B1	A9	FF
Ø2AØ	2Ø	C5	Ø2	1Ø	AA	A9	Ø1	2Ø	C5	Ø2	1Ø	A3	A9	3F	2Ø	C5
Ø2BØ	Ø2	1Ø	9C	A9	4Ø	2Ø	C5	Ø2	1Ø	95	A9	41	2Ø	C5	Ø2	1Ø
Ø2CØ	8E	A9	2Ø	1Ø	86	18	48	65	F8	85	F8	68	3Ø	Ø7	9Ø	Ø2
Ø2DØ	E6	F9	4C	D9	Ø2	BØ	Ø2	C6	F9	A5	F7	81	F8	6Ø	A5	F6
Ø2EØ	C9	Ø1	FØ	ØA	E6	F6	A5	F6	8D	44	DE	4C	4F	Ø2	C6	F6
Ø2FØ	A5	F6	8D	44	DE	4C	4F	Ø2	Ø1	FF	FF	FF	FF	FF	FF	FF

PROGRAM OPERATION

Ø22Ø - Ø22E	register initialization
Ø22F - Ø23F	clear screen
Ø24Ø - Ø247	place ? on screen
Ø248 - Ø24E	get an ASCII character to replace ?
Ø24F - Ø288	get instructions (command or movement vectors)
Ø289 - Ø2CØ	computation to determine next screen location
Ø2C1 - Ø2C4	get new instructions
Ø2C5 - Ø2DD	screen location subroutine
Ø2DE - Ø2F8	screen size subroutine

To run on a C1P with it's smaller screen size, make the following changes:

Location	Current	New
Ø239	D8	D4
Ø241	9F	6F
Ø28A	BF	DF
Ø291	CØ	EØ
Ø298	C1	E1
Ø2AD	3F	1F
Ø2B4	4Ø	2Ø
Ø2BB	41	21

The program should now work on the C1P. Let me know if you have any problems.

10
PRINT AT

Salomon Lederman

40 Waterside Plaza
N.Y., N.Y. 10010

One feature OSI Basic does not have, and others do is the "PRINT AT" command. This means that you can print your message at any part of the screen. There have been slow Basic substitutes to do the same thing, but now there is a machine language solution. The program below will do just that. The format of the substitute print at statement is:

X=USR(X)(message)

When the program encounters this statement it will jump to the subroutine at $222, located in the free part of page 2. After the program has been entered, the initialization is: POKE 11,34:POKE 12,2. The program has a starting screen value of $D1C9, (ie: where the program puts the message at) but this can easily be changed to any location the user wishes. Following is the program.

LOCATION	MACHINE CODE	MNEUMONICS
222	A9 00	LDA #$00
224	85 CC	STA $CC
226	20 BC 00	JSR $00BC
229	C9 29	CMP #$29
22B	F0 09	BEQ $0236
22D	8D C9 D1	STA $D1C9
230	EE 2E 02	INC $022E
233	4C 26 02	JMP $0226
236	A9 EF	LDA #$EF
238	85 CC	STA $CC
23A	20 BC 00	JSR $00BC
23D	60	RTS

11

Two Software Reviews, or, The President Speaks!

David Gillett
President, Ohio Scientific User's Group of New York
340 West 19th Street
N.Y., N.Y. 10011

Escape from Mars - $14.95
Time Trek - $9.95

Now this is something like what computer game playing is all about! 'Escape from Mars' is an Adventure style program, and 'Time Trek is Rodger Olsen's version of Star Trek. "Oh no! Not another Star Drek! Aargh!" I hear you cry? Star Trek it may be, but drek it's not. It has a pretty standard game board and instruction set, but the twist is that it's played in real time, with the stardate ticking away whilst you think what to do next, instead of just counting your number of moves. The real eye-opener, however, is the display, which does not scroll! The current quadrant map is shown in a frame in the top left corner, with the long-range scan framed beneath it, and the instrumentation details (stardate, energy, torpedoes left, klingons left,etc.) in the next column on the right-hand side. When you move or fire in the current quadrant, you can see the action in the left-hand frame and you even see the start of the journey when you warp from one sector to another. I haven't had time to play a full game yet, but I'm certainly looking forward to it.

'Escape from Mars' is an adventure game where you find yourself on the planet without any fuel for your spacecraft. The object, of course, is to re-fuel and take off. There are more than fifteen areas or rooms to traverse, and plenty of bits and pieces to pick up,hit, push, drop, or even ignore (there is at least one item that is a hindrace rather than a help) and they all have to be in the right place at the right time for you to be able to move on to the next step. It took me about two weeks of playing almost every night before I was able to take off, and I've still yet to find how to do it in the least number of moves. There are a couple of bugs in the program that show up as unexpected responses to a command but nothing that aborts the program. On second thoughts, I think the string bug got me one time, though I think that was my fault - there is a small program which is supposed to clear it which you run first and then warm start, which I had run and not warm started. You can even cheat slightly if you fall off something and die, or hit (return) without a command, by typing 'GOTO 500' which restarts the game at the point where you had the accident. If you run out of air, however, you are most

continued...

definitely dead, and have to start all over again.

I can recommend both these games, both for their high entertainment rating, and for all the neat programming tricks that must be in there if you want to hunt them out.

GOTO X

Danny Schwartz
2160 Matthews Avenue
Bronx, N.Y. 10462

Finally, a short little routine in machine language which may come in handy any number of times. It gives you the equivalent of the "GOTO X" command present in ATARI Basic and some others, where the "X" can be any numeric expression. Of course, you can't really change the GOTO statement in the ROM, but you can create a function that will act like a "GOTO".

The syntax is X=USR(NE), when NE is a numeric expression consisting of variables, arithmetic expressions, other functions, or whatever. When the statement is executed, the program will jump to the line number represented by NE and continue from there, providing there is such a line number in the program. If there isn't, you will get a "US ERROR", just like a real GOTO. (Yes, the "S" in the "US ERROR" will be that funny graphic character, but we can't print that in the magazine!)

The routine to accomplish this is just six bytes - two instructions. The instructions are:

```
JSR $B408
JMP $A6BC
```

In machine language this would be:

```
20 08 B4
4C BC A6
```

As a DATA statement in a Basic program this is

```
DATA 32,8,180,76,188,166
```

Point 11 and 12 at this routine and you're in business!

13

Some Thoughts on Cassette Recording

Terry Terrance
Orion Software Associates, Inc.
147 Main Street
Ossining, N.Y. 10562

Recently, I have learned some interesting things about cassette recording, and it has prompted me to write this cursory (and very incomplete) review of the subject. Let me preface this discussion with the disclaimer that most of my findings are strictly empirical, and the short section on theory that follows was gleaned from the various references that can be found at the end of this article.

Some Theory

Actually, it is a rather neat trick that digital data can be recorded on audio tape with an audio tape recorder at all. To accomplish this, both the recorder and the computer must be tricked into thinking that they are seeing more or less exactly what they want to see.

Today's cassette recorders are typical mass produced devices, as evidenced by their tolerances. Speed variations in the typical, mid-priced (i.e. $50) recorder are plus or minus 10 percent; harmonic distortion can approach a similar figure, and they suffer from limited fidelity and bandwidth inherent in the single chip designs that are common. And, of course, they are designed for audio applications.

By comparison, your computer is a precision instrument. It outputs, and expects on input, precisely timed trains of square waves. Drop a single bit, and you wind up with a garbled character. Your computer doesn't understand sine waves, tones, or any other thing your cassette likes.

