Processors Technology Access

Helios II Reviewed

WINZI: Hic, Opps, & Chase

Modified Bytesaver for Sol

The Floppy Discobolus

Newett Awl's Choo Choo Update for Sol
The feedback is starting to come in at a most gratifying rate, so we'll get on to the news of note after a brief commercial for our technical troubleshooting hotline -- call 415-829-2600 weekdays between 9:30 and noon or 1:30 to 4.

**Software availability -- good news time.** Are you ready? A good portion of Processor Technology Software is. Specifically, TREK-80, ALS-8, GAMEPAC, and BASIC/5 (all on the most-wanted list) are now available and in stock at your local dealer. No paper tape versions, though -- we're no longer producing any Processor Technology software on paper tape.

As for other software, you can currently look for SW#1 (Assembler) and MATHPAK in mid to late October; New 8080 FOCAL and 8K, and extended DISK BASIC on November 15. Incidentally, the two BASICS will have some capabilities we haven't advertised previously, such as complete matrix operations including matrix addition, subtraction, multiplication and inversion. Sure wish I'd had a Sol with 8K BASIC when I took linear algebra!

And look for some new software surprises turning up soon at your local dealer. We do appreciate your patience in waiting for software production and availability, and you won't be disappointed with the final product.

**Software cassettes.** Just a reminder that all our audio cassettes are recorded on two sides. Side 1 is in CUTS Standard (SOLOS/CUTER compatible, recorded at 1200 baud); Side 2 is Kansas City Standard at 300 baud, in case you don't have a Sol or CUTS audio cassette interface board.

**Hardware: Helios II is coming.** After many frustrating engineering and software development delays, deliveries have begun on the Processor Technology Helios II dual-drive floppy disk system. If you have one on order, either through a dealer or directly from us, rest assured that it's on its way. Unfortunately, we have so many orders we probably won't get through the entire backlog until late November or early December. If you want additional information, or you'd like to see a demonstration, check your local dealer, or give us a call.

We've been emphasizing the importance of our dealers all along, and they are your best first source. They're far more able to provide the service you deserve than we are from our distant factory, and they're always happy to hear from you. They're all authorized to perform service and repairs on all our products, and we're adding new dealers rapidly. Retail computer stores are multiplying like rabbits, in case you hadn't noticed.

---

*Lloyd's Second Law: Any program can be written in fewer commands.*

---

**ENTER THE GREAT SHOW-OFF-YOUR SOL CONTEST**

and maybe win a GPM/Sol with ALS-8/SIM-1/TXT-2 ROM set

All you have to do is tell us how you are using your Sol. The grand prize will be awarded for the most interesting and/or unusual application; runner-up gets the Software Technology MUSIC SYSTEM, an interesting
way for you and your computer to make music. And all of the more interesting entries have a chance to get published in future issues of ACCESS, with full credit to your resourcefulness and imagination.

Use the form on the back page to enter, using additional paper as needed. We'd like to have as much technical information as possible; i.e., what kind of hardware support was needed to get Sol to do your thing? And please be as specific as possible about when and where you're using it-business, science, industry? At home? What's it doing? Have you interfaced it to your lawn sprinklers, burglar alarm, Chevrolet? (We've heard of Sols used in the most provocative ways!)

We're holding this contest with three motives in mind. 1) To provide us with some feedback on what Sols are doing and how they're doing it, so we can direct our future efforts accordingly. 2) To give you, the Sol user, some insight into what other Sols are up to so you can pick up some nifty ideas for yours. 3) To provide a little excitement, fun and games for everyone.

The prizes will be awarded by the most impartial judges we can scout up to objectively determine the most original, unusual and imaginative application of a Sol. Programs and developmental work you may have done are valid entries, so get those pencils going on the entry form. Mail your entry to:

Sol Applications Contest
Processor Technology
7100 Johnson Industrial Drive
Pleasanton, CA 94566

Closing date is December 1, 1977.

---

**Table of Contents**

Reader Feedback .................................................................................................................................................. page 4

Sol User's Society gets into action with lots of plans and they're ready for input ........................................ page 6

Colgate Spinx opinionates re the software flexibility of Helios II................................................................. page 7

WINZI?! A collection of short programs, featuring the likes of Hic, Opps, and Chase ............................... page 8

Save your Sol's bytes with just a few little changes ............................................................................................ page 13

Some uppercase key software to save wear and tear on your patience whenever you restart or power up Sol .......................................................... page 14

How to hear your Sol keyboard, and shape up your touchkey typing ............................................................ page 15

Run/Stop, Part II, tells you more about monitoring the operation of your Sol ............................................. page 15

PLUS: A program for Home Accounting (p. 17), an 8080 Relocatable Assembler (p. 20), an ALS-8 to Sol Patch (p. 22), an Intel Tape Loader for Sol (p. 23), Newett Awl's Choo Choo Updated for Sol (p. 25), and Galaxy (p. 29).
A Letter From the Editor

"Any minor of order \( r \) in the adjugate determinant of a square matrix \( A \) is, on expansion, a certain polynomial in the elements of \( A \), fixed in form whether \( A \) is singular or not."

Determinants and Matrices

"There are no Jewish midgets."

Lenny Bruce

Well, how do you like our new format? Now you can keep our deathless prose (and occasionally valuable information) neatly in a 3-ring binder instead of stacked up on the floor someplace to hide the cigarette burns. (Oh, you liked having something to hide the cigarette burns?) It should prove a more useful reference format, and we'll keep working on making ACCESS responsive to your needs. May even get classy soon and go to two-color.

Notes from the show circuit: We had one display at NCC (Dallas) and another at WESCON (San Francisco). Seems like computer shows are cropping up everywhere you look, and unfortunately we just don't have time to keep up with them all. It's great to see so much interest developing though -- remember back to the dark ages when you were the only computer freak in town, all of a couple of years ago?

Meanwhile, instead of going to shows we're keeping our noses to the grindstone. The first Helios shipment is out and more are on the way. ALS-8 on CUTS cassette is now shipping, and 8K BASIC is set for mid-November. Ralph has more to say about new products, so take a look at his 1-to-1 column.

By the way, if you send us software contributions, try to give them to us on CUTS cassette. That way we can run out a listing, send it off to the printer, and it turns up in the next ACCESS in a neat standard format just the way you wrote it. If we have to have the listing typeset, there's just that much more chance of an error creeping in and turning your beautiful program into a debugger's nightmare. We are getting some nifty programs from you people -- take a look at Guy Campbell's home accounting system and Melvin Schehlren's modification to avoid erasing input lines, both in this issue.

We're always interested in all forms of communication with the outside world. If you are a member of a club, or publish a newsletter yourself, I'd really like to hear about it. We might be able to exchange membership or subscription lists. That way we'd all be getting more information, and know more about each other.

Well, bye now -- y'all come see us at our new home in Pleasanton, y'heah?

Aram Attarian II
Subscription Information

Access is published every six weeks. If you like what you see, we hope you'll send us $4.00 for a year's subscription so we can keep the info coming. Write to us at Processor Technology, 7100 Johnson Industrial Way, Pleasanton, CA 94566.

Have You Moved?

Please notify us of your change of address. Here's a handy form:

NAME _______________________________________
NEW ADDRESS ___________________________________
______________________________________________
______________________________________________

Dear Bob:

Thanks for the strokes. We'll put you on our mailing list right away, and we'll be glad to do the same for any other clubs who care to drop us a line.

Aram
Gentlemen:

As the satisfied user of two Sol systems I have encountered a problem when using the device as a terminal. The problem occurs because of the PCR routine in the VDM section of SOLOS. The attached assembly listing should be self explanatory as a successful solution. (See p. 20, this issue.)

Secondly, how come your company wasn't directly represented at Atlantic City this year? I know by talking to friends in the computer hobby field that you are having great difficulty delivering some of your new product line. Your equipment is of such high quality across the board most of us are willing to wait. Do you have something to hide or did you opt for Boston?

Yours truly,
Melvin E. Schehlein,
Ass't. Director of CMS

Dear Mel:

Thanks for a good idea: being able to check the last input certainly helps. Your program appears on page 20 of this issue. As for the shows, the cost of hitting all the ones cropping up on the East Coast especially is just prohibitive. We wouldn't have any money left to produce Sol's with. But we will make the New York City show in October. Right about now in fact.

Aram

Dear Aram:

I recently purchased, assembled and am running a SOL-20 under SOLOS. As a first trial exercise (having only 4K), I entered Newett Awl's Choo Choo Train. After much gnashing of teeth, wringing of hands and tracking of unstructured programming, I got it running. The trick was in the instruction in location 0156H; instead of an "IN 0" (DB00), SOLOS requires an "In FC" (DB FC). This is in addition to the necessary "C3 C9C1" in 0163H to return to SOLOS.

My purpose in writing this letter is not so much to relay the above software changes as to point out where I think ACCESS could play a large part in information dissemination. I received very little information with my SOL-20, regarding how to use the various devices with SOL (tape recorders, etc.). After scanning through the SOL manual and SOLOS listing, I found the skeleton references to the memory map and port addresses.

I could not locate any information of the complete hex-to-VDM character set, including all the obscure characters like <<headlessman graphic>> and <<triple horizontal lines>>. How about some examples of programs using the subroutine calls to SOLOS, particularly for animation on the VDM? In closing, I would like to compliment you on the publication of ACCESS. It is reassuring to have some further "feed-forward" from the vendor once the hardware has been sent. Keep up the good work!

