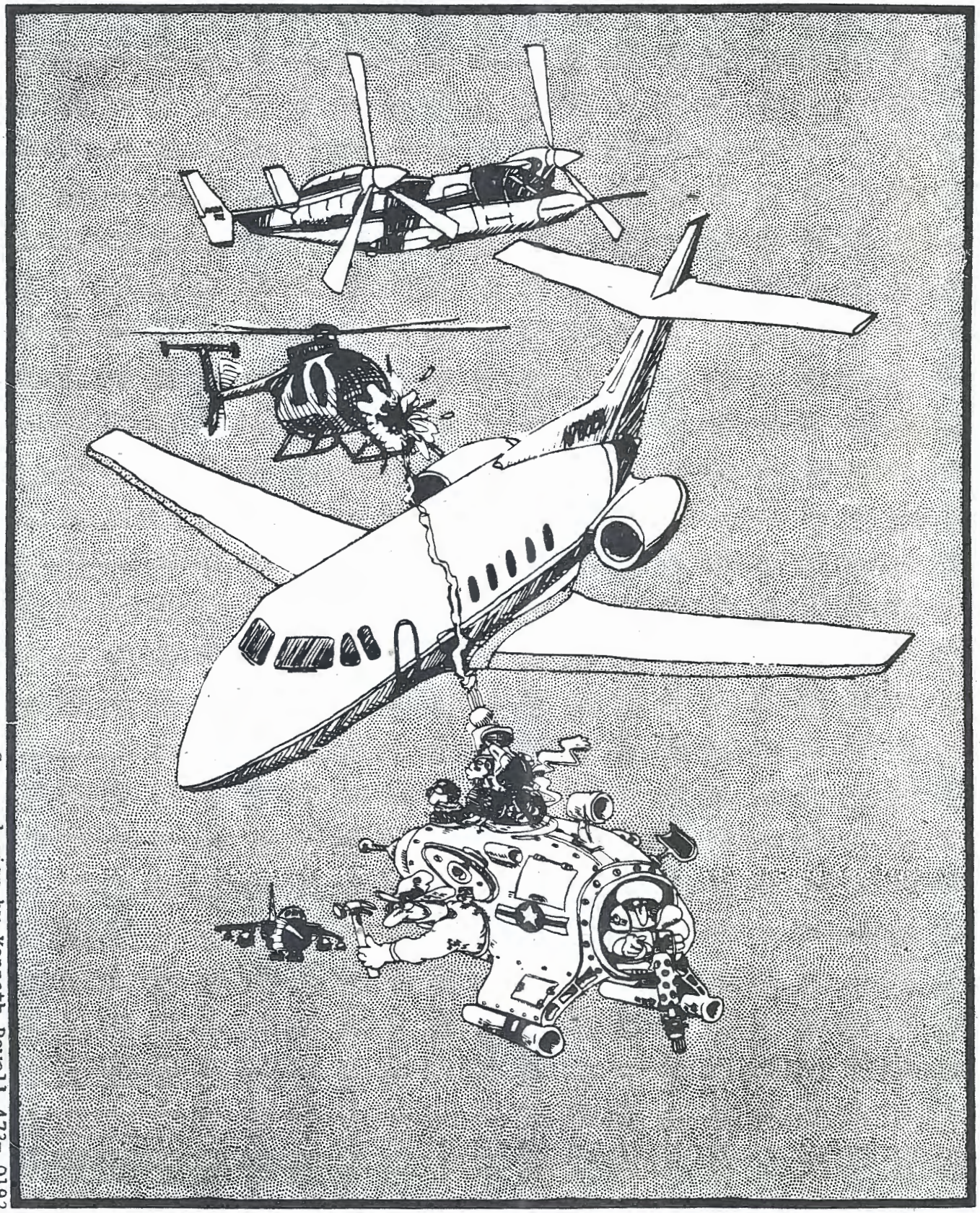
OSI-TEMS by OSUNY

APRIL 1982 VOL.IV NO.3



Cover design by Kenneth Powell 473- 0192

OSI-tems
April 1982 Vol IV No.2,3,&4

OSI-tems is a publication of the Ohio Scientific User's group of NY. Most articles were submitted by members.

cover by Kenneth Powell editor-Shel Sacks
production-Dale Jones

CONTENTS

NOTE$ FROM THE NOTEBOOK

By Shel Sacks, editor

We've got a nice turnout of articles this month. In fact, since we're aiming at 20 or so pages a month, I actually had to leave out 2 articles (but don't fear--you'll see them next month!). We've got a great cover by Ken Powell, 2 articles by Jack Suchy, a new contributor, 2 new columns,and a few more surprises!(ahem.)

As the new FULL-TIME editor (alog with Dale Jones, our production manager), we're rolling right along. By next month, I hope to have a minimal word processor in print in OSI-tems; once that is done, all contributions will be submitted on either paper (ugh!) or cassette! (and, if I can ever get my modem assembled, you'll be able to PHONE articles to me, as well! To facilitate this, I've reprinted both the RS232 and DUMB TERMINAL programs from our June, 1980 issue.

From here on in, it will make things alot easier if all articles are sent directly to me: Shel Sacks 2 Eldorado Blvd. Plainview NY 11803. (516) 681-2388. It will also help if we start to standardize our articles- it'll make them look more professional, and will be easier to format when we do go direct from cassette.

1)single space everything except the heading; paragraphs may be separated by a line, as done here.

2)Title your article--nothing fancy necessary, but something that describes the content. Center the title, and begin the byline ("by...") at the center. see above.

3)do not write page numbers in ink-pencil is fine, and can be erased -makes my job easier!

4)PLEASE get them to meat least one week before our meeting,i.e. the last Thursday of the month is our deadline! Anything received after this will be in the next issue. No great tragedy, but some months we run short, so let's try and get those articles in.

My WP program is coming along, but I'm stuck at the moment--I need three machine language routines, and that's a very bad weakness of mine.If someone would care to write them, it would probably result in completionnof our WP that much sooner. Specifically, I need:

a)a ML routine to transfer a line of screen (from screen Ram) into contiguous memory locations

b)a ML routine to scroll the screen (all lines)

The basic logic of my program is this: to eliminate string-bug, I am not using strings or arrays. The keybo ard is read and outputted directly to the screen, in consecutive locations. Rubout, line-feed, and carriage return work as expected; escape is used for insertion or deletion. As the bottom line is finished, the top line is entered and the screen is scrolled (ML). The top line is entered directly in memory, starting at memory location 2048_{dec}. Output is a simple memory dump, complete with Line feeds and carriage returns. Nothing fancy, but easy to use. Line length is 72 (this can be changed) and editing is done by either calling the line or scrolling to it. There is a warning when memory capacity is reached, at which point you dump to tape; and continue if necessary. I'm allowing 2K for overhead, including the program, so this leaves most of us at least 6K for the article or program. About a page and a halfof typical typing; about a page of typical printer output. If you have more than 8K ram, all the better.