The actual tape that you use is the final item in the mix. Audio tape doesn't record at the energy density of true digital tape. Audio tape also allows minor dropouts (areas of little or no magnetic medium). In the longer length cassettes (C90 and C120), where the tape is thin, you must contend with tape stretch.

Several things must be recorded on the tape besides the actual string of ones and zeros. Information on timing and byte boundaries must be included. Various tape formats accomplish this in different ways. All Ohio Scientific computers use the Kansas City Standard. In the Kansas City Standard, ones are recorded as eight cycles of a 2400 hz tone, and ones are recorded as four cycles of a 1200 hz tone. Why, you might ask, all this business with cycles?

continued...

Aren't the two tones enough to discriminate between zeros and ones? Well, the Kansas City Standard circuitry could discriminate on the basis of the tones alone if your recorder had near perfect speed regulation. But to accomodate speed variations in recorders, zeros and ones contain different numbers of cycles so there can be no mistake. This form of recording is called 'redundant'. The result is high reliability, given the simple Kansas City Standard hardware, and the low to moderate priced recorders it is used with.

The price redundancy extracts, as all of us know, is speed. (Ed note: the Kansas City Standard runs at 300 baud.) The Kansas City Standard can be upgraded to run at 600, 1200, or 2400 baud by reducing the number of cycles in each bit's representation. However, hardware costs go up and reliability goes down, so a high quality tape recorder is a must. Considering the reliability and ease of use of the Kansas City Standard, the price is a small one to pay. For a better comparison with other tape formats see reference 1. For a discussion of the physics of recording, see reference 3. For a discussion of true digital recording, see reference 2.

While we can do little to make our OSI computers run at higher speeds, without major hardware mods, we can do things to increase the cassette interface's reliability so that long 8K programs will record and reload with no errors.

Empirical Data

These are strictly my observations on using, or trying to use, several different cassette machines with a Challenger 1P and a Challenger 4P. I advise you to take everything I say with a grain of salt, especially before you chuck your tape recorder for another model.

I do not have the analytical instruments, time, money, or patience to sit down and survey a large number of cassette machines and find out exactly why some work well and some don't. However, I have come to some unscientific conclusions about what one should look for in a cassette recorder.

First of all, you get what you pay for. More expensive recorders have better waveform fidelity, better speed regulation, wider bandwidth, and frequency range. All of these factors will contribute to reliable operation. However, some people have found it difficult or impossible to record on stereo cassette decks which should rank highly in all these categories. In any case, forget the cheapies. Don't just use any cassette you happen to have lying around. Since cassettes are your primary storage medium at this point, look on your cassette recorder as an important part of your system.

continued...

It is important not to use an older machine. Older machines, even those which sold for a considerable sum several years ago, just don't have the advances in electronics and head technologies available on today's recorders. Both an expensive portable and a good cassette deck that I have, each over ten years old, absolutely will not work with a Challenger 1P.

What features should you look for? A tone control seems to be a must for reliable recording. Using a tone control is the only means you have of matching the tape to the recorder for the best playback fidelity. Everyone likes to have a tape counter, and they are found on most machines that have tone control. Remember, though, tape counter calibrations are arbitrary, and a program recorded at a certain spot on one recorder might have a totally different reading on another. If you find a portable with a good vu meter it might be worth the extra you will pay to buy that machine. A meter is both a convenience and a tool. A vu meter will let you set good recording and playback levels. It will also let you see, visibly, where programs begin and end on the tape. It saves your ears and your time to load unwanted programs into the computer.

A word about AGC and tape head alignment. AGC (Automatic Gain Control) is found on most portables. It is a circuit that attempts to prevent recording at levels which are too high. It can be both a blessing and a curse for our purposes. It can help prevent gross oversaturation of the tape and the resultant waveform distortion, which can degrade performance. But, if you have a machine with a relatively weak signal like a C4, it can prevent recording at the near oversaturated levels which seem to be desirable for good operation. Tape head alignment (in azimuth) is probably the single most common cause of being unable to read tapes recorded on another machine. Misalignment of the tape heads causes a loss in amplitude which can be tolerated on a audio recording, but on a digital recording, it can be trouble. If either your recorder's head or the head of a recorder that records a tape is out of line, it can possibly render the tape unreadable. It does not affect tapes recorded and played back on the same machine, only when two different recorders are involved. The only way to assure that your recorder's heads are in line when you buy one, is to buy a better quality recorder. Even so, any recorder's heads can get out of line with use or rough handling; that's why your recorder should be dedicated to your system. For more on cassette head alignment, see reference 4.

continued...

Care and Feeding of your Recorder

Tape heads can get out of line, and can be re-aligned, but one easy maintenance item is head cleanliness. Most of us periodically clean our cassette heads, and if you don't, you should. There are liquids, sprays, and special tapes that will all do the job. I find the tapes to be less messy and easier, although slower to use.

How many of you have ever thought of occasionally (say twice a year) demagnetizing your tape heads? If you don't, you should consider it. Tape heads can build up residual magnetization that can actually partially erase your precious program tapes as they pass through your machine. Various gadgets are available to do the job. Just be sure that your pre-recorded tapes are nowhere near when you perform the demagnetizing operation. Take your tape machine elsewhere when you demagnetize.

Tape is another exceedingly important consideration. If you want reliability, use good quality tape. Good tape will have few or no dropouts, will feed without sticking, and will generally outperform cheap tape. The problem is that good tape is usually not available in short lengths. Memorex can occasionally be found in C30 and C45 lengths. An expensive alternative may be to use Verbatim or other certified cassettes designed for word processors. These can be found in C10 and C30 lengths, but are expensive. You can identify these by a notch cut into the top of the cassette shell. Almost every review of an OSI cassette based computer I have ever seen comments on the reliability of the cassette port with cheap recorders and tape, don't risk your 8K machine language program on it... One tip for using any tape which has not been used for awhile, or a new tape, is to wind it back and forth a few times using the fast forward and rewind controls to free up any sticky spots. This is particularly important prior to recording.

The Stereo Connection

Many people have complained that they have had difficulty using a stereo deck to record data. I have had just the opposite experience. Using a stereo deck was a dream.

I bought a Sanyo RD 5035 to put some programs on chrome tape prior to being mastered for duplication. It recorded reliably and the vu meters were a big help. To successfully use a deck, one must follow certain rules. No matter what you read in the various magazines, an OSI computer only develops

continued...

enough voltage to drive the MIC inputs. It will not drive the line inputs. The deck you use must have some arrangement for mono recording. On my deck this function is provided. On a machine without such a feature, a 'Y' cable connected to both inputs might serve the same function.

Set the input level controls of the deck to read approximately 0 to 1 DB when your computer is outputting signal after the SAVE command and before the LIST command. This will produce tapes that are very near saturation. If you are going to use the deck to read tapes back into the machine, it must, unlike my Sanyo, have output level controls. They must be adjusted to give you reliable loads. You might pick the output off of the headphone jack rather than the line inputs.

Chrome Tape

The people at Cook Labs seem to feel that programs recorded on chrome tape and then played back on either a chrome tape compatible or ferric (normal) tape compatible machine is more reliable. (see reference 4). I've found this to be true and adds another advantage to using a deck; the ability to record on chrome tape and play back at ferric equilization for greater reliability.