Yours truly,
Warren L. Harkness
(In SOLOS UNUM)

Warren:

Thanks for the feedback, and we will keep supplementing the manual with helpful tips via ACCESS. Also, this issue has the Choo Choo listing modified to run on Sol.

Aram

To: Processor Technology Co. ACCESS Editor

Thanks for a fine publication and outstanding products! Thought you might include this modification to the Sol system in ACCESS.
I am presently stationed in Japan with the U.S. Air Force. The domestic power here is on the European standard, that is 50hz at 100 volts. Now the SOL-20 power supply doesn't seem to mind that at all, all bus voltages are right up there, but the "swim" effect on the display could give you a splitting headache in short order.

The answer to my problem was actually quite simple. I needed four more character rows during the blanked period of the display. To obtain the extra rows I changed the preset count of U62 during the high state of VDISP, the display blanked. This change was implemented by disconnecting pin 5 of U62 from the VDISP line and tying it to ground. This gives me a total of 16 displayed and 8 blanked character rows for a total of 24 rows with 312 scan lines, a close match for the 50hz operation.

I accomplished the MOD without cutting any P.C. foil or removing the main board from the Sol chassis. Parts needed are: one Molex Pin and a short length of flexible wire.

Remove IC 62 from its socket. Carefully scrape a small bare spot on the large ground bus running near the left of the IC socket; just remove the solder mask, don't cut the bus. Solder the Molex Pin to a one-inch length of wire and then the wire to the ground bus. Bend pin 5 of the 93L16 IC outward 45 degrees and insert the IC in the socket leaving pin 5 projecting outwards. Now slip the Molex Socket over the protruding IC pin .... That's it. And no permanent disfigurement of my precious Sol when I returned to the land of 60hz.

I hope that others will profit from this modification.

Ray D. Congdon
1956 COMM GP/OLC
APO San Francisco CA 96343

Ray:

Thanks much.

Aram

Dear Mr. Attarian

Enclosed you will find a check for $4.00 for my 1-year subscription to ACCESS. Also, I have a few questions and suggestions.

1. I am a firm supporter of your company. If there is any way I can contribute to its continued success, let me know. I am presently unemployed, so most of my time is spent exploring my Sol 20 system. Unfortunately, I paid for my system via a loan. In another month or so, I won't be able to pay off this loan. If I don't find some sort of income soon, I may have to sell my Sol. I don't want to have to do this!! My being located in the center of the eastern megalopolis should open some possibilities. If there is anyway possible, HELP!!

So far, I have written some original programs. One demonstrates the use of control characters and the escape sequences used by BASIC5. I have also written machine subroutines which can be used with BASIC5 to produce some special effects.

I also do some hardware design. I have completed design of a 16K static memory board using MOS TEC 4104's. I am working on a few other things as well.

If there is any way I can help, even with nothing in return, please let me know. As far as some sort of income, if you can't help, maybe one of the readers can.

Note: I will relocate.

2. It is said that escape sequences can be used to generate characters in inverse video. I have not been able to figure out how to do this. Could you please shed some light on this subject.

3. In regard to a letter from Joseph P. Chalala, Willow Street, Pa., in Vol. 1, #3, I agree completely with his suggestion for a notebook type publication for ease of filing.

Well, that's al I for now. I'm sure there will be more in the future. Remember Murphy's Law of Thermodynamics which states "Things get worse under pressure." So, take the time required to do, whatever, right (within reason, of course). If not, it will cost you more later.

A dedicated Sol user,
David F. Wrobel

P.S. The Bayshore Amateur Computer Group, of which I am president, consists of dedicated microcomputer hobbyists. Of which, only a few are Sol owners. We would like more. We are located in central N.J. Our address is: BACG P.O. Box 132, Holmdel, N.J. 07733.

David:

There's a good reason you haven't figured out how to generate characters in inverse video—it's not true that it can be done. Sorry. Hope the new format meets your filing needs.

And can anybody out there help David find work and save his Sol?

Aram

Dear Editor:

I am always reluctant to send one of my little masterpieces in for publication, but it seems like everyone else is a little shy also, and I know that there are a lot of Sol owners, like me, anxious to try out some of its unique features.

For whatever it's worth, I'm sending a short program that makes use of the File commands in Basic 5. It's not meant to be a finished product, but it does demonstrate one way to use commands. In fact, I hope someone will pick up the ball and make a better program out of it.
The purpose of the program is to allow the user to slip last month's data tape into one file, pay the bills, update the household accounts and store the updated accounts on the other file.

I have included a sample program for setting up the original data tape, (similar to the one in the Basic 5 manual), the actual program I use for working the accounts, and a sample run.

Sincerely yours
Guy W. Campbell
5815 Buckley Drive
Jacksonville, FL 32210

Guy:

Thanks. This is the kind of creative input we really appreciate. Readers: Guy's program appears on p.17 of this issue.

Aram

We've Moved Again!

For the same reason as last time -- we ran out of space and outgrew the facilities again. We've now forsaken Emeryville for sunny Pleasanton, CA, just over the hill. If you're curious about the history of Pleasanton, or just curious, I refer you to a book entitled "Mammy Pleasant," by Helen Holdreage.

The new address is
Processor Technology Corp.
7100 Johnson Industrial Way
Pleasanton, CA 94566
Phone: 415-829-2600

Join the Sol Users' Society

The Sol Users' Society got under way Sunday, July 31, when about 30-40 Sol users met for the first organizational meeting. The Society is open to everyone who has a Sol or a Sol-type compatible system, so they're hoping to see even more of you at the next meetings.

This first time out a steering committee was elected, and goals were set for the Society. These goals are:

1. To facilitate communication between Sol owners.
2. To provide feedback from Sol owners to PTC.
3. To provide a mechanism for exchanging Sol software.
4. To encourage development of Sol-compatible products by other manufacturers.
Some time was spent just getting to know one another and talking about various projects the club can tackle. Seems that most of the stuff users have to offer is software, but they're also interested in reviewing any hardware submitted to the Society, be it prototype or production. They can't supply certification, though.

One project definitely under way is a Sol Users' Society newsletter. Contributions and comments herewith solicited.

A tidbit that emerged from the first meeting: TDK Auda C-60 cassette tape performs best in a bitchopping test.

**Schedule of meetings.** The group is set to meet on Sundays Oct. 16, Nov. 20, and Dec. 18, at Varian Physics Lab, 2nd Floor, Stanford CA. Come meet the new steering committee: Bill Burns, Dave Fylstra, Ron Findlay, Ben Milander, Bill Holding, Stan Sokolow, David Fox.

For more information, please write to:

Bill Burns  
4190 Maybell Way  
Palo Alto, CA 94306  
(no phone calls, please . . .)

**Review:**

**Software Capabilities of the Helios II Disk System**

The Helios II system has several capabilities I haven't seen in other disk systems on the market, and one particularly notable advantage for use with the Processor Technology Disk Operating System: you can write I/O routines for the Helios which permit the use of any I/O controller in conjunction with PTDOS, including the Cromemco D to A board and just about any homebrew board you've already built. (Maybe not some that perform DMA or make use of the I/O ports as control ports by the disk controller.)

The reason for this flexibility is that Helios treats all files as data files, including the device files used for I/O routines. These differ from regular files in that data read from or to them will come or go directly to the devices controlled by the I/O routine. With Helios, the only thing you have to worry about is to make sure you follow the guidelines in the PTDOS user's manual when you write your I/O routine.

Software support is another big plus for Helios. It offers a disk assembler, two editors (one ALS-8 type, one Nova-type), library functions, a debugger, language systems, procedures (PROCS), and full interface to PTDOS on command or assembly level.

The disk assembler allows you to generate object and listing files from a source file. You have the options of specifying if the input file is ALS-8 type, if it has line numbers, if it has form control, and a few other things.

The ALS-8 type editor is especially useful on systems with a lot of memory because you can work with text, as in the ALS-8. You have the options of scrolling forward and backward through the text, deleting characters, searching character strings, moving blocks of code, replacing string patterns with others as found. The limitation to this editor is that it requires the VDM-1.

The Nova-type editor can be run on almost any terminal, since all I/O is run through the system console routines. It will yank pages into the edit buffer, change data in the page, and write it out. If offers many of the same functions as the ALS-8 type.

Library functions allow you to assemble several source files which make up one logical program. It's done through the use of a copy verb included in the PTDOS assembler. You could expand the use of this verb to build up a library of source files which perform common functions, then concatenate these files into an object file through the assembler.

The debugger serves a function similar to that of the simulator in the ALS-8: you can run object code in a controlled environment. However, the debugger runs real time instead of the simulator's interpretive mode. You can use it to set numerous break points; examine memory in hex, character, or instruction format, alter memory or output drivers; and do several other useful debugging tasks.
The broad spectrum of language systems available includes a DISK BASIC and DISK FOCAL; FORTRAN is rumored to be among those upcoming. Language support is definitely one of the big advantages of the Helios system.

The command interpreter gives access from the console to many of PTDOS's numerous entry points. A partial list of the commands available: SPACE, OPEN, CLOSE, KILL, RANDOM, SEEK, RENAME, REATR, RETYPE, CREATE, READ, WRITE.

With PTDOS, you can also enter procedures in ALS-8 type file formats as a series of commands which may include optional statements.