If someone out there has a real wP to share with the club, I'm more than open to it. After all, this is my 1st try! --Shel Sacks

OSUNY Happinings

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

OSUNY has been in a slump these past few months but now things are starting to turn around for the better. Starting this month there are a number of changes that, should improve OSUNY and OSI-tems.

Mr. Shel Sacks will be the full time editor and Mr. Dale Jones the reproduction manager for OSI-tems. These men should bring standardization and regularity to OSI-tems.

I would like to thank them for volunteering for this important task.

Mr. Warren Modell will be assisting in arranging for speakers for the monthly meetings. He would like to get together with a few members to do some planning. Please contact:

Warren Modell
3133 Rochambeau Ave.
Bronx, N.Y. 10467
212-654-3773

if you are interested in helping or would like to provide input.

I will be reporting on the important matters covered at the monthly meetings to keep our non-attending members informed.

Among the many improvements to OSI-tems will be the inclusion of photographs. If you have some photos, send them in with your article and we will print them. How about a photo of yourself (2X2 portrait) to print next to the by-line? Each portrait will be keeped on file for use with future articles. This is a great way to get to know each other. The next time you send in an article send a photo of yourself with it.

This month we start three new columns: Q & A , OSUNY Exchange and this column, OSUNY Happenings, to keep you informed of what is happening within OSUNY and at our monthly meetings.

Q & A is exactly what it says. If you have a problem, any problem, ask a question and some of our members will provide an answer or an opinion. The answer could be one paragraph or a full page article. Through this format we hope to exchange information more readily than we have through our full length articles. Don't stop submitting full length articles. If you have information or are working on something that you feel some other member is interested in then write it up and send it to Shel.

OSUNY Exchange is a marketplace where our members can exchange various items (hardware, software, etc.) for other items or cash. For members there are no fees or restrictions on the number or type of items listed. Non-members can only list computer related items at the rate of $2.00 per item.

Q: Do you have anything to exchange?

A: Yes, it's listed in the Exchange.

** Correction:

The questionnaire that I mailed stated that 6 points were required for one month's renewal. As pointed out by the members it should be 3 points and not 6 as stated. I stand corrected.

** Address Change:

Sometime in April I will be moving to a new job, therefor don't mail anything to my bussiness address. Please send all mail to my home address:

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

President's Remarks

In the latest issue of MICRO I counted 23 companies that advertised OSI products. Someone must be buying these products for these companies to stay in business. How about you? Have you purchased some software or hardware recently? Write up a product review and send it to OSI-tems. Lets share the information, you will be doing a service for the other members and maybe even for the company (increased sales).

Secertary's Report

In February, 76 questionnaries were mailed to all members on our list. As of March 19, 1982 18 questionnaires were returned. In the comming months I will publish the results of the survey along with a list of members. Remember if you don't return the questionnaire then I can't count you as a member and you name will not be on the list.

Replies were sent to 8 prospective members who inquired about OSUNY. I have also written to PEEK 65 and LICA to see if they would like to exchange newsletters with us. If you don't want your article sent to these other groups let me know and I will renove it from the issue before I send them a copy.

Treasurer's Report

In January the club had $ 511.90
Postal expenses for February and March were $ 35.82
Five members renewed their membership. $ 52.00

Total funds in April $ 528.08

Add An RS-232 Port to Your C1P

by Larry Thaler

Since Peripherals are your computer's link to the outside world, they must play an important role in the functioning of any computer. In the Challenger 1P, we are blessed with many of these peripherals, or attachments. Among these, are the built in keyboard, the video output, and the cassettte port. These 'arms' are what allows us to communicate with our computer, and what makes the computer able to communicate with us.

Although O.S.I. included with the Challenger 1P many of these peripherals, it sometimes becomes necessary to add others. If you want to add a printer so you can have hard copy of your programs, or if you'd like your computer to be able to communicate with other computers (to exchange programs, act as an intellegent terminal, etc.), you'll need a special accessory port with signals known as RS-232. These RS-232 signals are universal; That is, the RS-232 signals coming out of a Challenger 1P are the same as those which come out of an IBM 370. This standardization allows us, the computer users, to link devices which were not specially made for our computers. Thus, a printer made for the TRS-80 can be used with the Challenger in the same way that a modem can be used.

Although O.S.I. went through all the trouble to include the printed circuit layout for an RS-232 port, it is unfortunate that they didn't think it was worth their time or money to put the components in. For this reason, we can take up where

O.S.I. left off, and add the necessary components without changing any of the internal wiring of the Challenger 1P.

This article will describe how to add these parts that O.S.I. left off, adding an RS-232 port to your computer. I origonally did it so I could use my computer with a modem, and then be able to use large computer systems from my house. The modification is relly simple, will cost about ten dollars in parts, and take two hours to complete. If you have never done anything like this before, get someone to help you, but if you know how to handle a soledring iron, then by all means, go right ahead. Warning- doing this modification voids your warranty, so if you're going to have trouble with your computer, don't do it.

Here's what you'll need:

quan	description	Radio Shack #	price each	OSI part
1	7417 Integrated Cir Chip	*	*	U68
1	74LS14 I.C. Chip	*	*	U67
6	220 ohm resistors	271-015	2/19¢	R38-48 (even)
6	390 ohm resistors	271-018	2/19¢	R39-49 (odd)
3	10K ohm resistors	271-034	2/19¢	R's 63, 64,66
1	4700 ohm resistor	271-030	2/19¢	R62
1	1000 ohm resistor	271-023	2/19¢	R72
1	470 ohm resistor	271-019	2/19¢	R65
1	1N914 diode(1N4048)	276-1122	10/99¢	D16
1	2N 4401 NPN Transistor	276-2009	79¢	Q2
1	2N 2907 PNP Transistor	276-2023	79¢	Q1
1	DPDT SWITCH on-none-on	275-614	$2.19	SW1
2	16 Pin DIP sockets	276-1998	2/89¢	U67,68

You'll also need:

1½ feet hook up wire solder soldering iron
diagonal cutters wire strippers exacto knife
drill w/ ¼" bit.