Some Gadgets

I've dug through my magazine collection and come up with three articles that describe building gadgets which will help the non vu-meter equipped recorder set proper levels (see references 4,5,6). One of these is available from Cook Labs for $49.50 assembled, and with an azimuth test cassette. These projects are not like the data dubber in that they don't condition the signal in any way. I doubt if the data dubber, designed for the TRS 80 (which uses a tone/no tone system), would work with the OSI Kansas City Standard.

I hope that the foregoing will help some of you through any tape problems you may have.

References

(1) Audio Cassette Recording Formats, M. Chamberlin, Computer Bits Popular Electronics, 7/79, p. 80

(2) Digital Magnetic Recording, M. Chamberlin, Computer Bits Popular Electronics, 9/79, p. 98

(3) Magnetic Recording for Computers, W. Manly, Scelbi Byte Primer, p. 2-60

(4) Peek Reading Meter for Data Cassettes, Emory Cook, Popular Electronics, 2/80, p. 84

continued...

18

(5) Cassette Tape Monitor, Salerno, Creative Computing 9/79, p. 112

(6) Hear and See It, S. Gibson, Kilobaud 3/78, p. 40

I-Ching

Mike Cohen
2255 Barker Avenue
Bronx, N.Y. 10467

Computerized fortunetelling is here... This program will draw hexagrams, give you it's number, then all you have to do is look up it's interpretation to find out the meaning.

```
1 REM **      I-CHING
2 REM ** BY MIKE COHEN
3 REM **
10 INPUT"RANDOM NUMBER";Z
20 POKE 530,1
30 T=53675:P=57088
40 PRINT:PRINT:PRINT:PRINT"HIT RETURN TO CONTINUE";
50 Z=RND(Z):POKEP,223:IFPEEK(P)<>247THEN50
60 PRINT:FORM=53248TO54272:POKEM,32:NEXTM
70 H=0:FORI=1TO6:M=INT(2*RND(Z))
80 H=H+INT(M*(2↑(6-I))+.4):K=T+32*I
90 FORJ=1TO9:POKEK+J,148:NEXTJ
100 IFMTHENPOKEK+5,32
110 NEXTI:PRINTTAB(10);"#";54-H
120 PRINT:PRINT:PRINT"WANT ANOTHER ?";
130 POKEP,239:IFPEEK(P)=247THENPRINT"Y":PRINT:GOTO40
140 POKEP,251:IFPEEK(P)<>247THEN130
150 PRINT"N":PRINT
160 POKE 530,0
190 END
```

19

Quiz System

Mike Bassman
39-65 52 street
Woodside, N.Y. 11377

As of yet, there has been no educational software for the Challenger 1P. Consequently, most people have been forced to write their own. Here is a simple quizzing system which will allow you to enter questions on varied topics. Best of all, the response is timed, so that if an answer is not entered within a specified period of time, the program can continue. In this program, as an example, I have used national capitals for questions. Questions are stored in DATA statements in 4 parts. The first two parts are the question. The next is the time given for that question, in units, with each unit being .015 second. The last part is the answer. Following is an example:

1000 DATA What's the capital,of Spain,200,MADRID

part 1. 2. 3. 4.

```
10 FORX=64768TO64974:I=PEEK(X):POKEX-64222,I:NEXTX:POKE662,66
30 POKE11,34:POKE12,2
40 FORK=1TO35:PRINT:NEXT
50 NQ=26
100 FORK=1TOINT(RND(2)*NQ+1):READA$,A$,A,A$:NEXT
110 READA1$,A2$,T,A3$
120 GOSUB200
200 PRINTA1$:PRINTA2$:I$=""
210 FORX=1TOT
220 X=USR(X):X$=CHR$(PEEK(531))
230 IFASC(X$)=2THEN250
240 PRINTX$;:I$=I$+X$:IFI$=A3$THEN300
250 NEXTX
260 PRINT:PRINT:PRINT
270 PRINT"Too late, john"
```

CONTINUED...

```
280 PRINT"The answer was",A3$
290 GOTO350
300 PRINT:PRINT:PRINT
310 PRINT"You win, john"
350 PRINT:PRINT:PRINT
360 PRINT"Hit space to try again":PRINT"or return to end"
370 X=USR(X):IFPEEK(531)=2THEN370
380 IFPEEK(531)=32THENRUN40
390 IFPEEK(531)<>13THEN370
400 FORK=1TO35:PRINT:NEXT:END
999 DATAX,X,1,X
1000 DATAWhat's the capital,"of Spain, john ?",200,MADRID
1010 DATAWhat's the capital,"of France, john ?",150,PARIS
1020 DATAWhat's the capital,"of Italy, john ?",200,ROME
1030 DATAWhat's the capital,"of England, john ?",250,LONDON
1040 DATAWhat's the capital,"of Ireland, john ?",275,DUBLIN
1050 DATAWhat's the capital,"of Russia, john ?",250,MOSCOW
1060 DATAWhat's the capital,"of Israel, john ?",325,JERUSALEM
1070 DATAWhat's the capital,"of Egypt, john ?",250,CAIRO
1080 DATAWhat's the capital,"of Iran, john ?",250,TEHRAN
1090 DATAWhat's the capital,"of West Germany, john ?",250,BONN
1100 DATAWhat's the capital,"of Portugal, john ?",400,LISBON
1110 DATAWhat's the capital,"of East Germany, john ?",250,BERLIN
1120 DATAWhat's the capital,"of Canada, john ?",350,MONTREAL
1130 DATAWhat's the capital,"of Colombia, john ?",350,BOGOTA
1140 DATAWhat's the capital,"of Venezuela, john ?",400,CARACAS
1150 DATAWhat's the capital,"of Poland, john ?",300,WARSAW
1160 DATAWhat's the capital,"of Hungary, john ?",400,BUDAPEST
1170 DATAWhat's the capital,"of Switzerland, john ?",300,BERN
1180 DATAWhat's the capital,"of Austria, john ?",300,VIENNA
1190 DATAWhat's the capital,"of Iraq, john ?",400,BAGHDAD
1200 DATAWhat's the capital,"of Afghanistan, john ?",275,KABUL
1210 DATAWhat's the capital,"of China, john ?",250,PEKING
1220 DATAWhat's the capital,"of Japan, john ?",250,TOKYO
1230 DATAWhat's the capital,"of Syria, john ?",400,DAMASCUS
1240 DATAWhat's the capital,"of Rhodesia, john ?",500,SALISBURY
1250 DATAWhat's the capital,"of Libya, john ?",250,TRIPOLI
OK
```

21

Memory Swapping

Salomon Lederman
40 Waterside Plaza
N.Y., N.Y. 10010

If you write machine language routines through Basic, as I do, then you've probably noticed that there are many special page 0 instructions for the user. This is very nice, because instructions for page 0 save program memory and generally run faster than other programs. Also there are some page 0 instructions that are not easy to duplicate in higher memory.

The problem with all this is that Basic also uses most of page 0 for its vital routines and pointers. So, how can we share page 0 with Basic? Well, it's not too difficult if you know about memory swapping. Here's the theory behind it. Let's say that you have a routine that you'd like to use on page 0. What you do is to put it into higher memory for storage for a while. Now this routine is written so that it will run only on page 0. When you need to run this routine, all you have to do is swap it with whatever is on page 0. And when you want to return to Basic, you just swap again, and page 0 is not changed at all.