The PROCs itself is simply a list of commands which you can enter and allow to execute consecutively. Very useful for setting up, say, a 3-hour listing to print out while you get some sleep.

All in all, the Helios II disk memory system has proved well worth the time and price from this user's point of view.

Colgate Spinx

_Cynic: One who is enough to make anyone a pessimist._
WINZI

WINZI is a collection of programs that were originally written last summer for VDM1 and published in D.D.J. They have been rewritten for a Sol with a SOLOS personality module and addressed to use the 1K of RAM available on the Sol P.C. board.

The programs are:

H ic -- a random walk
O pps -- draw a picture without returning
C hase -- make the turtle catch the bouncing bug
L ife - the 'game' of LIFE

The speed of Hic, Opps, Chase and Life are under control of the SET command (see SOLOS manual). Before EXectuting C900 SET S=80. If you fail to set the speed first, the speed is so fast that you won't see it happen.

Type 'H' for Hic, 'O' for Opps, or 'C' for Chase

In Hic the beastie should be moving around leaving asterisks. If it is not moving, the random number generator might not be working. It is important not to zero memory before loading this program. In particular the Data Storage area SH should be nonzero. When you get tired of watching it, hit CR and return to the executive.

In Opps, you control the direction the beastie moves. The directions are as shown, upper right.

Type the number corresponding to the desired direction. The beastie will proceed in that direction until another direction is given. '0' will stop the beastie. '5' will cause a wipe out. 'k'. If the beastie ever returns to a location where it's been a 'k' will appear and the program will return to the executive, hence the name: Opps. If you wish to return to the executive at any other time type CR.

Life requires that there is an initial population of asterisks on the screen. Place them there by either Hic or Opps. Then enter Life by typing LI from the executive. The CR will return you to the executive. 1K of RAM 0-3FF Hex is needed by Life as a scratchpad.

```
C900 CD C5 CA 0090 CALL CS Clear the screen
C903 CD FE C9 0100 RET CALL KB The executive routine
C906 FE 43 0120 CPI 'C' branches to chase on C,
C908 CA 1D C9 0130 JZ CH
C90B FE 43 0140 CPI 'H' Hic on H,
C90D CA 5D C9 0150 JZ HI
C910 FE 4F 0160 CPI 'O' Opps on O, and
C912 CA 76 C9 0170 JZ OP
C915 FE 4C 0180 CPI 'L' Life on L.
C917 CA 09 CA 0190 JZ LI
C91A C3 03 C9 0220 JMP RET
C91D CD C5 CA 1000 CH CALL CS Chase... Clear the screen.
C920 CD 84 CA 1005 CALL INIT Place the bug
C923 3A CF CA 1010 LDA FO+1 at a random location
C926 77 1020 MOV M,A on the screen.
C927 EB 1030 XCHG
C928 CD 84 CA 1040 CALL INIT Place the turtle
C92B 3A CE CA 1050 LDA FO at a random location
```
C92E 77 1060 MOV M,A on the screen.
C92F CD F1 C9 1070 C1 CALL DL Wait a while.
C932 CD FE C9 1075 CALL KB Get input.
C935 36 20 1080 MVI M,20H Put a space where you are,
C937 CD A3 C9 1090 CALL MV then move.
C93A 3A CF CA 1100 LDA F0+1 If the bug is there,
C93D BE 1110 CMP M you've got him.
C93E CA 9E C9 1120 JZ HLT
C941 3A CE CA 1130 LDA F0 Place turtle at new screen
C944 77 1140 MOV M,A location.
C945 EB 1150 XCHG
C946 CD A0 CA 1180 CALL RND The bug moves at random
C949 36 20 1185 MVI M,20H Put a space where it was,
C94B CD 3C C9 1190 CALL MV then move.
C94E 3A CE CA 1200 LDA F0 If turtle is there,
C951 BE 1210 CMP M the bugs been had.
C952 CA 9E C9 1220 J2 HLT
C955 3A CF CA 1230 LDA F0 Place bug at new screen
C958 77 1240 MOV M,A location.
C959 EB 1250 XCHG
C95A C3 2F C9 1260 JMP C1 Do it all again
C95D CD C5 CA 1300 HI CALL CS HIC...Clear the screen
C960 21 20 CE 1305 LXI H,0CE20H Place the turtle at the
C963 36 07 1310 H1 MVI M,7 center of the screen.
C966 CD FE C9 1315 CALL KB Should I return to the executive?
C968 CD F1 C9 1320 CALL DL Wait awhile
C96B CD A0 CA 1330 CALL RND Move the turtle at random
C96E 36 2A 1340 MVI M,2AH
C970 CD A3 C9 1350 CALL MV leaving asterisk behind
C973 C3 63 C9 1370 JMP H1 Do it again.
C976 CD C5 CA 1600 OP CALL CS Opps Clear the screen
C979 06 00 1602 MVI B,0 Place turtle at the center
C97B 21 20 CE 1605 LXI H,0CE20H of the screen.
C97E 06 07 1610 OF MOV A,M If you have been
C97F FE 2A 1620 CPI 2AH here before,
C981 CA 9E C9 1630 J2 HLT halt.
C984 36 07 1640 MVI M,7 Put the turtle on the screen
C986 CD F1 C9 1650 CALL DL wait awhile.
C989 CD FE C9 1655 OF CALL KB Get input.
C98C CA 90 C9 1658 J2 O4 If none, continue
C98F 47 1660 MVI B,A
C990 78 1662 O4 MOV A,B
C991 E6 0F 1664 ANI 0FH If it is a '0', don't move.
C993 CA 89 C9 1666 J2 O2
C996 36 2A 1670 MVI M,2AH place asterisk in old location
C998 CD A3 C9 1680 CALL MV then move.
C99B C3 7E C9 1690 JMP O1 Do it again
C99E 36 04 1700 HLT MVI M,4 Halt--Place '!' on the screen.
C9A0 C3 03 C9 1710 JMP RET Return to the executive
C9A3 E6 0F 1800 MOV A,B
C9A5 FE 06 1850 CPI 6
C9A7 C2 AF C9 1860 JNZ M1 This routine uses the
C9AA 3E 01 1870 MVI A,1 curser move routines
C9AC C3 C0 C9 1880 JMP M3 in SOLOS (PUP, PLEFT, PDOWN,
C9AF FE 01 1890 M1 CPI 1 and PRIT) to move the
C9B1 C2 B9 C9 1900 JNZ M2 contents of the screen
C9B4 3E 06 1910 MVI A,6 location pointed to
C9B6 C3 C0 C9 1920 JMP M3 by the address contained
C9B9 FE 07 1930 M2 CPI 7 in the H & L registers.
C9BB C2 C0 C9 1940 JNZ M3
C9BE 3E 0C 1950 MVI A,0CH The right 4 bits of the
C9C0 4F 2000 M3 MOV C,A accumulator are used
C9C1 CD E0 C9 2010 CALL SVDA to determine the direction,
C9C4 3E 08 2020 MVI A,8 or combination of
C9C6 A1 2030 ANA C directions, of the move.
C9C7 C4 04 C1 2040 CNZ 0C104H
C9CA 3E 01 2050 MVI A,1
The ASCII code for all of the digits except 6, 1, and 7 allows a compass rose that agrees with the 10 key numeric pad. 6, 1 and 7 are changed to agree.

This routine assumes that the contents of the H,L register pair is a screen address.

It then stores the line number in the data storage location used by SOLOS called LINE. Similarly the character position is stored in NCHAR for use by PUP, PLEFT, PDOWN, and PRIT.

Delay...Save address.

Get data byte from SET command: SPEED

Delay...Save address.

Get data byte from SET command: SPEED

Use it for a counter.

Use it for a counter.

This routine assumes that

Use it for a counter.

Use it for a counter.

Use it for a counter.

Restore address

This routine assumes that

Restore address

Clear and return.

Clear and return.

Clear and return.

Clear and return.

Clear and return.

Clear and return.

Clear and return.

Clear and return.

Clear and return.

Clear and return.

Clear and return.

Clear and return.

Clear and return.

Clear and return.

Clear and return.

Clear and return.

Clear and return.

Clear and return.

Clear and return.

Clear and return.

Clear and return.

Clear and return.

Clear and return.

Clear and return.

Clear and return.
Copy the next generation as stored in the 1K of scratch pad RAM. 

Wait awhile. Should I return to the exec.? 

Follow the rules of LIFE to determine the next generation for this location.

Counter If the neighbor is an asterisk count it.

Initialize. number in the data storage of SOLOS called LINE and the position of the character in NCHAR.

Random number generator from Peoples Computer

Random for it to work properly the 4 data storage locations should not be zero.
Bytesaver Modification for Sol

If you want to use a Chromemco Bytesaver in the Sol, you'll need to make the following modification of the Bytesaver. Data will then be gated onto the Bus only when PDBIN is high or active, necessary in the Sol because the Data IN and Data OUT busses are connected together.

First cut the trace connecting pin 11 of IC 15 (7432) to pin 15 of IC 16 (74367). Now make these connections with small gauge insulated wire:

1. Connect pins 11 and 10 of IC 15.
2. Connect pin 8 of IC 15 to pin 15 of IC 16.
3. Connect pin 8 of IC 11 to pin 9 of IC 15.
4. Connect S-100 Bus pin 78 (PDBIN) to IC 11 pin 9. Pin 78 is the 23rd from the left on the solder side of the board.