* Radio Shack does not sell these components, they can be purchased from Hobby World 19511 Business Center Dr. Nothridge, Ca. 91324 (800) 423-5387 The cost is about 25¢ for U68 and 1.25 for U67. They can also be ordered through many elelectronic hobby places.

The first thing you have to do is find a place to install the switch (sw1). I put mine on the front lip of the computer, next to the plaque which says 'Challenger 1P'. This is a good place for it since it can be reached easily, and does not interfere with the installation of a '610' expansion board. First, I used a center punch to make an indentation where I wanted the hole. Next, I drilled out the hole using a 1/8"drill bit. Then, I widened the hole with a 1/4" bit. When you do this, be sure to remove the cover of the computer, and do the drilling far away from the 600 board, otherwise, little metal filings will land in your computer, and the next time you turn it on, you'll get a great suprise.

Electronic Instructions

1. Install the switch at W10. (see parts layout and schematic). Cut the jumper shorting what OSI calls W10. Connect switch according to schematic diagram. Keep the leads as short as possible. Note: OSI sheet 6 (in your owner's manual) lists 2 W10's, ours is the one on the right side of the page.
2. Insert sockets for U67 and U68 into their respective spaces on the board. Be sure pin 1 (the one with the cut, or hole in it) is facing in the right direction. Solder the leads from the bottom of the board. The plated-through holes will allow the solder to make good contact on the top as well.
3. Install resistors. All the resistors get inserted standing up. Note the way OSI installed their resistors. Solder all the resistors from the bottom of the board only. Be sure none of the leads are touching each other. Cut off the

excess leads from the bottom of the board using the diagonal cutters.

4. Install diode. D16 gets installed lying down, not like the resistors; see the way OSI inserted D1. Be sure you put the diode in the right direction. The part with the line is called the cathode, and goes in the same direction as the arrow points on the schematic.
5. Install transistors. Be careful when soldering, too much heat and they'll burn out. Be sure to install them in the right positon, the base is the one in the center, the emitter has an arrow, and the collector doesn't have anything. See the spec. sheets that come with the transistors to find which pins correspond.
6. Insert the IC's into their sockets. Be sure you put the right IC's in their correct sockets. Be sure pin 1 is pointing in the right direction.
7. Check the bottom of the board for uncut resistor/diode/transistor leads. Check for any solder splashes (solder in the wrong places). Be sure there are no extra bits of wire lying around. Check the top of the board for all these too.
8. Install your modem or printer to J3 pin 4 (Rx Data) pin 2 (Tx Data) and pin 1(ground). If oyur printer needs hand shaking, you'll have to hook up RTS and CTS also, and modify W3 (another article to come). If your device needs a negative voltage source, then cut the jumper at w10A (the lower one on sheet 6) and connect a negative voltage source to pin 7 (J3). A nine volt battery (+terminal to ground) should work just fine.

Now, you have RS-232 signals to use your modem or printer with. If the mod is not working, check your connections, and the polarity of the transistors and diode, not to mention the IC's! If your computer is not working at all, turn off the power FAST!!!!!! Go through step 7 again, very carefully. If there's still a problem, get help from a friend, sometimes you just miss something that you did.

Our frinds who decide thestandards for computer signals did us a real favor when they invented RS-232. Now, we like all the other computer hobbyists can enjoy the large list of peripherals available, and make the most out of our computers.

Editor's note: For a working dumb terminal program, see the listing by Danny Schwartz on page 17 of this issue.

SOUND FROM YOUR 1P

-A.J. SUCHY

OK, Series II owners, with the addition of a simple amplifier system (such as the $11.95 amplifier sold by --ugh-- Radio Shack) connected to your DAC, you can start developing sound subroutines. Unless you are skilled in machine language techniques, which I am not, you'd better get the Aardvark compiler too.

The CIP USERS MANUAL from OSI (a disappointment generally) gives little else than a general indication of how to use the DAC to make sound. After you insert the two leads from the amplifier and speaker system to DAC you can start sending signals to your DAC. You must enable the DAC by POKING 55296 with 16 (use 17 instead of 16 if you are in the 12x48 mode). Series I users generally don't install a software oriented DAC disable so for those of you who populate your Series I DAC, all you need to do is turn on your DAC. A software DAC enable is not required for your Series I.

The next step in the process is to POKE various sequences of values into the DAC output port (located at 57088--series I and II both). The last step for the Series II users is to turn off your DAC by POKING 55296 with a 0(use 1 in the 12x48 mode). This DAC disabling is not essential to sound production, but the shrill tone produced after you "play your tune" will dictate that you turn off the DAC. Series I users will disable the DAC manually until they wire their own hardware DAC disable.

Now the tricky part in music generation is not in plugging in the speaker amplifier or in enabling the DAC, but in sending to 57088 the right numbers fast enough. Let's say you are going to POKE your DAC 1000 times. If you poke it high (POKE 57088,255) that many times, nothing is noticed, just as poking it low (POKE 57088,0) produces nothing. But if you poke it low 100 times (FORX=1TO100:POKE57088, 0 :NEXT) then high 100 times, then low, etc back and forth until you have poked the DAC a total of 1000 times.* One additional trick is that you better be sending the pulses at machine code speeds, otherwise all you will hear is a series of clicks rather than a tone. This technique is called generating a square wave.

To vary the sound, a number of things can be done. For example, instead of shifting high or low every 100 pokes, you can shift every 50 or 25 pokes. Since you by this change create more boxes (waves), the frequency goes up & a higher pitch is produced. Sound engineers call the width of the plateau or wave the "sustain".

Another variation you can introduce instead of poking each square wave completely high or low is to poke in the values of say 0's and 50's. Doing this makes the sound quieter since the DAC "amplitude" output decreases. Its like hitting a gong softer.

A final variation you can introduce instead of using a square wave is to vary the wave form. you can slowly build up to a peak in the wave (attack), then introduce some bumps and jiggles into the high points of the wave, and then let the wave die out (decay) or go back to poking 0's . It is clear that if you do 2000 pokes, the sound goes on for twice as long.

*you get sound.

A piano tuning book I own gives the frequencies of the 8 notes of the scale as 128, 144, 164, 171, 196, 215,242 and 256. The last frequency is twice that of the first corresponding on the piano to notes that are an octave apart. The spacing of the notes between is only approximate, but can be used by you sound experimenters to generate a scale.