The swapping routine is fairly simple in nature. Here's one that will swap page 0 with some other page in memory: it starts at 1800_{hex}, make sure that you do not swap a page that the routine is on to do the swapping:

LOCATION	MACHINE CODE	MNEUMONICS
1800	A2 00	LDX #00
1802	BD 00 00	LDA $0000,X
1805	BC 00 15	LDY $1500,X
1808	9D 00 15	STA $1500,X
180B	98	TYA
180C	9D 00 00	STA 0000,X
180F	E8	INX
1810	D0 F0	BNE $1802

Locations $1807 and $180A contain 15 to swap page $15. You can change those to any page you want. The routine itself is totally relocatable. If you run the routine twice, then everything will be back where it started.

22

BAR GRAPH

Mike Cohen
2255 Barker Avenue
Bronx, N.Y. 10467

```
1 REM ** BAR GRAPH DEMO
2 REM ** BY MIKE COHEN
3 REM ** BASED ON PROGRAM FOR PET IN COMPUTE #3
4 REM ** GENERAL PURPOSE SUBROUTINES:
5 REM **   1000 PLOT SCALED DATA
6 REM **   3000 SCALE DATA FOR BAR GRAPH
7 REM ** MAIN ROUTINE IS FOR DEMO ONLY
8 REM ** DATA CAN BE GENERATED BY ANY METHOD
9 REM **
10 GOSUB 2000
15 TC=53092:SF=20
20 DIM Y(12)
30 PRINT"ENTER DATA:":PRINT
40 FOR J=1 TO 12
50 PRINT J; TAB(8);:INPUT Y(J)
60 NEXT J
70 GOSUB 3000
80 GOSUB 2050
90 GOSUB 1000
100 GOTO100
1000 FORI=0TO24:POKETC+992+I,150:NEXTI
1010 FOR I=1 TO 12
1020 X=INT(I*2-1)
1030 FOR J=1 TO Y(I)
1040 POKE TC+((32-J)*32+X),161
1050 NEXT J,I
1060 RETURN
1999 REM CLEAR SCREEN
2000 M1=65316:M2=546:POKEM2,160:POKEM2+1,0:M2=M2+2
2010 POKE 11,34:POKE 12,2
2020 FORX=0TO16:Y=PEEK(M1+X):POKEM2+X,Y:NEXT:POKEM2+17,95
2050 X=USR(X):RETURN
3000 REM SCALE DATA
3010 L=1E6:G=-1E6
3020 FOR I=1 TO 12
3030 IF Y(I)<L THEN L=Y(I)
3040 IF Y(I)>G THEN G=Y(I)
3050 NEXT I
3060 FOR I=1 TO 12
3070 Y(I)=Y(I)-L
3080 Y(I)=INT(Y(I)/G*SF+1)
3090 NEXT I
3100 RETURN
OK
```

23

Warren Modell
3133 Rochambeau Avenue
Bronx, N.Y. 10467

REVIEW

"THE FIRST BOOK OF OHIO SCIENTIFIC" Volume 1
by J. Clothier and W. Adams $7.95, Elcomp Publishing, Inc.

I'm reviewing this book from a novice's point of view as I only have had my C1P for about one month and don't program as yet.

The author's opening statement is as follows, I quote:

"Forward"

"These volumes of 'The First Book of Ohio Scientific' are our initial effort at getting Ohio Scientific information to their many deserving customers. Ohio Sientific is sincerely dedicated (in our opinion) to providing good computer hardware value. The equiptment is fairly priced and well designed for ease of expansion. However, Ohio Scientific has lacked the sophistication required for writing and distributing a good set of Technical Manuals. At Elcomp, we felt that we could no longer wait for the appearance of good documentation.

We have been astounded at the large number of fellow C1,C2, and C3 users who have also been unable to obtain copies of badly needed listings, schematics, software definitions, and operating instructions. 'Let's get the Ohio Scientific bandwagon rolling,' we say. And we ask others to help by sending contributions and suggestions to this first edition. At the earliest possible date, Elcomp will finaliza the manuals, incorporate new information, and add additional volumes, as necessary."

John Clothier
W. Adams
March, 1980

End Quote.
Amen.

This book is difficult for a novice to understand, but you do get the "feel" that it's really going to help in the very near future. I bought the book on the recommendation by a game programmer named Ken Madel, who told me that it took him a long time to learn all these tricks and this book has it all - or most of it. (Ed. note: I am a fairly accomplished game programmer myself, and I didn't find one useful thing in here - MB)

continued...

Since it is sometimes difficult to get books of this type I decided to get it now, while it was available, instead of waiting till I arrived "There".

Page 10 through 18 are almost like paid advertising for Ohio Scientific line, but it did describe the entire line completely and simply. In May and June they're supposed to be publishing Vol II and Vol III. (Ed note - volume 2 is now available.) If anyone else has read the book let us know your opinion. If it is as good as it appears perhaps Polk's can order for the users group at a special rate if enough people are interested.

Random Numbers in Machine Language

Salomon Lederman
40 Waterside Plaza
N.Y., N.Y. 10010

While going through a copy of The First Book of KIM I ran into an interesting article by Jim Butterfield on random numbers in machine language. He suggests a way that can yield random numbers. It's a fairly simple algorithm. Let's say that you've got a list of five numbers, any five, in the range 0 through 255. Then these five 'seeds' can be used to generate s sequence of random numbers in the following way:

Let's say that your five numbers are A, B, C, D, and E. To get the next random number you evaluate A+B+E+1, and reduce mod 256. This is your next term. This number is then stored as your new E, but first B, C, D, and E are pushed over to the left, and A is lost. You go through this routine before you need a new number.

The numbers are not greatly random, as I found out from some testing through Basic, but for most purposes it works well. Here's the Butterfield routine.

```
D8 38 A5 13 65 16 65 17 85 12 A2 04 B5 12 95 13 CA 10 F9
```

In this routine locations 12-17 are used to store the numbers. Remember this is machine code and the numbers are in hexadecimal. In the OSI this would have to be changed because Basic is on page 0. Location 12 will store the new random number.

SERIAL INTERFACE

Ugo Re
167 Sprucewood Drive
Levittown, N.Y. 11756

Are you having troubles with your printer? Does it loose some characters at the start of a new line? The trouble may not be the printer but may be due to the computer not providing sufficient delay after a carrage return.

I had this problem between a C2-4P and the printer and was able to correct the problem by using the control lines that are built into the 6850 ACIA.

The 6850 is a 24 pin DIP that interfaces the CPU to outside devices by converting parallel I/O to serial I/O. Either a byte of parallel data or a Control Code is transmitted by the CPU to the 6850, while data or Status is received by the CPU. In addition to the data bus and the transmit and receive clock and data leads, the 6850 has various other control leads. There are the modem control leads $\overline{CTS}$, $\overline{RTS}$, and $\overline{DCD}$, the RS (Register Select) lead which determines which of two addressable locations will be accessed and the R/W lead that determines whether a Read or Write operation is in progress. (The two addressable locations are the Control/Status address (FC00 hex) and the Data address (FC01 hex)).

The Control Register is used to control the operation of the 6850. A code, which establishes the parameters for the 6850's operation (see Table 2), is written in to the control register by the CPU. When the computer is reset the CPU first loads the register with 03 hex (Reset), and then B1 hex ($\div$16 clock rate, 8 bits, no parity, 2 stop bits). If this operation does not meet your needs reset the Control Register and load in a new code (see Table2).

The Status Register uses status flags to monitor the serial data transfer logic (see Table 3). The Status Register is read in to the CPU's accumulator, then bit '0' or '1' is shifted right into the carry register. The carry register is then checked for the status of the Receive Data Register or the Transmit Data Register. If the register is set the CPU will proceed to read or write data to the 6850 (see Table 3).