. . . AND A BYTESAVER PROGRAMMING ROUTINE

This short routine will program the contents of any 1K block of memory into a 2708 EPROM installed in socket 1 of Bytesaver. The Bytesaver should be addressed at 6000H.

A15-L, A14-H, A13-H

The routine is used as a custom command with the Solos/Cuter operating system. Enter the program at C900H, or reassemble it elsewhere if you wish. Then create a custom command by typing:

CU BURN C900 (CR)

NOTE: CR means "strike the return key;" do not type the letters as part of the command.

If the program has been reassembled at an arbitrary address of NNNN, type: CU BURN NNNN (CR) Now to use the BURN custom command, type:

BURN AAAA (CR)

AAAA being the starting address of the 1 K block you wish to program into the 2708. The programming operation takes about 5 minutes, which is in accordance with the published programming instructions for the 2708. When the programming is complete, the routine will return control to Solos/Cutter and a prompt will reappear on the screen.
**BYTESAVER ROUTINE**

C33A       0000 SCONV EQU    0C33AH  GETS PARAMETERS

C900 CD 3A C3 0000 BURN CALL SCONV SOURCE ADRS TO HL
C903 7D   0000 MOV A,L GET LO ADRS BYTE
C904 B7   0000 ORA A
C905 C2 04 CO 0000 JNZ 0C004H IT MUST BE

C908 22 28 C9 0000 SHLD SAD KEEP SOURCE ADRS
C90B 01 00 00 0000 LXI B,0 PASS COUNT=0

C911 2A 28 C9 0000 BLOOP LXI D,6400H PROM ADRS
C914 7E 0000 PLOOP MOV A,M GET SOURCE DATA
C915 12 0000 STAX D ZAP THE PROM
C916 23 0000 INX H BUMP SOURCE
C917 13 0000 INX D & PROM ADRS
C918 7A 0000 MOV A,D CHECK HI ADRS
C919 FE 68 0000 CPI 68H PASS COMPLETE?
C91B C2 14 C9 0000 JNZ PLOOP NOT YET
C91E 03 0000 INX B BUMP PASS COUNT
C91F 78 0000 MOV A,B
C920 FE 04 0000 CPI 4 IK PASSES?
C922 C2 0E C9 0000 JNZ BLOOP NOT YET

C925 C3 04 CO 0000 JMP 0C004H ALL DONE

**RAM AREA**

A Sol Keyboard Fix:

So You Won't Have to Hit the Upper Case Key Each Time You Restart

*EDITOR'S NOTE: Our thanks to Jay Bell for contributing this suggestion. There are a couple of minor differences between the procedure he describes and our PTC standard modification for keyboard upper case initialization, so we're printing our version along with his.*

So after days of constructing your Sol, you're finally ready to input the first command, hit the carriage return, and check the screen. WHAAT??! All you get is some question mark nonsense. So you check the software manual again. Sure enough, it wants uppercase. So, you put the keyboard into alpha-shift by pressing the upper case key.
Later you notice your program isn't doing what you expected, so naturally you restart the old four-phase wonder by simultaneously pressing the upper case and repeat keys. More question marks-the restart left you in lower case mode.

By now you've realized this is going to happen every time. There are three solutions: 1) Change the software to accept both upper and lower case commands. 2) Change the keyboard to come up in uppercase mode. 3) Hit the upper case key every time you restart.

Number 3 had already worn me out. I personally prefer to change software, even though I'm a hardware freak. But I figured the chances of Processor Technology changing software at this late date were sub-minimal.

Out with the keyboard schematics. The fix looked simple enough: just CLEAR the upper case flipflop rather than PRESETTING it. The keyboard gets preset when power is first applied through an RC circuit that is initially low and slowly comes up to +5 volts. Since the signal coming off the keyboard to restart the 8080 is driven by an open collector inverter, it could also be connected to the power-up RC circuit. Then whenever you reset the processor, you also reset the keyboard to its initial power-up state. To make that power-up state turn the upper case flipflop on, you cut the land leading to pin 4 of U15, and the land leading to pin 1 of U15. Then connect the trace that used to lead to pin 4 to pin 1 instead. Similarly, connect the pull-up resistor that was tied to pin 1 to pin 4 instead. Then connect pin 8 of U24 to pin 1 of U15. This last connection ties the restart signal to the clear input of U15 (as well as to the rest of the chips that are initialized at power-up).

There is only one remaining problem for the purists. The flipflop that sets the machine in the local mode will come up in an undetermined state, since its preset pin is tied to pin 4 of U15. Now that you've cut the land to pin 4 and pulled it high, the local flipflop is not being properly reset. Unfortunately, you have to remove U15 in order to cut the land between pin 4 of U15 and pin 10 of U15, then connect pin 1 of U15 to pin 10 of U15.

Now you should be able to interact with your Sol the instant you power up or restart, without the bother of hitting the upper case key first.

THE PTC MODIFICATION:

1. Cut trace located between U15 pin 4 and plate through 1/8 inches below pin on the component side.
2. Remove R31, 1.5K 1/4 watt Carbon Film, and save for later use.
3. On the Solder Side of the board:
   a. Insert one end of R31 in plate through adjacent to U15 pin 14 and solder.
   b. Bend the other lead of R31 to pin 4 of U15 and solder.
   c. Add a 5/8 inch jumper, stripped 1/8 inch from each end, to the plate through located just below U15 pin 4.
   d. Insert the other end of the jumper through plate through located just above U24 pin 12.

A Keyclick (Audible) Circuit for Sol

Silence may be golden, but there is an advantage to making your Sol keyboard sound like a typewriter. If you're a good fast touch typist entering data from a printed source, it's easier to listen for missed keys than to glance up at the screen all the time. Thanks to Jack Kinney for this audible circuit design; he says that the sound can be altered to suit individual tastes by varying R1 burst length and R3 for burst frequency. The circuit operates as follows:

The first section of the dual timer is connected in the monostable mode, and the keyboard strobe triggers a positive-going pulse approximately four milliseconds long. This pulse is connected to the reset of the second section of the timer, which is operating in the astable mode, and is set for an output frequency of approximately 1.5 Khz, gating it "on" for a four-millisecond burst. The output transistor inverts the signal to prevent current draw in the "off" condition. The collector resistor is set for the desired loudness.
Kinney is also checking out a more complex circuit (three more IC’s) which will decode the “BEL” code and produce a beep. One of the computers on the network signals for attention by transmitting the “BEL,” and this will provide an audible monitor. We’ll print this circuit in the next issue of ACCESS.

Katchum’s Correction Corollaries: (a) In debugging any type of program, no corrections can be made correctly after 1600 hours Friday. (b) The corrections will be self-evident at 900 hours Monday. (c) When in doubt divide by (2,0).

Goren’s Law of Graphing: First draw the curves, then plot the data.

Run/Stop Circuits: Part II

In ACCESS #3, I described a Run/Stop circuit for user control of Sol’s X-Ready line. Now here’s the circuit that will let you monitor the operation of your S-100 system by connecting LED’s to the buss lines. Each Light Emitting Diode is driven by 1/6 of a hex inverter package (74LS04), current limited by a 470-ohm 1/4-watt resistor for each.

To monitor the operation of your system, compare the addresses and data displayed on the LED’s as you single step, with the program listing. Most malfunctions can then be seen and corrected with very little effort. A couple of examples:

**Quick test for data and address lines.** You can discover major failures in these by toggling the reset line while the Run/Stop circuit is enabled in the Stop position. All the LED’s should light when the reset line is enabled. All except M1, PDBIN, and PWAIT should darken when the line is disabled. Any LED’s that don’t respond as indicated reveal a malfunction in the corresponding lines and should be checked with a meter or scope.

**Testing the Input/Output lines.** Single step until an input or output instruction is executing. When the SINP or SOUT LED is lit, you can stop stepping, and start following the logic signals in the I/O section, with your troubleshooting equipment.

Next issue, this series will continue with advice on implementing traps on the front panel.

Steve Wong
Bug Squad

Changes in Assembly Instructions for Sol:

The bug: You'll get an incorrect test result at Step 38 in the instructions; instead of the display shown in Figure 3-9, a display of random characters comes up.

The squasher: Install U93-74LS175 and U107-74LS367 at Step 35.

The bug: Your Sol doesn't work at Step 59.

The squasher: After you do Step 28 of the assembly, perform step 73 before proceeding to Step 29.

The bug: The waveforms are incorrectly shown in Figure 3-2 on page 3-15 of your Sol manual.

The squasher: Turn your manual upside down—the waveform shown for Pin 5-U104 is inverted. Same for Pin 7-U104.

A Program for a Home Accounting System

Contributed by Guy Campbell

See the Letters to the Editor for Guy's comments on his program. We're delighted to get this kind of input from our readers and pass it on for all Sol users.

PROGRAM FOR HOME ACCOUNTING SYSTEM .

THIS PROGRAM WILL RECORD PAYMENTS, UPDATE BALANCE AND PROVIDE ACCUMULATED INTEREST FOR TAX PURPOSES.

NEW ACCOUNTS CAN BE ADDED ONLY BY CHANGING THE PROGRAM.

PUT OLD DATA TAPE ON FILE #2 AND SET FOR PLAY.

PUT NEW TAPE ON FILE #1 AND SET FOR RECORD.