I am enclosing a copy of a basic language program which will compile in the Aardvark compiler and produces the tones of a chord (the first, third, fifth and eighth notes in the scale.) I am also leaving in the unused gosubs for producing a whole scale. It is possible with a little work to generate a sound routine and have real sound in your C1P without resorting to sound generation. Perhaps in the near future, you eprom types can come up with a sound command rom with some of these suggestions. Probably the easiest technique is to develope a USR routine that PEEKs a location such as $222 for note, $223 for time or duration, and $224 for loudness and then executes the DAC output pokes that are appropriate. In this way you can from basic "load" the USR subroutine and play the note, go back to basic and load another note and doing this time after time play a whole song.

Have fun with your new musical instrument!!

REFERENCES: DIGITAL MUSIC SYNTHESIS, Creative Computing June 81, p. 88

" " July 81, p. 140

SUPER SOUND FOR YOUR SUPERBOARD, Kilobaud, Dec. 1980, p.130

MUSIC PRIMER, Personal Computing, July 1981, p.87

A.J. SUCHY
5538 North Thornwood Ave.
Davenport, Iowa 52806

```
5 Z=55296
10 S=57088
15 F=1
18 G=1
20 R=0
25 A=3000
30 B=1500
35 C=4
100 POKEZ,16
200 L=3000
300 N=N+G
310 GOSUB3100:GOSUB1100
330 GOSUB3300:GOSUB1100
350 GOSUB3500:GOSUB1100
380 GOSUB3800:GOSUB1100
500 IFN=CTHENGOTO7000
510 GOTO300
1100FORI=1TOL
1200R=R+F
1300IFR=ATHENGOSUB2000
1305IFA<RTHENGOSUB2000
1310 T=0
1320IFB<RTHENGOSUB2020
1400POKES,T
1500 NEXTI
1610 RETURN
2000 R=0
2010 RETURN
2020 T=255
2030 RETURN
3100 F=128
3110 RETURN
3200 F=144
3210 RETURN
3300 F=164
3310 RETURN
3400 F=171
3410 RETURN
3500 F=196
3510 RETURN
3600 F=215
3610 RETURN
3700 F=242
3710 RETURN
3800 F=256
3810 RETURN
7000 POKE Z,0
7999END
```

Q & A

Q: Has anyone replaced the 6520 PIA with a 6522 VIA? How did you make the hardware and software changes?

Ugo V. Re'
167 Sprucewood Dr.
Levittown N.Y. 11756

Q: Does anyone have a circuit diagram of a data seperator that can be used with the USI disk controller? The FDDS chip did not work, article this issue.

Ugo V. Re'

Q: The Aardvark journal listed two POKEs to make the Cegmon operate at 600 baud without garbaging up the initial letters of the second line. Does anyone know any similar techniques for the CIS ROM? The ROM routines at $13 and $538 are similar in both (I think) but it doesn't appear to work.

Jack Suchy
5538 N. Thornwood Ave.
Davenport, Ia. 52806

Q: Is there anyone with a machine language routine which by software technique only will output at 110 baud-suitable for driving a teletype, without a hardware modification? Aardvark's suggested new chip works pretty well, but not perfectly. Any suggestions?

Jack Suchy

OS65D V3.3

By Mike Cohen

OSI finally did something right when they came out with OS65D Version 3.3. It contains many features that all earlier disk operating systems lacked, including 65U. Some of the major enhancements are a 'normal' keyboard routine, a video driver even better than the one in CEGMON, and several BASIC extensions, including forms of PRINT AT, PRINT USING, and even a form of error trapping. Also, a DISK FIND command similar to 65U has been added.

Less obvious are improved disk access and better random access files, with no unnecessary disk accesses. In older versions of 65D, a DISK GET would read a track from disk even if that track was already in the buffer. Version 3.3 will not read a track if it's already in the buffer.

Basic has been modified to take advantage of the new keyboard and video drivers. BASIC now allows lower case to be used freely, no longer giving a syntax error if commands are entered in lower case and considers upper and lower case letters to be equivalent in string comparisons.

Also, the PRINT command has been extended in several ways. PRINT &(X,Y) will position the cursor at (X,Y). PRINT !(n) is used to send escape sequences and is equivalent to PRINT CHR$(27)+CHR$(n). PRINT USING "#####" allows neat dollar and cents formatting, although is limited to only "#" and "." in the format string, which can only be up to 17 characters long. This means you can't insert commas and dollar signs in a number. Still, it's better than nothing, and you can't expect OSI to do everything!

Also, when outputting to a printer, there is an optional automatic paging feature, enabled by PRINT #d,!(67,n) where 'd' is the printer device number and 'n' is the number of lines per page. When this is enabled, however, sending CHR$(12) to that device number will output continuous linefeeds rather than a formfeed character. This is useful for printers which don't respond to formfeed, but for the EPSON, which 3.3 is supposed to be tailored to, this only slows things down. Also, PRINT #d,!(80) is supposed to print a screen dump on a MX-80, although I haven't been able to get it to work.

Another new command is TRAP which is equivalent to ON ERROR GOTO. It provides a way for a program to remain in control even if an error occurs, although it suffers from 2 minor problems. First, the error message still appears on the screen, although it can be erased by moving the cursor and writing over it. Also, when TRAP is in effect, it will also trap a control-c and cause the break message to appear twice.

There doesn't seem to be any way to find out which error occurred, but if anyone figures out how, I would like to hear it.

Here is a phone book program which demonstrates some features of 3.3. These are the control codes used and their function:

```
!(28).....Clear screen
!(12).....Cursor up 1 line
!(15).....Erase to end of line
```