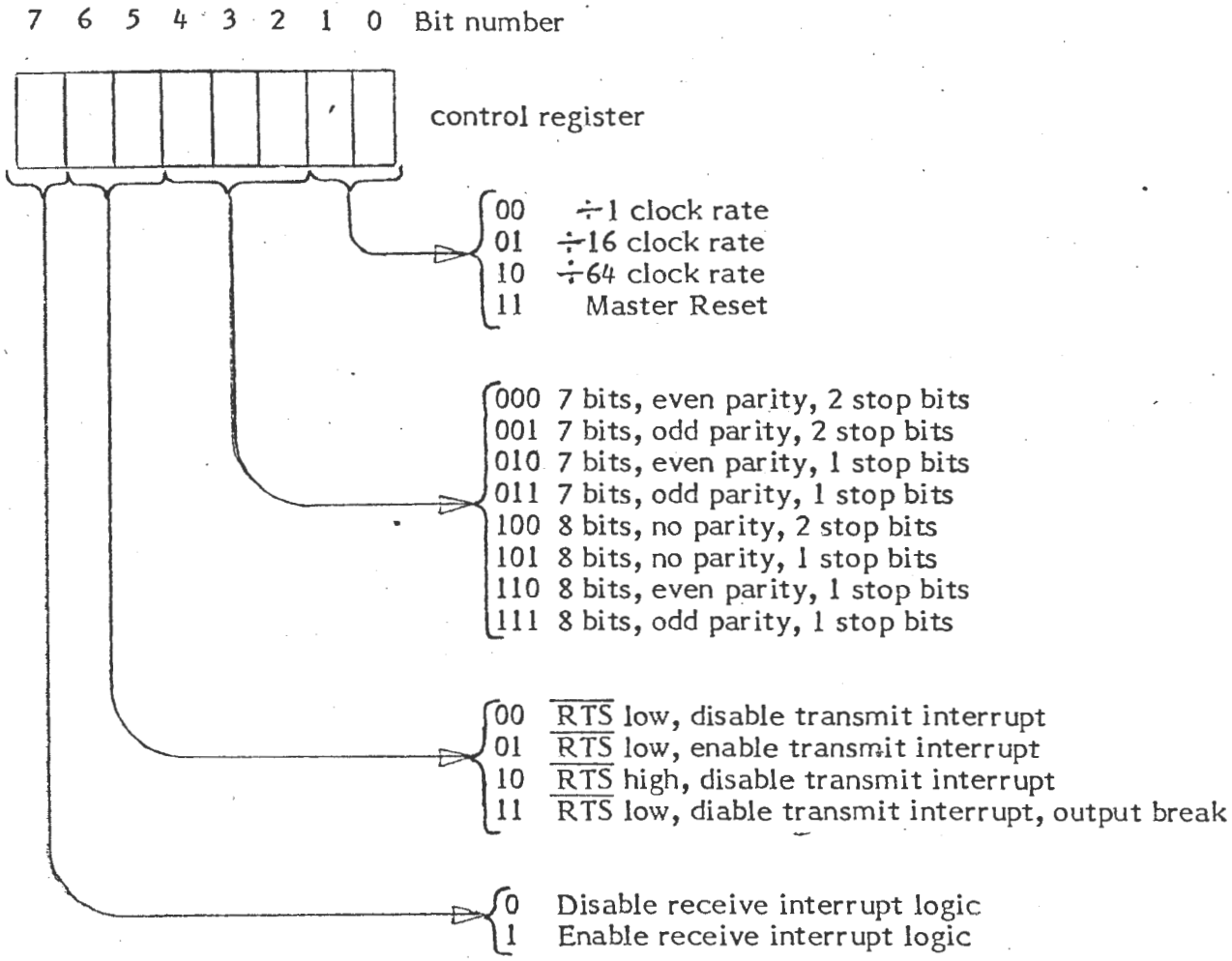
The other bits in the Status register are used by the 6850 to determine the status of external devices, bits 2 and 3, to detect various receiving errors, bits 4, 5 and 6, and to determine the source of an unacknowledged interrupt request, bit 7.

The computer does not check the status of bits 2 through 7 and in most cases we do not need to check them for status during normal operation.

continued...

26

Control Register



Program to change the control register

A9; 03	LDA	Master Reset Code
8D; 00; FC	STA	Write code to register
A9; XX*	LDA	new parameters
8D; 00; FC	STA	write new parameters to control register

*NOTE: Select new parameters from list above. Byte is then converted to hexadecimal notation and entered in program.

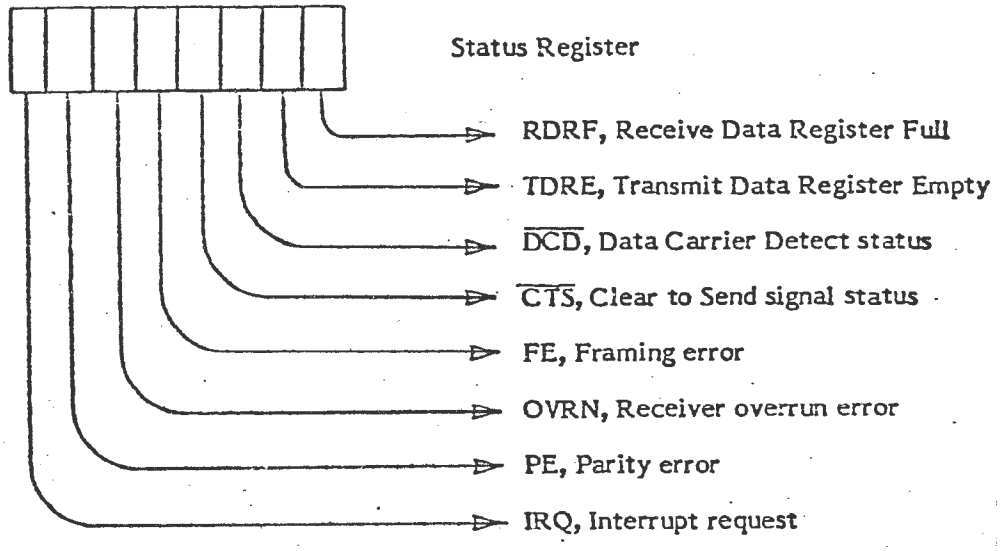
EXAMPLE: ÷16 clock rate; 7 bits, even parity, 2 stop bits; $\overline{RTS}$ low, enable transmit interrupt; Enable receive interrupt logic
10100001 binary = A1 hex

Table 2

continued...

Status Register

7 6 5 4 3 2 1 0 Bit number



. Table 3

A program to check the Status register prior to transmitting or receiving data, is listed below.

Serial Data Operation

A. Receive data from the Serial port.

AD, OO, FC	LDA	load status register
4A	LSR	shift bit '0' to carry
90, FA	BCC	not ready, check again
AD, 01, FC	LDA	load receive data
9D, XX, XX	STA	store data in memory

This program would continue until all data is received.

This program should check for end of data character, and the index register must be incremented to access the next memory address.

B. Transmit data from memory to Serial port.

AD, OO, FC	LDA	load status register
4A	LSR	
4A	LSR	shift bit '1' to carry
90, F9	BCC	not ready, check again
BD, XX, XX	LDA	load Acc from memory
8D, 01, FC	STA	store data in AC1A

The program would continue until all data has been transmitted

The program must increment the index register to access the next memory address. It must also check for the last data address.

continued...