PRESS ANY NUMBER & RETURN TO CONTINUE. 0

---

YOUR FRIENDLY FINANCE CO.
3958 TUFFLUK STREET
CHICAGO, ILL 60683

---

ACCOUNT NUMBER - 12345A

---

BALANCE=$ 51.60                             ANNUAL INTEREST RATE= 21%
TOTAL PRINC. PAID=$ 108.00                   TOTAL INT. PAID=$ 6.39

---

YOUR LAST PAYMENT WAS MADE ON 82877 FOR $ 36.00

---

ENTER CHANGES TO ADJUST BALANCE. $ 25.00
ENTER CHANGES TO ADJUST BALANCE. $ 0
YOUR PRESENT BALANCE IS $ 76.60

---

ENTER PAYMENT TO THIS ACCT. $36.00

---

ENTER TODAYS DATE 90277
ACCOUNT COMPLETE
PRESS ANY NUMBER TO GET NEXT ACCOUNT. 0
***************************************************************
NEXT ACCOUNT COMES UP - WILL CONTINUE IN THIS FORMAT.
LIST
5    SET S=05
10   REM THE HOME ACCOUNTING PROGRAM
20   REM CREATED BY G. W. CAMPBELL - 1977
30   REM ORIGINAL DATA BANK (TAPE) PREPARED WITH
40   REM SEPARATE PROGRAM.
50   PRINT "PROGRAM FOR HOME ACCOUNTING SYSTEM."
60   PRINT
70   PRINT "THIS PROGRAM WILL RECORD PAYMENTS, UPDATE"
80   PRINT "BALANCE AND PROVIDE ACCUMULATED INTEREST"
90   PRINT "FOR TAX PURPOSES."
100  PRINT
110  PRINT "NEW ACCOUNTS CAN BE ADDED ONLY BY CHANGING"
120  PRINT "THE PROGRAM."
130  PRINT
140  FOR I=1 TO 1200:NEXT
150  PRINT "PUT OLD DATA TAPE ON FILE #2 AND SET FOR PLAY."
160  PRINT
180  PRINT "PUT NEW TAPE ON FILE #1 AND SET FOR RECORD."
190  PRINT
200  INPUT "PRESS ANY NUMBER & RETURN TO CONTINUE."
210  PRINT
220  GOSUB 770
225  FILE #1
230  FILE #2
240  READ #2,A,B,C,D,E,F,G: PRINT "END OF FILE";: GOTO 700
250  IF A=1 THEN GOSUB 1000
260  IF A=2 THEN GOSUB 1070
270  IF A=3 THEN GOSUB 1140
400  GOSUB 770
410  PRINT "BALANCE=$";%Z2%;C,
420  PRINT TAB (30);"ANNUAL INTEREST RATE=";%%;B;"%";%Z2%,
430  PRINT "TOTAL PRINC. PAID=$ ";F,
440  PRINT TAB(30);"TOTAL INT. PAID=$";G
450  GOSUB 770
451  PRINT "YOUR LAST PAYMENT WAS MADE ON ";%%;D,
452  PRINT "FOR $";Z2%;E
453  GOSUB 770
460  INPUT "ENTER CHANGES TO ADJUST BALANCE. ";C1
470  IF C1=0 THEN 500
480  LET C=C+C1
490  GOTO 460
500  PRINT "YOUR PRESENT BALANCE IS $";C
510  PRINT
520  INPUT "ENTER PAYMENT TO-THIS ACCT. ";F1
521  LET T1=T1+F1
530  IF F1=0 THEN 583
531  PRINT
532  INPUT "ENTER TODAYS DATE "D1
533  LET D=D1
550  LET G1=( (B/100) *C)/12
560  LET G=G+G1
570  LET C=C-(F1-G1)
580  LET E=F1
581 PRINT "PRINCIPLE PAID=$";(E-G1),
582 PRINT TAB(30);"INTEREST PAID=$";G1
583 PRINT
590 PRINT "YOUR NEW BALANCE IS $";C
595 LET F=F+(E-G1)
600 PRINT
610 PRINT "TOTAL PRINCIPLE PAID TO DATE=$";F
620 PRINT "TOTAL INTEREST PAID TO DATE=$";G
630 PRINT
640 PRINT "ACCOUNT COMPLETE"
650 INPUT "PRESS ANY NUMBER TO GET NEXT ACCOUNT. "Y
670 PRINT #1,A,B,C,D,E,F,G
680 PRINT
681 PRINT "****************************************************
690 TOTO 240
700 CLOSE #2
710 CLOSE #1
720 PRINT
730 PRINT "TRANSACTIONS COMPLETE"
740 PRINT
741 PRINT "YOUR PAYMENTS TOTALED $";T1;" THIS MONTH."
750 PRINT
751 PRINT "GOODBYE, SEE YOU NEXT MONTH."
752 SET S=0
760 END
770 PRINT "----------------------------------------------------
780 RETURN

*****THIS IS WHERE YOU PUT THE ACCOUNTS*****

1000 PRINT "YOUR FRIENDLY FINANCE C0."
1010 PRINT "3958 TUFFLUK STREET"
1020 PRINT "CHICAGO, ILL 60603"
1030 GOSUB 770
1040 PRINT "ACCOUNT NUMBER - 12345A"
1050 GOSUB 770
1060 RETURN
1070****CONTINUE TO PUT IN ACCOUNTS IN THE SAME FORMAT.

THIS IS A SAMPLE PROGRAM THAT CAN BE USED TO ESTABLISH THE ORIGINAL DATA BASE TAPE.

LIST
10 FILE #2
20 INPUT "ACCOUNT IDENTIFICATION NO. ?"A
30 IF A=0 THEN 110
35 PRINT
40 INPUT "ANNUAL INTEREST RATE (WHOLE NUMBERS) ?"B
45 PRINT
50 INPUT "BALANCE ?"C
55 PRINT
60 INPUT "DATE AND PAYMENT (LAST PMT MADE) ?"D,E
65 PRINT
70 INPUT "TOTAL PRINCIPLE PAID ?"F
75 PRINT
80 INPUT "TOTAL INTEREST PAID TO DATE ?"G
85 PRINT
90 PRINT #2,A,B,C,D,E,F,G
100 GOTO 20
110 CLOSE #2
120 END

8080 Relocatable Assembler
THE PURPOSE OF THIS PROGRAM IS TO CONFIGURE THE SOL TERMINAL
COMPUTER AS A STANDARD VIDEO TERMINAL TO ACCEPT THE HALF-DUPLEX
CR, LF RESPONSE FROM COMMUNICATIONS AFTER HAVING SENT A CR
WITHOUT ERASING THE LAST INPUT LINE. THIS HAPPENS BECAUSE
THE SOLOS MONITOR CLEARS THE LINE FROM ITS PRESENT CHARACTER
POSITION TO THE END OF THAT LINE. DURING COMMUNICATIONS
THE KEYBOARD SENDS OUT A CR, WHICH IS FEED BACK INTO THE SERIAL
INPUT PORT, AND THEN SENT TO THE VDM DRIVER TO BE PROCESSED.
THEN THE COMPUTER SENDS OUT A CR LF WHICH GOES TO THE VDM DRIVER
CLEARING THE LAST INPUT LINE (BECAUSE THE RESPONSE CR WAS IN COLUMN
1) THUS DENYING THE USER THE ABILITY TO CHECK THE ACCURACY OF
HIS OR HER LAST INPUT.