(note: Before using it the first time, RUN 8000 to initialize the file)

```
10 REM phone directory program
20 GOSUB 200
30 PRINT!(28);"Phone directory":PRINT
40 PRINT"<A>dd, <C>ondense, <D>elete, <P>rint, <F>ind"
50 INPUT "or <Q>uit";f$:f$=LEFT$(f$+"*",1)
60 IF f$="a" THEN 1000
70 IF f$="c" THEN 2000
80 IF f$="d" THEN 3000
90 IF f$="f" THEN 4000
100 IF f$="p" THEN 5000
110 IF f$="q" THEN 9000
120 GOTO 30
200 POKE 2888,0:POKE 8722,0:TRAP 0
210 DEF FN dec(bcd) = (bcd AND 15)+ (INT(bcd/16)*10)
220 RETURN
500 ab=PEEK(12042)*(FNdec(PEEK(9004))-FNdec(PEEK(9002)))
510 re=(PEEK(9133)-PEEK(8999))*256+PEEK(9132)-PEEK(8998)-1
520 rn=ab+INT(re/2^PEEK(12076))
530 DISK get,rn:RETURN
1000 TRAP 1900:DISK open,6,"ph#"
1010 DISK get,0:INPUT #6,num
1020 num=num+1
1030 PRINT!(28)"Name <return> to end";
1040 INPUT name$:IF name$="" THEN 1200
1050 INPUT"Address ";addr$
1060 INPUT"Area Code";area$
1070 INPUT"Phone Number";num$
1080 DISK get,num
1090 PRINT #6,name$; ","; addr$; ","; area$; ","; num$
1100 DISK put
1110 GOTO 1020
1200 num=num-1
1210 DISK get,0:PRINT#6, num:DISK put:DISK close,6
1220 GOTO 20
1900 PRINT !(12) !(15) " *** Disk error! ***"
```

```
1910 GOTO 3100
2000 TRAP 1900:DISK open,6,"ph#"
2010 DISK get,0:INPUT#6,num
2020 FOR i=1 TO num:DISK get,i:INPUT#6,flag$,b$,c$,d$
2030 IF flag$<>"####" THEN 2090
2040 num=num-1
2050 FOR j=i+1 TO num+1
2060 DISK get,j:INPUT#6,a$,b$,c$,d$
2070 DISK get,j-1:PRINT#6,a$;",";b$;",";c$;",";d$:DISK put
2080 NEXT j
2090 NEXT i
2100 DISK get,0:PRINT#6,num:DISK put:DISK close,6
2110 PRINT "Completed.":GOTO 3100
3000 TRAP 3900:DISK open,6,"ph#"
3010 INPUT "Name to delete";name$
3020 DISK find,name$:GOSUB 500:flag$="####"
3030 INPUT #6,name$,addr$,area$,num$
3040 PRINT !(28);Name$:PRINT Addr$: PRINTArea$,Num$
3050 PRINT:INPUT"Do you want to delete this one";y$:y$=LEFT$(y$+"n",1)
3060 IF y$<>"y" THEN 3020
3070 DISK get,rn:FOR i=1 TO 4:PRINT#6,flag$:NEXT i:DISK put
3080 DISK close,6
3090 PRINT"Deleted."
3100 PRINT:INPUT "Hit <RETURN> to continue";a$:GOTO 20
3900 PRINT !(12) !(15) "Record not found."
3910 GOTO 3100
4000 TRAP 3900:DISK open,6,"ph#"
4010 PRINT!(28);:INPUT"Name";a$
4020 DISK find,a$:GOSUB500:INPUT#6,name$,addr$,area$,num$
4030 PRINT !(28);"Name       - ";name$
4040 PRINT"Address    - ";addr$
4050 PRINT"Area Code - ";area$
4060 PRINT"Number     - ";num$
4070 PRINT:INPUT"Continue search";y$:y$=LEFT$(y$+"n",1)
4080 IF y$="y" THEN 4020
4090 PRINT:INPUT"Want to change this record";y$:y$=LEFT$(y$+"n",1)
4100 IF y$<>"y" THEN 3100
4110 PRINT !(28);"Name       - ";name$
4120 PRINTTAB(10);!(12);:INPUTa$:IFa$<>"" THEN name$=a$
4130 PRINT"Address    - ";addr$
4140 PRINTTAB(10);!(12);:INPUTb$:IFb$<>"" THEN addr$=b$
4150 PRINT"Area Code - ";area$
4160 PRINTTAB(10);!(12);:INPUTc$:IFc$<>"" THEN area$=c$
4170 PRINT"Number     - ";num$
4180 PRINTTAB(10);!(12);:INPUTd$:IFd$<>"" THEN num$=d$
4190 DISK get,rn
4200 PRINT #6,name$; ","; addr$; ","; area$; ","; num$
4210 DISK put:DISK close,6:GOTO 3100
5000 TRAP 1900:DISK open,6,"ph#":GOSUB 5100
5010 DISK get,0:INPUT#6,num
5020 FOR i=1 TO num:DISK get,i
5030 INPUT#6,a$,b$,c$,d$:IFa$="####"THEN5060
5040 PRINT#d,cl$;a$:PRINT#d,b$:PRINT#d,c$,d$
5050 PRINT#d:IFd=2THENPRINT:INPUT"hit <return> to continue";a$
5060 NEXT i
5070 GOTO 20
5100 PRINT!(28):INPUT"Output to printer";p$:p$=LEFT$(p$+"n",1)
5110 IF p$="y" THEN 5130
5120 cl$=CHR$(27)+CHR$(28):d=2:RETURN
5130 cl$="------------------------------------------------"
5140 cl$=cl$+CHR$(13)+CHR$(10):d=1:RETURN
8000DISK open,6,"ph#":DISK get,0:PRINT#6,"0":DISK put:DISK close,6
9000 PRINT "Good-bye."
9999 END
```

1 KEYSTROKE PROGRAMMING UTILITY

Jack (A.J.)Suchy

Here is a modified 1 keystroke programming utility in BASIC modified from some old MICRO articles. Its is relocated to start at $236 so as to be compatable with a C1S ROM. For some reason it doesn't appear to function with the C1E ROM though I tried to relocate the code to do this, also. I've also modified the program so that M gives a MID$ and R a RETURN and G a GOSUB and many other logically related shorthand cues (e.g., J(ump) for GOTO).

```
54000 REM1 KEYSTROKE PROGRAMMING UTILITY-C1S ROM
55000 REM ADAPTED FPOM SEVERAL ISSUES OF MICRO
55500 REM JACK SUCHY   5538 NOPTH THORNWOOD AVE DAVENPORT IA 52806
56000 FORX=566TO658
56010 PEADP
56020 POKEX,P
56030 NEXTX
56100 DATA32,186,255,201,27
56110 DATA240,1,96,152,72,138,72,169,18,141
56120 DATA1,2,32,194,191,162,67,32,186,255
56130 DATA 221,148,2,240,6,202,16,248,76,74
56140 DATA 2,232,160,255,202,240,8,200,185,132
56150 DATA 160,16,250,48,245,104,170,200,185,132
56160 DATA 160,48,15,224,71,176,4,149,19,232
56170 DATA 44,169,7,32,229,168,208,235,41,127
56180 DATA 224,71,176,4,149,19,232,44,169,7
56190 DATA32,229,168,104,168,76,54,2
56300 FORY=660TO727
56310 PEADQ
56320 POKEY,Q
56330 NEXTY
56340 POKE537,2:POKE536,54
56400 DATA0,70,78,68,80,0,81,0,74,57
56410 DATA73,0,71,82,0,0,49,0,0,48,56
56420 DATA0,79,63,0,55,0,0,45,50,0
56430 DATA0,84,0,0,0,0,0,0,0,0
56440 DATA0,0,0,0,0,0,66,85,0,0
56450 DATA0,90,0,0,0,0,0,0,69,76
56460 DATA83,86,65,67,56,57,77
56500 NEW
READY
```

Exchange

Exchange: My knowledge of OSI hardware & software for your knowledge of same. All knowledge to be printed in OSI-tems.