The 6850 uses a low $\overline{CTS}$ to set the Status Register and report a TDRE (Transmit Data Register Empty) condition. The computer determines when to transmit a byte of data to the 6850 by testing the register for the TDRE condition. A high $\overline{CTS}$ signal will prevent the register from reporting a TDRE condition.

The $\overline{CTS}$ is normally strapped to ground, however the lead can be used as an interrupt signal for loss of transmission facilities, low paper alarm, printer or cassette not on line, etc., to prevent the useless transmission of data to an open line.

To modify the $\overline{CTS}$ lead remove the strap between pin 24 and ground and install a 1K resistor from pin 24 to +5v. and a wire from pin 24 to pin 5 of the EIA connector. The $\overline{CTS}$ lead, which will be high, can now be used to determine the status of the receiving device.

The printer must ground the $\overline{CTS}$ lead when it is ready to receive a character and let the lead go high when it is preforming a carrage return or line feed.

NOTE: do not make this modification unless you can control the $\overline{CTS}$ lead as the computer will go into a loop on 'SAVE' where it will remain until $\overline{CTS}$ is brought low.

The 6850 also has an $\overline{RTS}$ (Request To Send) control lead which is used to inform a data set that it is ready to transmit data. The data set will return the $\overline{RTS}$ signal as a $\overline{CTS}$ signal when it establishes a data link. The 6850 outputs a low $\overline{RTS}$ therefore $\overline{CTS}$ will be low and the computer will proceed to output the data

---

Both the serial and cassette interface use the 6850 ACIA to convert serial to parallel and parallel to serial. The following two tests can be used to test the serial and cassette ports.

NOTE: The first program tests that the byte received is the same as the one transmitted, while the second program records the characters on a tape which is then played back and displayed on the screen. There is no comparison made between transmit and receive data except what is displayed on the screen.

Test set up:

Test One - a small amplifier is connected to the cassette interface input/output jacks. Adjust the volume and tone controls and run the program.

continued...

I was able to use my cassette recorder (Radio Shack CTR-39) connected normally. A blank cassette, (no tape) is placed in the recorder and the Record/Play buttons are operated. The input signal is amplified and sent to the EAR jack. During record the volume control is inoperative however the amplified signal is more than adequate to drive the cassette interface.

Test Two - the cassette recorder is connected normally. The signal from the cassette interface is recorded, then played back. An ACSII character will be displayed on the screen 256 times, then the next character will be displayed. This will continue until all 256 characters have been displayed.

Test one can also be used with a data set that has a loop-back test. This test will verify the operation of the computer, the serial interface, and the data set.

If test one is OK but test two fails then the problem is in the cassette play-back. Clead the play/record head, the rollers, and the capstand in the cassette. If it still fails then try a new cassette tape.

Test two can also be used to test a tape for data drop-outs (every thing else should be working before you can verify that the tape is at falt.)

Cassette Interface Tests

Program Operation

- Connect the amplifier or cassette to the input/output jacks and adjust the volume at the midpoint.
- Type in the machine level program for the Cassette Loop test (Table 8) and the Cassette Record/Play test (Table 9). (Refer to the C2 or 500 board manuals for instructions on entering machine level programs into the computer).

continued...

address	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
022	A9	00	AA	A8	85	AO	85	A1	85	A2	85	A3	AD	00	FC	4A
023	4A	90	F9	A5	AO	8D	01	FC	AD	00	FC	4A	90	FA	AD	01
024	FC	C5	AO	DO	OF	E6	AO	DO	E3	E6	A2	DO	DF	E6	A3	DO
025	DB	4C	00	FF	85	A1	A9	4D	4C	46	FF	—	—	—	—	—

Table 8

NOTE:

- This first test will check 256 characters continuously for 65,000 cycles.
- Load the starting address 0220 and press the G key. (Restart test if address 0220 immediately changes to 0000. This may happen once or twice however the program should run without any problem at 300 baud. If it fails to run check the amplifier's connections and volume setting).
- When the program fails to compare the character received with that transmitted it will bring up address 0000 on the screen. (Memory address 00A0 to 00A3 will loaded with the information need to analyze the failure).
- Access address 00A0 for the character transmitted and 00A1 for the character received. (The information is in hexadecimal and must be converted to binary so that the bits can be compared).
- Access address 00A3 and 00A2 to compute the number of cycles completed. (A3 is the high byte and A2 is the low byte in hexadecimal).

Cassette Record/Play Test

address	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
026	AO	00	84	AO	AD	00	FC	4A	4A	90	F9	A5	AO	8D	01	FC
027	88	DO	04	E6	AO	FO	03	4C	64	02	4C	00	FF	—	—	—
028	AO	00	AD	00	FC	4A	90	FA	AD	01	FC	99	00	D4	C8	4C
029	82	02	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Table 9

NOTE:

- Operate Record/Play on cassette, load starting address 0260 hex and press the 'G' key.
- When the program is complete the C/W/M? will appear.
- Access machine monitor and load address 0280 hex, press 'G' key then operate Play on the cassette.
- An ASCII character will be displayed 256 times on the screen before a new character is displayed. All 256 characters will be displayed before the program completes. By observing the screen you can detect any failures (different character appears in place of character being displayed).

31 USR Dispatch

Salomon Lederman
40 Waterside Plaza
N.Y., N.Y. 10010

This machine language program makes using machine language subroutines from a Basic program much easier. Instead of the ungainly method of poking locations 11 and 12 with your beginning in low, high byte format, merely enclose the address (in hex) in the parentheses of your USR statement. An example is as follows:

X=USR(FDØØ) .

This would jump to a subroutine located at FDØØ, in this case, the character input routine in the Basic ROM. To initialize the program, do a POKE 11,34:POKE 12,2. And now, the program.

LOCATION	MACHINE CODE	MNEUMONICS
0222	A5 C3	LDA $C3
0224	18	CLC
0225	69 FA	ADC #$FA
0227	85 C3	STA $C3
0229	A5 C4	LDA $C4
022B	69 FF	ADC #$FF
022D	85 C4	STA $C4
022F	A2 00	LDX #00
0231	20 BC 00	JSR $00BC
0234	C9 3A	CMP #$3A
0236	10 05	BPL $05
0238	29 0F	AND #$0F
023A	4C 40 02	JMP $0240
023D	38	SEC
023E	E9 37	SBC #$37
0240	9D FC 02	STA 02FC,X
0243	E8	INX
0244	E0 04	CPX #04
0246	D0 E9	BNE $0231
0248	AD FC 02	LDA $02FC
024B	0A	ASL A
024C	0A	ASL A
024D	0A	ASL A
024E	0A	ASL A
024F	18	CLC
0250	6D FD 02	ADC $02FD
0253	8D 66 02	STA $0266
0256	AD FE 02	LDA $02FE
0259	0A	ASL A
025A	0A	ASL A

continued...