C900                        ORG     0C900H
C900              ;  THIS PROGRAM MODIFICATION WILL SIMPLY TEST THE CHARACTER POSITION
C900              ;   WHENEVER A CR IS RECEIVED. IF WE ARE IN COLUMN 1 THE CR IS
C900              ;   IGNORED, IF WE ARE NOT IN COLUMN 1 IT IS PROCESSED NORMALLY.
C900              ;   THE ONLY CODE THAT IS DIFFERENT FROM THE STANDARD SOLOS TERM
C900              ;   ROUTINE CAN BE FOUND IN LINES 63 TO 70 TO TEST THE CHARACTER
C900              ;   POSITION.
C900              ;   THE PROGRAM HAS BEEN ASSEMBLED BEGINNING AT ADDRESS C900
C900            ;   SO IT CAN EASILY BE STORED IN SOL SYSTEM RAM.
C900            ;      +++++ TERMINAL UPDATE ROUTINE +++++
C900 CD 10 C3     TERMU:    CALL   PSCAN    ;FIND IF INPUT PARAMETER IS PRESET
C903 32 06 C8               STA    IPORT    ;SINP WILL USE THIS DRIVER (DEFAULT IS 1)
C906 CD 10 C3               CALL   PSCAN    ;NOW FOR THE OUTPUT DRIVER
C909 32 07 C8               STA    OPORT
C90C                                        ;
C90C CD 2E C0     TERM1U:   CALL   KSTAT    ;IS THERE A CHAR WAITING AT THE KEYBOARD
C90F CA 24 C9               JZ     TINU     ;NO, THEN WHAT ABOUT THE SERIAL INPUT PORT
C912 47                     MOV    B,A      ;THE KEYBOARD HAD ONE, PUT IT IN B
C913                                        ;
C913 FE 80                  CPI    MODE     ;IS IT A COMMAND MODE
C915 CA C0 C1               JZ     COMN1    ;YES--GET OUT OF TERMU>>AND 00 TO SYSTEM
MONT
C918 DA 21 C9               JC     TOUTU    ;NON-CURSOR KEY--SEND TO TERM PORT
C91B CD 54 C0               CALL   VDMOT    ;TO THE VDM IT IS A CURSOR CONTROL
C91E C3 24 C9               JMP    TINU    ;IS THERE A CHAR AT THE INPUT PORT
C921            ;
C921 CD 19 C0     TOUTU:    CALL   SOUT    ;OUTPUT IT TO THE SERIAL PORT
C924 CD 1F C0     TINU:    CALL   SINO    ;IS A CHAR WAITING AT THE INPUT PORT
C927 CA 0C C9               JZ     TINUU    ;HOW ABOUT THE KEYBOARD?
C92A E6 7F                     ANI    07FH     ;GET RID OF THAT HIGH ORDER PARITY BIT.
C92C CA 0C C9               JZ     TERM1U    ;A NULL IS PROCESSED BY DOING NOTHING
C92F 47                     MOV    B,A      ;IT IS OUTPUT FROM 'B'
C930 FE 1B                  CPI    01BH     ;IS IT A CONTROL CHAR TO BE IGNORED
C932 D2 5E C9               JNC     TERM2U    ;NO IT IS >1B SEND TO THE VDM
C935            ;
C935            ;
C935 FE 0D                  CPI    CR      ;CR OR LF ARE SPECIAL CHARACTERS
C937 C2 46 C9               JNZ    NOCR     ;IT WASN'T A CR, BYPASS THE COLUMN TEST
C93A            ;
C93A 3A 08 C8               LDA    NCHAR    ;HAS A CR ALREADY PUT US IN COLUMN 1
C93D FE 00  CPI  000H  ;THIS WILL TELL US
C93F CA 0C C9  JZ  TERM1U  ;IT IS COLUMN 1, DON'T PROCESS IT
C942 78  MOV  A,B  ;LETS GET IT BACK
C943 C3 5E C9  JMP  TERM2U  ;IT ISN'T COLUMN 1, PROCESS IT
C946  ;
C946 FE 0A  NOCR:  CPI  LF  ;WAS IT A LINEFEED
C948 CA 5E C9  JZ  TERM2U  ;PROCESS IT
C94B  ;
C94B 3A 0C C8  LDA  ESCFL  ;A CTL CHAR---ARE WE IN AN ESC SEQUENCE
C94E B7  ORA  A  ;IF YES THEN OUTPUT CONTROL CHAR DIRECTLY TO
C94F C2 5E C9  JNZ  TERM2U  ;WE SURE ARE LET THE VDM DRIVER HANDLE IT
C952 C5  PUSH  B  ;SAVE THE CHARACTER
C953 06 1B  MVI  B,ESC  ;CTL CHAR TO VDM VIA ESC SEQUENCE
C955 CD 54 C0  CALL  VDMOT  ;PUT IT ON THE SCREEN
C958 06 07  MVI  B,7  ;SAY TO PUT OUT NEXT CHAR AS IS
C95A CD 54 C0  CALL  VDMOT  ;ALMOST READY
C95D C1  POP  B  ;GET IT BACK
C95E  TERM2U:  EQU  $  ;LETS PUT OUT THE CHARACTER
C95E CD 54 C0  CALL  VDMOT  ;PUT IT ON THE SCREEN
C961 C3 0C C9  JMP  TERM1U  ;LETS KEEP DOING IT
C964  ;
C964  ;
C964  ;  ;  S Y S T E M  E Q U A T E S
C964  ;  ;  ---------------  ---------------
C964  ;
C966  ;
C966  ;
C966  ;  ;  000A  LF  EQU  00AH  ;THE LINE FEED ASCII CHAR
C966  ;  ;  000D  CR  EQU  00DH  ;THE CARRIAGE RETURN ASCII CHAR
C966  ;  ;  001B  ESC  EQU  01BH  ;THE ESCAPE ASCII CHAR
C966  ;  ;  0080  MODE  EQU  080H  ;THE P.T. MODE CONTROL CHAR
C966  ;
C968  ;  ;  C919  SOUT:  EQU  0C019H  ;SYSTEM OUTPUT ENTRY POINT
C968  ;  ;  C91F  SINP:  EQU  0C01FH  ;SYSTEM INPUT ENTRY POINT
C968  ;  ;  C92E  KSTAT:  EQU  0C02EH  ;KEYBOARD INPUT ROUTINE
C968  ;  ;  C954  VDMOT:  EQU  0C054H  ;THIS ONE PUTS IT ON THE SCREEN
C968  ;  ;  C1C0  COMM1:  EQU  0C1C0H  ;RE-ENTRY INTO SYSTEM MONITOR
C968  ;  ;  C310  PSCAN:  EQU  0C310H  ;INPUT PARAMETER ROUTINE
C968  ;  ;  C806  IPORT:  EQU  0C806H  ;CURRENT INPUT PSEUDO PORT
C968  ;  ;  C807  OPORT:  EQU  0C807H  ;CURRENT OUTPUT PSEUDO PORT
C968  ;  ;  C808  NCHAR:  EQU  0C808H  ;PRESENT VDM CHARACTER POSITION
C968  ;  ;  C80C  ESCFL:  EQU  0C80CH  ;ESCAPE FLAG CONTROL BYTE
C968  ;
C96A  ;
C96A  ;
C96A  ;
C96A  ;  ;  END
0 ERRORS DETECTED
------------------------------------------------------------------------
PA

0E 01  SYMBOL TABLE

COMM1  C1C0  CR  000D  ESC  001B  ESCFL  C80C
IPORT  C806  KSTAT  C02E  LF  000A  MODE  0080
NCHAR  0808  NOC1  0946  OPORT  0807  PSCAN  C310
SINP  C01F  SOUT  C019  TERM1U  C90C  TERM2U  C95E
TERMU  C900  TINU  0924  TOUTU  C921  VDMOT  0054

PA
ALS-8 to Sol Patch

0000 *
0001 * This program alters the ALS-8 ram to
0002 * allow an ALS-8 that has not been updated
0003 * to interface with a Sol-20 using Solos.
0004 *
0005 *
0006 * Change standard input port to Sol
0007 * keyboard.
0008 *
D09F 0009 ORG 0D09FH
D09F FC 0010 DB KDATA
0011 *
0012 * Alter STAT routine in ALS-8 to match Sol
0013 * keyboard status port and compliment data
0014 * available flag.
0015 *
D0A4 0016 ORG 0D0A4H
D0A4 DB FA 0017 STAT IN KSTAT
D0A6 2F 0018 CMA
D0A7 E6 01 0019 ANI KDR
D0A9 C9 0020 RET
0021 *
0022 * This output driver saves the accumulator
0023 * and register B before calling SOUT in
0024 * Solos. It also checks the keyboard to see
0025 * if an escape has been hit. If so, a lump
0026 * will be made to FORMS in ALS-8. If not,
0027 * registers A and B will be popped off the
0028 * stack after the character has been printed.
0029 *
0030 OUTP8 PUSH PSW
D0AA F5 0031 PUSH B
D0AB C5 0032 CALL STAT
D0AC CD A4 D0 0033 JZ NOCHR
D0AF CA BB D0 0034 IN KDATA
D0B2 DB FC 0035 ANI 7FH
D0B4 E6 7F 0036 CPI ESC
D0B6 FE 1B 0037 JZ FORMS
D0B8 CA 60 E0 0038 NOCHR CALL SOUT
D0BB CD 19 C0 0039 POP B
D0BE C1 0040 POP PSW
D0BF F1 0041 RET
D0C0 C9 0042 *
0043 * Now make SYSIO and OUT8 point to this
0044 * version of OUTP8 as the standard output
0045 * driver.
0046 *
0047 D096 0048 ORG 0D096H SYSIO output driver
D096 AA D0 0049 *
0050 D0D0 0051 ORG 0D0D0H second byte of OUT8
D0D0 0051 DW OUTP8
D0D0 AA D0 0052 *
0053 D01B 0054 ESC EQU 1BH ASCII escape
0054 KDR EQU 01H data available flag
0054 D019 0055 SOUT EQU 0C019H Solos output entry
0055 KDATA EQU 0FCH Sol keyboard data port
0056 00FC 0057 KSTAT EQU 0FAH keyboard status port
0057 00FA 0058 FORMS EQU 0E060H entry point to ALS-8
0058 E060 0059 *
0060 * One more thing needs to be mentioned.
0061 * A non-updated version of the TXT-2
0062 * editor does not reset the hardware
0063 * scrolling port on a Sol. If this is
0064 * not done before entering the editor
0065 * the first line of the file may be
0066 * on a line other than the first line
0067 * of the screen. One remedy for this
0068 * is to always hit the CLEAR key before
0069 * executing the TXT-2 editor.
0070 *