Ugo V. Re'

For sale: 6-32 hex head screws used on OSI systems. $1.75 per doz.

Ugo V. Re'

For sale: Thermal paper rolls (8 3/4 in. X 100ft.) for use in Texas Instrument, Silent 700 or similar terminal. $ 2.00 per roll.

Ugo V. Re'

Exchange: I have a Russian Vocabulary Quiz program for CIP machines which I will like to exchange with someone. The Russian charactors are fully displayed. I would like to exchange this program for a copy of an Adventure game per Mike Bassman.

Jack Sucky

Buy or Trade: I am interested in obtaining a used 610 board to try to upgrade to disk. Anyone interested in selling or trading a 610 board or a 610 board with disk attached?

Jack Suchy

For Sale; my completely <u>untouched</u> modem kit. In original box - Complete with instructions, resistors, diodes, dashpots, geegaws, himalayas, LFB's, AFD's (low speed), AT+T's, and a spare (used) rear axle bearing. No time to assemble. $45.00 takes <u>it ALL</u>.

See Shel Sacks
your friendly local OSI-tems editor.

Dumb Terminal

By Danny Schwartz

This program allows you to turn your C1P into a dumb terminal for use with time-share systems. To use it, type in

POKE 122,4:POKE1024,0:NEW

before using it, or loading the program. The program works on an even parity system, and has been tested on such systems as CUNY (CALOS, WYLBUR), DEC 20, HP 2000, and PDP11.

```
 0 REM BE SURE TO TYPE "POKE122,4:POKE1024,0:NEW"BEFORE LOADING PROGRAM
 1 REM ***************
 2 REM
 3 REM  DUMB TERMINAL
 4 REM
 5 REM       FOR C1P
 6 REM
 7 REM  BY D. SCHWARTZ
 8 REM
 9 REM ***************
 10 DATA 173,0,240,74,144,12,173,1,240,41,127,201,127,240,3,32
 20 DATA 45,191
 21 FORX=754TO771:READI:POKEX,I:NEXT
 30 FORX=772TO953:I=PEEK(64000+X):POKEX,I:NEXT
 40 POKE853,156:POKE884,94
 50 REM EVEN PARITY
 51 DATA 160,9,162,0,24,42,144,1,232,136,208,249,133,17,138,41,1,24,106
 60 DATA 106,5,17,32,177,252,76,242,2
 70 FORX=954TO981:READI:POKEX,I:NEXT
 100 PRINT"READY":POKE11,242:POKE12,2:X=USR(X)
 190 END
OK
```

MY BEEPING C1

Dave Cantrell & Terry Terrance

No, this is not an article full of expletive-deleted condemnation of the Challenger 1P. It is an article on how, with 3 chips and a handful of other components, you can make your C1 sound like a "real" computer, and you won't have to buy a $30.00 or $60.00 ROM and tie up your C1's CTS line to do it.

The background for this project started several months ago when I picked-up a copy of the Fall 1981 issue of "Radio Electronics Special Projects". For those of you who are unfamiliar with this publication, it is a projects-only (i.e. no editorial content, reviews, etc.) quarterly. The project that attracted me to the magazine turned out to be a turkey, as were most of the other articles, and in desparation I read the final project in the book entitled "Build a Beeper for Your TRS-80".

The premise of the project was good, that is the TRS-80, like some other computers that we know of, has no tactile or audible feedback for keyboard entry; and looking at the screen to assure correct entries is not condusive to good typing practices. The solution that the author implemented on his TRS-80 was a small add-on board that caused a piezo beeper to sound off every time a key was pressed AND the video changed within a specified time period.

Further reading of the article revealed something I had not known until now; the TRS-80's keyboard hardware is remarkably close to a C1's. A row is latched by keyboard scanning software while eight key columns are held high by pull-up resistors until a key is pressed, at which point they go low -- exactly the situation on a C1. Hopefully I read on, only to find that the circuit's operation also required a certain video signal. Undaunted, I pulled out the C1 schematics and found several similar video signals one of which should work, or could be made to work. At this point I resolved to send to the listed supplier for the etched and drilled PC board.

The reply that I received from the supplier illustrates the difference between OSI people and TRS-80 people. He apparently had had NO inquiries for the board, and, therefore, could not fill an order simply for one. I didn't have the heart to write back telling him I wasn't even a TRS-80 owner. I guess TRS-80 owners wanted to be part of the "herd" so that they could buy plug-ins. Whereas a lot of OSI owners are not afraid of a soldering gun and will "roll their own". By the way, this project is not a bad first or second project in as much as you will not have to do any cutting on your board and the soldering that is necessary is not difficult or extensive. Actually, this project could be built on a solderless breadboard, perfboard, in the proto area, etc. and connected to the Superboard by means of alligator-clip jumpers until such time as you are convinced that you want to make a permanent connection.

The problem of what to do for a PC board was, ironically, solved by another article in the same magazine from which this project came. On

page 36 of this same issue was an article entitled "Custom printed Circuits for your Projects". This piece dealt with how to use the new "rub-on" (dry transfer) pc resist. It looked like just what I needed; so I made a trip to my local Radio Shack (I know, I'm sorry, I'll do penance for a week in hopes of purification) and bought the following items: Dry transfer resist, a $2.00 package of assorted PC material, and a bottle of PC board etchant. The total for these items came to about the same $6.00 that the PC board would have cost, and I still have plenty of material left over to do other boards. Let me point out that it is not necessary to assemble this project on a PC card, it will fit very neatly into the proto area of your 600 board, using either wire-wrap or tack soldering techniques. However, it is also a good project for which to etch your first PC board if you follow the steps in the abovementioned article carefully. In addition to the items I picked up at "The Shack" (oh boy, the contamination's worse than I thought, I'll have to do another week's penance), I used a plactic tray to etch the board in (any old plastic or glass container you have around will do, so long as you can lay the board down in it - DO NOT use a metal container), as well as a drill and a #58 drill bit (available for about $ 0.25 at a well-stocked model shop). To go into the etching details here would be too long, so if you buy the magazine or Xerox the construction article be sure to look into the etching article.