LOCATION	MACHINE CODE	MNEUMONICS
025B	0A	ASL A
025C	0A	ASL A
025D	18	CLC
025E	6D FF 02	ADC $02FF
0261	8D 65 02	STA $0265
0264	20 00 00	JSR $0000
0267	20 BC 00	JSR $00BC
026A	20 BC 00	JSR $00BC
026D	60	RTS

Trig Quiz

Mike Cohen
2255 Barker Avenue
Bronx, N.Y. 10467

This program will quiz you on finding sides of a triangle. It will draw the triangle, give you two sides of it, and ask you to find the third side. You may select the number of questions to answer, and at the end of the quiz, you will be given your score.

```
10 FORM=546TO571:N=PEEK(M+64490):POKEM,N:NEXTM:POKE572,96
15 GOSUB 500
18 K=0
20 PRINT:PRINT:INPUT"HOW MANY QUESTIONS";Q1
25 FOR Q=1 TO Q1:GOSUB1000
30 T=INT(RND(1)*3)+1:ONTGOTO40,80,120
40 PRINT"X=";X:X1=X
50 PRINT"Y=";Y:Y1=Y
60 INPUT"R=";R1
70 GOTO150
80 PRINT"X=";X:X1=X
90 PRINT"R=";R:R1=R
100 INPUT"Y=";Y1
110 GOTO150
120 PRINT"Y=";Y:Y1=Y
130 PRINT"R=";R:R1=R
140 INPUT"X=";X1
150 Z=(ABS(X-X1)/X)+(ABS(Y-Y1)/Y)+(ABS(R-R1)/R)
160 PRINT:PRINT:IFZ=0THEN250
170 IFZ<.08THENPRINT"CLOSE ENOUGH.":K=K+.5:PRINT:PRINT
175 PRINT"THE ACTUAL VALUE WAS"
180 IFX1<>XTHENPRINT"X=";X
185 IFY1<>YTHENPRINT"Y=";Y
190 IFR1<>RTHENPRINT"R=";R
200 GOTO300
```

continued...

```
250 PRINT:PRINT"YOU GOT IT EXACTLY!!!"
260 K=K+1
300 FORM=1TO3500:NEXTM,Q
400 PRINT:PRINT:PRINT"YOUR SCORE WAS";
410 PRINTK/Q1*100
420 END
500 POKE11,34:POKE12,2:X=USR(0)
510 PRINT"TRIGONOMETRY QUIZ"
520 FORM=1TO6:PRINT:NEXT
530 L=4:GOSUB 1030
540 PRINT"YOU WILL BE GIVEN THE   LENGTH OF 2 SIDES AND"
550 PRINT"YOU WILL BE ASKED THE   LENGTH OF THE THIRD SIDE"
560 PRINT"TO KEEP IT SIMPLE, Y    WILL ALWAYS BE 1/2 OF X"
580 PRINT:PRINT"R=SQR((X↑2)+(Y↑2))"
590 PRINT:FORM=1TO1000:NEXT
600 INPUT"READY TO START";X$:RETURN
1000 POKE11,34:POKE12,2:X=USR(X)
1010 DATA196,1,195,-31
1020 L=INT(3+RND(1)*7)
1030 CU=54053
1040 FORM=1TOL
1050 RESTORE
1060 FOR N=1 TO 2
1070 READ C,D
1080 POKE CU,C
1090 CU=CU+D
1100 POKE54082+2*M+N,135
1110 NEXT N
1120 POKE54085+2*L-32*M,136
1130 NEXT M
1140 POKE54022+2*L,121
1150 X=2*L:Y=L
1160 POKE54022,114
1170 PRINT" ";CHR$(120):PRINT
1180 R=SQR(X↑2+L↑2)
1190 RETURN
```

OS-65U . patch

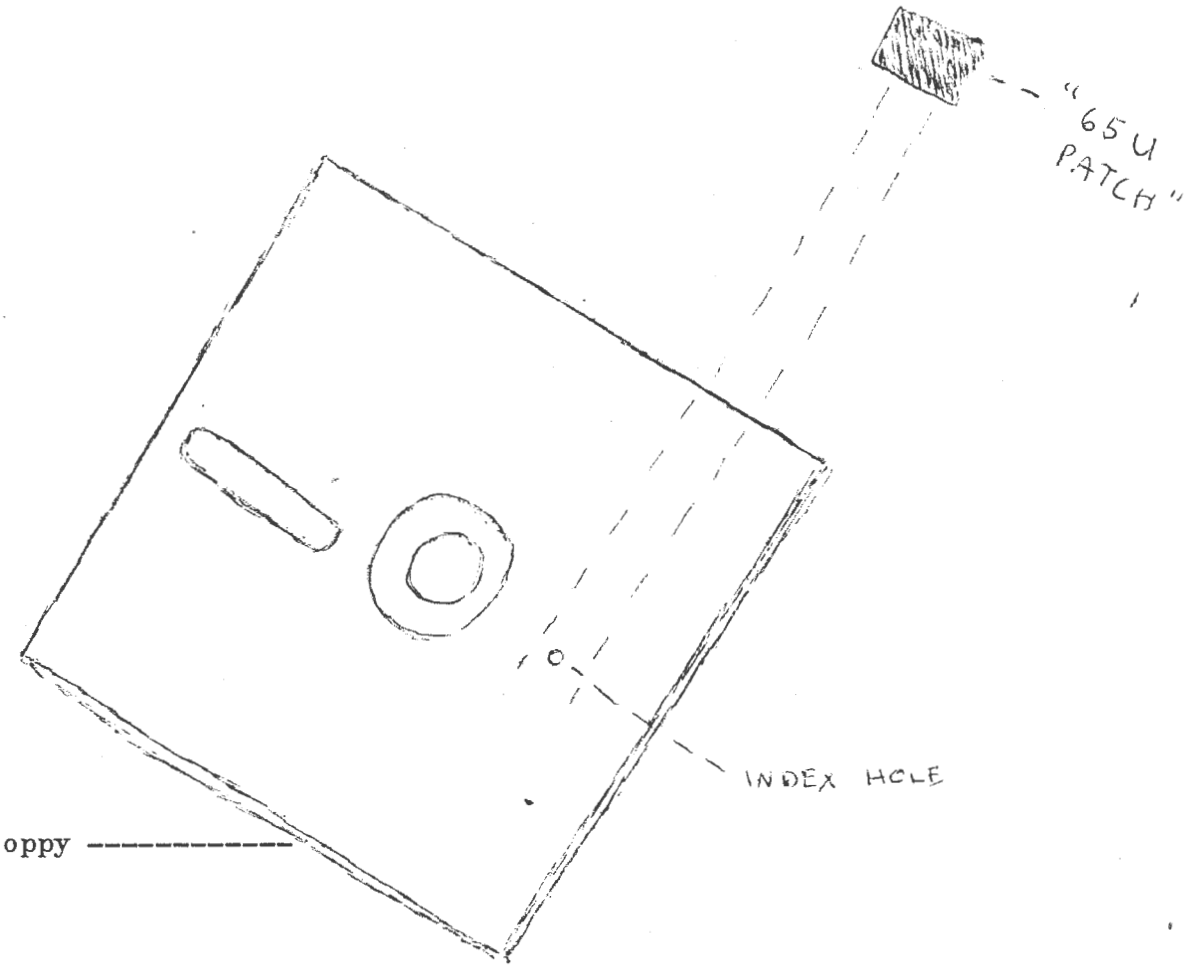
Patch for OS65U v. 1.1 Level 1 through 39

Numerous bugs have recently appeared in the operating system. Instead of going through a bunch of software modifications we have come up with a special hardware mod that eliminates all problems on disk based systems. For this incredible feat of engineering to be inserted, consult the blueprints on the next page.

34

v. 1.9

65U
PATCH



INSTRUCTIONS

Glue 65U patch securely over index hole.

Additional patches for copies may be made from old tech newsletters, OSI ads, or back issues of OSI-tems.