Intel paper Tape Loader for Sol

| 0000 | LST |
| 0000 | STAK |
| 0000 | EQU |
| 0000 | $-1 |

0100 31 FF 1E 0000 LXI SP,STAK
0103 CD 06 1F 0000 CALL READ
0106 CD 45 1F 0000 READ CALL TTYIN
0109 FE 3B 0000 CPI ‘,’
010B CD 45 1F 0000 JNZ READ
0111 CD 2A 1F 0000 CALL CHAR
0117 57 0000 MOV D,A
0119 C8 0000 RZ
011C CD 2A 1F 0000 CALL CHAR
0121 67 0000 MOV H,A
0124 CD 2A 1F 0000 CALL CHAR
0127 6F 0000 MOV L,A
012A CD 2A 1F 0000 CALL CHAR
012E CD 2A 1F 0000 LOOP CALL CHAR
0131 77 0000 MOV M,A
0134 CD 45 1F 0000 CHAR CALL TTYIN
0137 CD 3D 1F 0000 CALL HEX
013A 07 0000 RLC
013D 17 0000 RAL
0140 17 0000 RAL
0143 17 0000 RAL
0146 5F 0000 MOV E,A
0149 CD 45 1F 0000 CALL TTYIN
014C CD 3D 1F 0000 CALL HEX
014F 83 0000 ADD E
0152 C9 0000 RET
0155 3D 30 0000 HEX SUI 48
0158 FF 0A 0000 CPI 10
015B D8 0000 RC
015E D6 07 0000 SUI 7
0161 C9 0000 RET
0164 DB F8 0000 TTYIN IN 0F8H
0167 E6 40 0000 ANI 64
016A CA 45 1F 0000 JZ TTYIN
016D DB F9 0000 IN 0F9H
016F E6 7F 0000 ANI 7FH
0172 C9 0000 RET

CHAR 1F2A HEX 1F3D LOOP 1F1E READ 1F06
A lot of you had fun with this program when we listed it in the first issue of ACCESS. We're repeating it now with modifications to run on Sol, and we think you'll like it even better this time around.

```
0000 * 0001 TRAIN XRA A ONCE UPON A TIME,
0100 AF 0002 OUT OFEH IN A CURIOUS LITTLE
0101 D3 FE 0003 LXI H,S6ED PLACE THERE WAS -
0106 01 00 08 0004 LXI B,2048 ..A TINY TRAIN
0109 36 20 0005 EMPTY MVI M,20H ..AND IT STAYED IN A
010B 23 0006 INX H ..TINY SHED
010C DB 0007 DCX B ..THAT WAS ALL EMPTY
010D AF 0008 XRA A
010E A8 0009 XRA B
010F C2 09 01 0010 JNZ EMPTY
0112 21 FA 03 0011 LXI H,CLOUD ..EXCEPT FOR A HUGE
0115 11 01 00 0012 LXI D,1
0118 06 0E 0013 MVI B,14 B
011A CD 50 01 0014 CALL SMO1 I
011D 11 28 00 0015 LXI D,43 L
0120 06 0C 0016 MVI B,12 L
0122 CD 50 01 0017 CALL SMO1 0
0125 11 30 00 0018 LXI D,48 W
0128 06 09 0019 MVI B,9 Y
012A CD 50 01 0020 CALL SMO1
012D 11 34 00 0021 LXI D,52 C
0130 06 04 0022 MVI B,4 L
0132 CD 50 01 0023 CALL SMO1 0
0135 11 3A 00 0024 LXI D,58 U
0138 06 02 0025 MVI B,2 D
013A CD 50 01 0026 CALL SMO1
013D 11 3D 00 0027 LXI D,61 0
0140 06 01 0028 MVI B,1 F
0142 CD 50 01 0029 CALL SMO1
0145 11 3E 00 0030 LXI D,62 0
0148 06 01 0031 MVI B,1 M
014A CD 50 01 0032 CALL SMO1 0
014D C3 59 01 0033 JMP ENGIN K
0134 *
0150 19 0035 SMO1 DAD D
0151 36 6F 0036 SMO2 MVI M,SMOKE ..COUGH
0153 23 0037 INX H
0154 05 0038 DCR B ..COUGH
0155 C2 51 01 0039 JNZ SMO2
0158 C9 0040 RET COMING OUT OF THE STACK
0141 *
0159 21 A8 05 0042 ENGIN LXI H,S6ED+322H A TINY LOCOMOTIVE
```
with a little smokestack and a little bell.

And a little dome and a cab.

With windows so that you could see into where the engineer and the fireman sat.

It had a beautiful polished brass boiler with the number "99" on the side of the cab.

It had a cowcatcher and lots of funny wheels and the things that connected.

And two very tiny wheels at the very back.

Quite lonely and yet it appeared very funny just sitting there on the rails with nothing at all to do well!

Let's run it just for fun.

Make train out of shed.

Move it from shed.

H, shed+3deh there on the.

Rays rails.

With nothing at all.

Ever.

To do.

W e l l !

Let's run it just for fun.

Make trip 13 miles.

Chuff once.

Make forward motion.

Make it turn.

Click odometer.

Make trip 13 miles.

Chuff once.

Make forward motion.

Make it turn.

Push B.

Make it turn.

B.