About 2 hours after I started on the board (I'm a slow worker) and following the etching article closely, I had a perfectly good, and very professional-looking, PC card. Drilling the board was a pain, but it came out all right. Populating (i.e. soldering on the parts) was going well until I couldn't find any .047 caps (Radio Shack should have these). The project was then shelved for a couple of months by other things, but it finally got done, thanks to Dave's willingness to make other peoples' non-completed projects work.

The theory of operation of the board is this (please refer to the ORIGINAL schematic, hereafter all IC's on the 600/Superboard will be refered to with "U" letter designations and all chips on the beeper board will have "IC" designation). The inputs of the 8-input NAND gate (74LS30, IC1) are connected to the columns of the keyboard at the pull-up resistors on your 600 board (Sam's designations R 1-8). The columns and therefore the NAND gate inputs, are held high by these same resistors until a key is pressed, at which point a single input goes low. Quick reference to a truth table for the 74LS30 shows that when this happens the gate's output will go high. This output is then inverted by using one gate on the 74LS00 quad 2-input NAND (IC2 of the beeper circuit) as an inverter. This active-low signal, which is labeled $\overline{KF}$, is then fed to 1/2 of a 556 (IC3) timer wired as a one-shot. The output of the one-shot, is controlled by the .047 caps connected to the 556, and in the original circuit is 50 ms long. Meanwhile, remember that this circuit is not supposed to trigger unless the video acknowledges a change. The original article had some TRS-80 signal fed through an inverter and then into our circuit at hand. An equivalent signal can be found on your C1's U56 pin 6 (Sam's calls this signal $\overline{VA}$). We have to invert this signal too. This can be done using an unused gate in chip U18. Jumper U56 pin 6 to U18 pin 9. Now the active high signal VA can be had on U18 pin 8 and that can be brought into IC2 pin 10 of the original beeper circuit. Another gate in IC2 now combines the output of the one-shot and the VA signal to

create an active low AUDTRIG signal. This is fed to the other half of the 556, again wired as a one-shot, which feeds the piezo buzzer for 50 ms.

The original circuit can be used as is following the hook-up outlined above, however, the circuit does have some problems. Fortunately, Dave has come up with some fixes.

Since the C1 is, as we well know, faster than the TRS-80, the 50 ms time constants on the two one shots allow the beeper to stay on for too long. The result is that even a moderately fast typist can make the beeper go almost constantly. This is easily remedied by changing C1 and C2 (on the beeper board) to 0.01 caps, giving time constants of about 11 ms. resulting in a very "big computer" sound.

You don't want to connect a 74LS30 (IC1) input to the column with the shift lock key, or the thing will sound-off constantly. Simply do not connect a column wire to R4 on your 600 board, tie the unused input to +5. A better use of the unused column line would be to tie it to the BREAK key so that your BREAK key, not normally in the polled-keyboard matrix, will now give an audible output.

One final annoyance occurs when one is doing a listing. Many people have the habit of holding the "CTRL" key down and allowing the listing to scroll by while scanning for some lines of code. When these lines appear they hit the "C" key to stop the listing. When this technique is used with the beeper board, however, it annoyingly beeps as long as the "CTRL" key is held down. This can be fixed without adding any components at all to the beeper circuit simply by making use of the two unused gates in IC2.

The schematic for the "Improved" circuit shows the changes that are necessary. The VA signal, coming onto the beeper board after inversion by U18, is now fed to pin 1 of IC2. Pin 2 of IC2 now receives the signal coming from the ROW 2 latch of the 600 board. This line can be picked off of the anode side of D5 (Sam's designation) or the keyboard side of R68 (the usually unpopulated "noise port" on Series 1 machines). Remember this is a ROW line we are sampling here and not a column. The NANDed signal is now taken from pin 3, fed to both pins 12 and 13 of IC2 where an inversion takes place. The resultant signal is brought from pin 11 to pin 10 (again of IC2) where the VA signal originally went.

These modifications allow one to press the "CTRL" key, which will result in a single beep, hold the "CTRL" key down without any addtional beeps until the "C" key is pressed, which will give another beep. In all very, very nifty!

One final and obvious modification is an on/off switch because there will be times when you will want to shut this little bugger up. Rather than turn off the power to the board, which could result in damage to the IC's, install a SPST switch in the line going from IC3 pin 9 to the buzzer. Switching on and off could be made software controllable, but this would require adding another chip and making a connection to the 6850; this is probably best left to another article.

C4 users are on their own for converting this circuit. Probably the

easiest way to do it would be to invert the column lines coming onto the beeper board with a couple of hex inverters (it's a shame, but after one line is tied to the break key, you'll still need 7 inverters adding two extra chips). Otherwise, you'll have to create the equivalent of an eight-input OR gate. You'll also have to find the equivalent of the C1's VA signal, but with all of the signals available on the 540 board you should have no trouble.