35 Software Reviews

Salomon Lederman
40 Waterside Plaza
N.Y., N.Y. 10010

I've tried to make these reviews as objective as possible. Ratings are from 0 to 10 stars (*).

from Orion Software Associates.

GOMOKU - Here is a very well done program, in machine language. Its allows the computer to play the game of Gomoku against a human opponent. It has a nearly unbeatable strategy, and well done graphics. My only complaint is that it makes its move too quickly, which is very depressing to people like me who can't beat the thing! (9*)

DUNGEON CHASE - A nicely done graphics game. In dungeon Chase you are an adventurer in a 20 - level world. You must destroy beasts, and collect gold as you gain experience. Nice graphics, and will appeal to those among us with high ambitions. (8*)

HEAD ON - This game has magnificient graphics, for a C1, C2... It is an excellent simulation of the arcade game. You ride around a track, collecting points, and avoiding the enemy car, which is computer controlled. Very nice in all respects, and very hard to beat. (9¼*)

TRAP - A pretty good program, modelled after an arcade game, where you have to try to trap an opponent by building walls. This version has various options, barriers, wraparound, various spped levels, and the computer can even play against you if you like. The computer strategy is not that great, but at a high speed it is very hard to beat. There should have been machine language screen and color clear routines to keep up with the rest of the program. (8½*)

continued...

GREMLIN HUNT - A fun game, for up to three players. You must try to kill lots of Gremlins before your opponents do. This game is a bit addicting. The kids will love this one. (7*)

INDY 5000 - It's you against the computer car as you try to cover an arbitrary number of laps before the computer does. You can even design your own track once you understand how the program works. The program bothers me though, because it is not smooth. I know that in Basic this is not easy to do, but it still bothers me. (6*)

GUNFIGHT - Another game where you are out against the computer. Has cute graphics. The only problem is that the real time action is not really so. Once the computer fires, you can not get out of the way. Also, the computer thinks so quickly that it seems to have an unfair advantage. (6*)

from Aardvark Technical Services

CANNONEERS - In this game you are shooting cannonballs over mountains, trying to destroy your opponent. A pretty fun game. Requires some skill to master. You enter the angle, and the amount of powder. (8¼*)

ORBITAL LANDER - This game is not too great. You are supposed to land on these bases that float by, before you run out of fuel. For one or two players. (4*)

ADVENTURE - There are currently two adventures available, in 8K. They are very nicely done. A real mindbender. You must figure out what to do with all the things that you find in order to win. There are some nice tricks which make both adventures fun to play. (9*)

from Aurora Software Associates

DUNGEON - A real fun one player game, based on the D&D games. You have to kill monsters, in order to gain experience, hit points,etc. This one will keep you happy for a long time. (8½*)

37

23 Androids

Mike Cohen
2255 Barker Avenue
Bronx, N.Y. 10467

This is a new variation on the old theme of '23 matches' in which you and the computer pick matches and whoever is left with the last one loses. This one has cute animation and graphics.

```
 10 GOSUB3000
 50 P1=53924:LN=53732:N1=23
 60 POKE11,0:POKE12,253
 70 FORX=1TON1:POKELN+X,INT(240+RND(1)*2):NEXT
 80 X$="WANT TO GO FIRST ?":P2=P1+LEN(X$)+1:GOSUB2500
 90 X=USR(X):X=PEEK(531):POKEP2,X
 95 GOSUB650
 100 IFX=89THENGOSUB2700:POKEP2,32:GOTO200
 110 IFX<>78THEN90
 120 GOSUB2700:POKEP2,32:D=INT(RND(1)*3+1)
 150 IFN1>4THENC=4-D:GOTO165
 155 IFN1=1THEN300
 160 C=N1-1
 165 GOSUB600
 170 X$="I TAKE"+STR$(C):GOSUB2600
 180 GOSUB2500:GOSUB2600
 200 X$="HOW MANY ?":P2=P1+LEN(X$)+1
 201 IFN1=1THEN350
 205 GOSUB2600
 210 GOSUB1500:POKEP2,48+D:IFD<=N1THEN240
 220 PRINT"ILLEGAL MOVE";:FORI=1TO1000:NEXTI
 230 PRINTCHR$(13);SPC(15);CHR$(13);:GOTO210
 235 GOSUB600
 240 GOSUB2000:GOSUB2700:POKEP2,32
 250 IFN1=0THEN400
 260 GOTO150
 300 X$="I TAKE THE LAST ONE":GOSUB2600
 310 FORX=0TO32:POKELN+1,X:NEXTX
 320 PRINT"YOU WIN!!"
 330 GOTO500
 350 X$="YOU MUST TAKE ONE"
 360 GOSUB2600:FORY=1TO4
 361 POKELN+(Y-1)*32+1,32:POKELN+Y*32+1,240
 362 NEXTY
 365 FORX=1TO-1STEP-1
 366 POKELN+129+X,32:POKELN+128+X,240:NEXTX
 370 GOTO500
 400 PRINT"YOU LOSE!!!"
 500 PRINT:PRINT:PRINT
```

continued...

```
510 PRINT"WANT TO TRY AGAIN ?";
520 X=USR(X):X=PEEK(531):X$=CHR$(X)
530 POKE54017+PEEK(512),X:IFX$="N"THEN599
540 IFX$<>"Y"THENPOKE54017+PEEK(512),32:GOTO520
550 PRINT
560 RUN10
599 END
600 FORJ=1TO3*N1
610 F=INT(N1*RND(1)+1):G=PEEK(LN+F)
620 POKELN+F,G+SGN(240.5-G)
630 NEXTJ:RETURN
650 FORJ=1TO8:F=INT(N1*RND(1)+1):G=PEEK(LN+F)
660 POKELN+F,G+SGN(240.5-G)
670 NEXTJ:RETURN
1000 POKE530,1:POKE57088,127:D=255-PEEK(57083):POKE530,0
1010 D=8-INT(LOG(D+.01)/LOG(2)):IFD>3THEND=0
1020 RETURN
1500 GOSUB1000:IFD>0THENRETURN
1510 X=INT(N1*RND(1)+1):Y=PEEK(LN+X)
1520 IFY=240THENPOKELN+X,241
1530 IFY=241THENPOKELN+X,240
1540 GOTO1500
2000 N1=N1-D:FORX=DTO1STEP-1
2005 POKELN+N1+1+X,22
2010 Y=PEEK(LN+N1+X):POKELN+N1+X,32:POKELN+32+N1+X,Y
2020 FORZ=N1+XTO-1STEP-1
2030 POKELN+33+Z,32:POKELN+32+Z,Y
2035 FORJ=1TO4:F=INT(N1*RND(1)+1):G=PEEK(LN+F)
2036 POKELN+F,G+SGN(240.5-G)
2038 NEXTJ
2040 NEXTZ:POKELN+N1+1+X,32
2050 NEXTX:RETURN
2500 N1=N1-C:FORX=CTO1STEP-1
2510 POKELN+N1+X+1,244
2515 GOSUB650
2520 FORZ=0TO32:POKELN+N1+X,Z:NEXTZ
2525 GOSUB650
2530 POKELN+N1+X+1,32:NEXTX
2535 GOSUB650
2540 RETURN
2600 FORI=1TOLEN(X$)
2610 POKEP1+I,ASC(MID$(X$,I)):NEXTI
2620 RETURN
2700 FORI=1TOLEN(X$)
2710 POKEP1+I,32:NEXTI
2720 RETURN
3000 FORX=53248TO54171
3010 POKEX,32:NEXTX
3020 RETURN
```