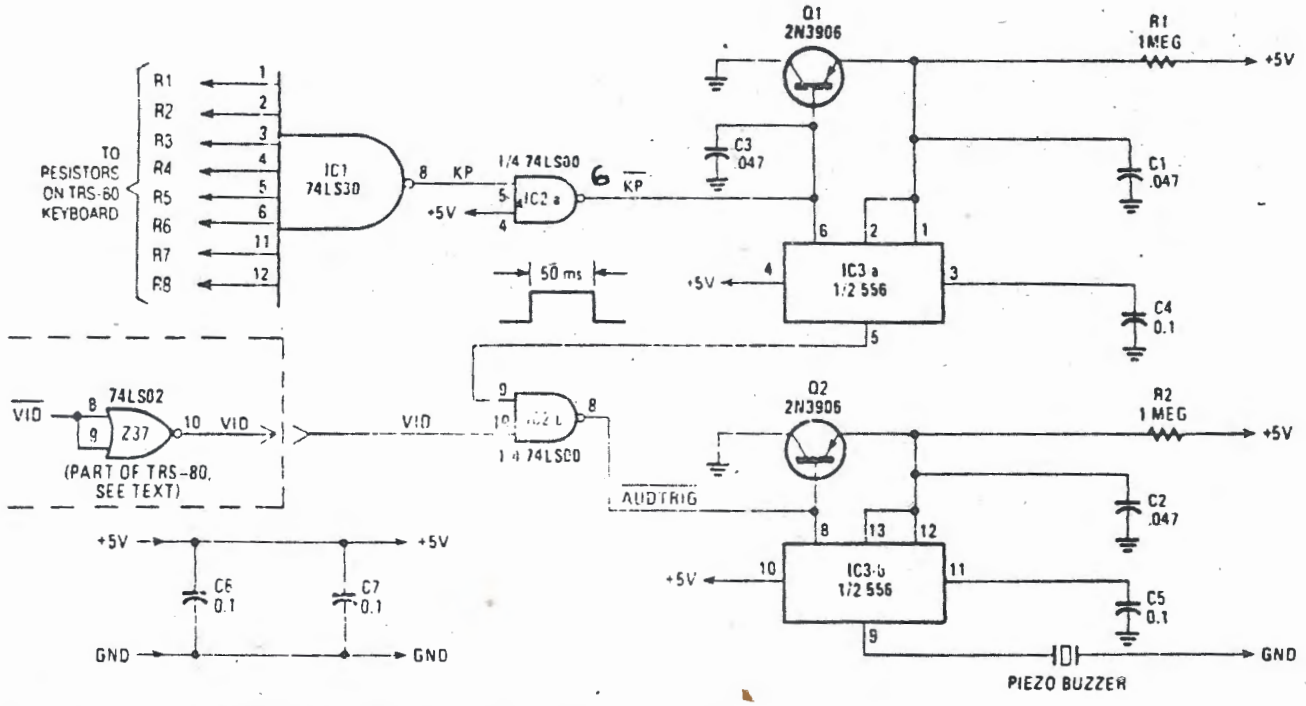
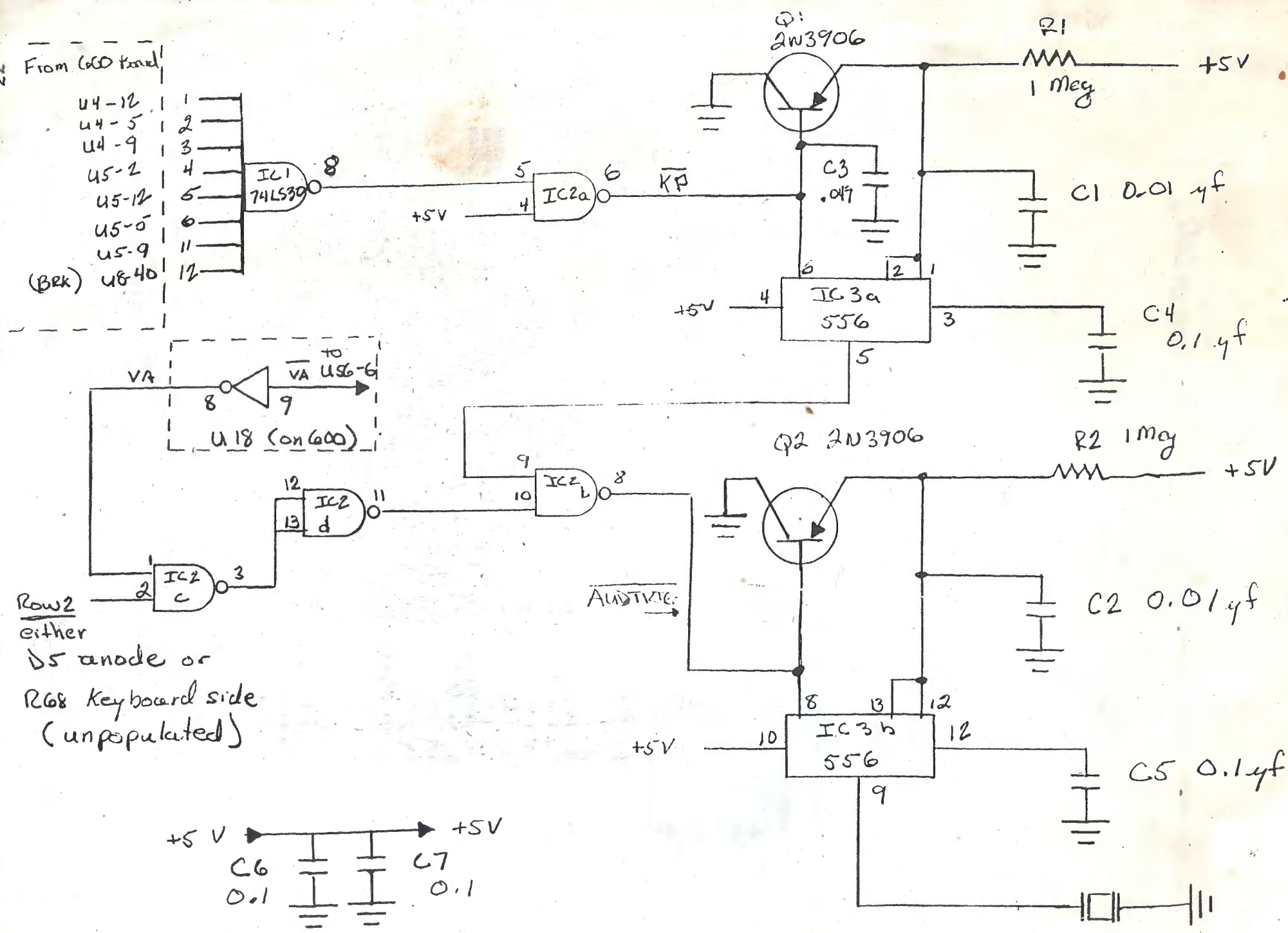


FIG. 1—SCHEMATIC DIAGRAM OF THE BEEPER showing connection to the TRS-80.

Parts List

RESISTORS
R1, R2—1 megohm, ¼ watt
CAPACITORS
C1, C2, C3—.047 μF, 50 VDC, disc
C4, C5, C6, C7—0.1 μF, 50 VDC, disc
SEMICONDUCTORS
IC1—74LS30 eight-input NAND gate
IC2—74LS00 quad NAND gate
IC3—556 dual timer
Q1, Q2—2N3906, PNP silicon
MISCELLANEOUS
Piezo Buzzer, Radio Shack No. 273-060 or equal
Printed circuit board
Foam mounting tape
Wire and ribbon cable
An etched and drilled circuit board is available for $6.00 postpaid from: Lynn R. Erickson, P.E., 1432 Rockhaven Place, Manteca, CA 95336. California residents please include 6% sales tax.

"ORIGINAL"



"Improved" Beeper