Click odometer.
01F9 AF 0110 XRA A
01FA A8 0111 XRA B  13 MILES YET?
01FB C2 F0 01 0112 JNZ CHOO NO
01FE C9 0113 RET .  DO NEXT TRIP
01FF 21 00 CC 0114 *
0202 0E D0 0116 MVI C,0000H
0204 7E 0117 COAL MOV A,M  ADD MORE COAL
0205 2B 0118 DCX H
0206 77 0119 MOV M,A
0207 23 0120 INX H
0208 23 0121 INX H
0209 7C 0122 MOV A,H
020A B9 0123 CMP C  ENOUGH COAL?
020B C2 04 02 0124 JNZ COAL NO, PUT MORE ON!
020D C9 0125 RET
020F 21 00 CC 0126 *
0212 7E 0127 HISS MOV A,M  TURN WHEELS
0213 23 0129 INX H
0216 C2 12 02 0131 JNZ HISS LOOK AGAIN.
0219 23 0132 FOUND INX H
021A 7E 0133 AXLE MOV A,M
021B FE 11 0134 CPI 11H WHAT QUARTER TURN
021C 23 0135 JNZ NXT1
0220 11 7E 14 0136 LXI D,147EH NEXT QUARTER TURN
0223 06 07 0137 *
0225 72 0138 AXL2 MOV M,D  WHEELS MOVED HERE
0226 05 0140 DCR B  LAST WHEEL?
0227 CA 52 02 0141 JZ WORK I'VE BEEN WORKIN'...
022A 23 0142 INX H  ..ON THE RAILROAD...
022B 73 0143 MOV M,E (SIDE RODS MOVED HERE)
022C 23 0144 INX H  ..ALL THE LIVE..
022D 05 0145 DCR B  ..LONG DAY.....
022E C2 25 02 0146 JNZ AXL2
022F 01 B4 14 0147 *
0231 1F 0148 JNZ AXL2 MOV M,D  WHEELS MOVED HERE
0232 05 0150 DCR B  LAST WHEEL?
0233 C2 3C 02 0151 JMP AXL1
0236 11 2D 13 0152 NXT2 CPI 13H QUARTER TURN
0239 C3 23 02 0153 JNZ NXT3
023C FE 13 0154 LXI D,125FH
023E C2 47 02 0155 JMP AXL1
0241 11 5F 12 0156 JZ QUIT QUITTIN' TIME
0244 C3 23 02 0157 JMP AXL1
0247 FE 12 0158 VX A NO, THEN HIGHBALL
0249 C2 1A 02 0159 XRA A
024C 11 2D 11 0160 JNZ D,112DH
024F C3 23 02 0161 JMP AXL1
0252 01 B4 14 0162 WORK LXI B,5300
0255 DB 0163 IN OFAH DID THE CONDUCTOR
0256 DB FA 0164 JNZ A NO, THEN HIGHBALL
0258 E6 01 0165 ANI RDA ..FLAG US DOWN
025A DB FC 0166 J2 QUIT QUITTIN' TIME
025C CA 65 02 0167 XRA A
025F AF 0168 XRA B
0260 B8 0169 JNZ OIL NO SQUEEKS, PLEASE!
0264 C9 0170 RET
0267 *  PUT A JUMP TO WHERE EVER YOU NEED TO
026C C3 04 CC 0172 *  TO RETURN IN THE NEXT STATION.
026E C3 04 CC 0173 *
0001 0174 QUIT JMP 0C004H YOUR RETURN TICKET HERE
0175 *
CC00 0177 RRY EQU 0CC00H VDM RAM
006E 0178 SMOKE EQU 6FH SMOKE CHARACTOR
0268 01 10 0179 CAB1 DW 1001H CAB DESCRIPTION
026A 10 5D 0180 CAB2 DW 5D10H ""
026C 7E 20 0181 CAB3 DW 207EH ""
026E 28 0A 0182 BOI1 DW 0A28H BOILER DESCRIPTION
0270 0A 19 0183 BOI2 DW 190AH ""
0272 0A 0A 0184 BOI3 DW 0A0AH ""
0274 0A 5B 0185 BOI4 DW 5BOAH ""
0276 39 39 0186 BOI5 DW 3939H ""
0278 5D 20 0187 BOI6 DW 205DH ""
027A 06 20 0188 FRA1 DW 2006H FRAME DESCRIPTION
027C 11 2D 0189 FRA2 DW 2D11H ""
027E 11 20 0190 FRA3 DW 2011H ""
0280 6F 2D 0191 FRA4 DW 2D6FH ""
0282 6F 20 0192 FRA5 DW 206FH ""
0284 19 19 0193 TIES DW 1919H TIES DESCRIPTION
0286 00 0194 SHED NOP . THIS IS SHED AREA
03FA 0195 CLOUD EQU $+0173H BEGINNING OF CLOUD
0001 * 0002 * SHOOTING STARS -- A BRAIN TEASER GAME 0003 * AN INTERACTIVE GAME FOR 8080 COMPUTER 0004 * THIS VERSION IMPLEMENTED FOR Sol 0005 * DEVELOPED AND DEBUGGED WITH ALS-8 0006 * PROGRAM DEVELOPMENT SYSTEM 0007 * 0008 * 0009 * PROGRAM ORIGINALLY SUBMITTED TO THE 0010 * HEWLETT-PACKARD SOFTWARE LIBRARY IN BASIC 0011 * SUBSEQUENTLY PUBLISHED BY PEOPLE'S COMPUTER 0012 * COMPANY IN SEPT 74 AS "TEASER" 0013 * WRITTEN IN 8008 CODE BY WILLARD I. NICO 0014 * IN BYTE MAY 76 AS "SHOOTING STARS" 0015 * ADAPTED TO 8080 CODE FOR USE WITH VDM-1 0016 * BY JERRY BACON 0017 * S.I.L. PROGRAMMING LABORATORY 0018 * BRASILIA, D.F. BRASIL 0019 * 0020 * DATE: 14 JAN 77 0021 * 0022 * 0023 ORG 0 0024 SHSDR LXI H,HEADR POINT TO HEADER MESSAGE 0025 CALL SCRN DISPLAY IT 0026 CALL KBD 0027 CPI 'N' CHECK IF RULES WANTED 0028 JZ START IF NOT, START GAME 0029 LXI H,PAGE1 OTHERWISE POINT TO 1ST PAGE 0030 CALL SCRN 0031 CALL KBD 0032 LXI H,PAGE2 POINT TO 2ND PAGE 0033 CALL SCRN 0034 CALL KBD 0035 START LXI D,VDM1 0036 CALL CLER CLEAR THE SCREEN 0037 LXI B,0001H INITIALIZE UNIVERSE 0038 MOV D,B CLEAR SHOT COUNTER 0039 CNTST INR D INCREMENT SHOT COUNTER
0040 DISP LXI H,STR1  DISPLAY UNIVERSE
0041 CALL TYPE
0042 LXI H,STR2
0043 CALL TYPE
0044 LXI H,STR3
0045 CALL TYPE
0046 LXI H,STR4
0047 CALL TYPE
0048 LXI H,STR6
0049 CALL TYPE
0050 LXI H,STR7
0051 CALL TYPE
0052 LXI H,STR8
0053 CALL TYPE
0054 LXI H,STR9
0055 CALL TYPE
0056 XRA A
0057 MOV A,C
0058 RRC
0059 LXI H,STR5
0060 CALL TYPE1
0061 WNTST MOV A,B     GET UNIVERSE PATTERN
0062 CPI 0FFH    CHECK FOR FRINGE STARS
0063 JNZ LSTST    IF NOT ALL PRESENT, CHECK FOR LOSS
0064 MOV A,C
0065 ORA A        GET CENTER STAR
0066 JNZ GTSTR    CONTINUE IF PRESENT
0067 LXI H,MSS4  IF NOT, GAME IS WON. POINT TO WIN MESS
0068 CALL SCRNB
0069 * 0070 * 0071 * 0072 * 0073 *
0074 * 0075 MVI E,'0'  INITIALIZE BINARY TO DECIMAL Conv.
0076 MOV B,E     0077 MOV C,E
0078 DCR D GET RID OF LAST SHOT
0079 MVI A,'9'+1 SET OVERFLOW CHECK
0080 MRDEC INR E INCREMENT 1'S
0081 CMP E CHECK FOR OVERFLOW
0082 JNZ TALLY CONTINUE IF NOT
0083 MVI E,'0' OTHERWISE, RESET 1'S
0084 INR C INCREMENT 10'S
0085 CMP C
0086 JNZ TALLY
0087 MVI C,'0'
0088 INR B INCREMENT 100'S
0089 TALLY DCR D DECREMENT SHOT COUNTeR
0090 JNZ MRDEC
0091 MVI A,'0'
0092 CMP B CHECK FOR LEADING 0
0093 JNZ THREE IF NOT, DISPLAY 3 DIGITS
0094 CMP C
0095 JNZ TWO
0096 JMP ONE
0097 THREE MOV M,B DISPLAY SCORE
0098 INX H
0099 TWO MOV M,C
0100 INX H
0101 ONE MOV M,E
0102 INX H
0103 XCHG
0104 LXI H,MSS5 POINT TO REST OF WIN MESS.
0105 CALL SCRNB
0106 PRNT1 CALL KBD
CPI 'Y'    CHECK FOR RESTART
JZ START IF YES, START AGAIN
JNZ GTSTR IF ANY ARE PRESENT CONTINUE GAME
MOV A,C
ORA A   IF NOT, CHECK FOR CENTER STAR
JNZ GTSTR IF PRESENT CONTINUE
LXI H,MESS3 OTHERWISE POINT TO LOST MESS.
CALL SCNB
JMP PRNT1
GSTR LXI H,MESS7 ASK FOR SHOT
CALL SCNB
NXTST CALL KBD
INX H
MOV M,A     ECHO SHOT
CALL DELAY
MVI E,9 SET MASK COUNTER
LXI H,MASK POINT TO MASKS
NXGRP CMP M CHECK FOR SHOT
JZ FOUND
DCR E
JZ INVAL   INVALID SHOT IF NOT FOUND
INX H POINT TO NEXT ENTRY
INX H
INX H
INX H
 JMP NXGRP
FOUND INX H
MOV A,M CHECK STAR POSITION
JNZ UNIV2 JMP IF FRINGE STAR
MOV A,C
CPI 1 CHECK FOR CENTER STAR
JNZ BDFEL IF NOT PRESENT, BAD SHOT
JMP NXBYT
UNIV2 MOV A,B
ANA M ISOLATE STAR SHOT
JZ BDFEL IF NOT PRESENT, BAD SHOT
* * *
* * *
* * *
* * *
INX H
MOV A,B
XRA M ALTER GALAXY
MOV B,A SAVE NEW PATTERN
INX H
MOV A,C
XRA M CHANGE CENTER STAR, IF NECESSARY
MOV C,A
JMP CNTST COUNT SHOT AND DISPLAY NEW UNIVERSE
INVAL CPI ESC CHECK IF INVALID SHOT WAS AN ESCAPE
JNZ NTVAL
LXI H,MESS6 IF SO POINT TO SURRENDER MESSAGE
CALL SCNB
JMP PRNT1
NTVAL LXI H,MESS2 POINT TO INVALID STAR MESSAGE
CALL SCNB
JMP NXTST GO TO NEXT SHOT
SCRN MOV A,D
STA TEMP SAVE D (SHOT COUNTER)
LXI D,VDM1 SET SCREEN ADDRESS
SCRN1 CALL CLER CLEAR & INITIALIZE SCREEN
SCRN2 MOV A,M
CPI EM CHECK FOR END OF MESSAGE
0174  JZ END
0175  STAX D DISPLAY CHARACTER
0176  INX H
0177  INX D
0178  JMP SCRN2
0179  END LDA TEMP GET SAVED SHOT COUNTER
0180  XCHG
0181  MOV D,A PUT IT BACK IN D
0182  RET
0183  SCRNB MOV A,D
0184  STA TEMP
0185  LXI D,SCRB T SET LOWER DISPLAY ADDRESS
0186  JMP SCRN1
0187  CLER PUSH D SAVE STARTING ADDRESS
0188  XRA A
0189  OUT VDM INITIALIZE VDM-1
0190  CLER1 MVI A, " " GET A SPACE
0191  STAX D
0192  INX D
0193  MOV A,D
0194  CPI BOTT CHECK FOR END OF SCREEN
0195  JNZ CLER1
0196  POP D RESTORE STARTING ADDRESS
0197  RET
0198  TYPE XRA A CLEAR A & CARRY
0199  MOV A,B GET UNIVERSE
0200  RRC
0201  MOV B,A
0202  TYPE1 JC STAR
0203  MVI M,'0' DISPLAY HOLE
0204  RET
0205  STAR MVI M,'* ' DISPLAY STAR
0206  RET
0207  KBD CALL DAV KEYBOARD INPUT ROUTINE
0208  JZ KBD
0209  IN DATA
0210  ANI 127
0211  RET
0212  DAV IN STAT
0213  CMA . CAN BE CHANGED TO A NOP
0214  ANI DAVM
0215  RET
0216  BDFEL LXI H,MESS1 POINT TO ERROR MESSAGE
0217  CALL SCRNB
0218  JMP NXTST
0219  DELAY PUSH D 2 SECOND DELAY ROUTINE
0220  PUSH PSW
0221  MVI D,2
0222  DLY1 MVI E,100
0223  DLY2 XRA A
0224  DLY3 DCR A
0225  JNZ DLY3
0226  DCR E
0227  JNZ DLY2
0228  DCR D
0229  JNZ DLY1
0230  POP PSW
0231  POP D
0232  RET
0233  TEMP DS 1
0234  *
0235  *
0236  *
0237  *
0238  *
0239  *
0240  *
MESS1 ASC "HEY! YOU CAN ONLY SHOOT STARS, NOT BLACK HOLES."
MESS2 ASC "THAT WASN'T A VALID STAR NUMBER. TRY AGAIN."
MESS3 ASC "YOU LOST THE GAME! WANT TO SHOOT SOME MORE"
MESS4 ASC "YOU WIN!! GOOD SHOOTING! YOU FIRED"
MESS5 ASC "SHOTS. BEST POSSIBLE SCORE IS 11 SHOTS."
MESS6 ASC "YOU GIVE UP TOO EASILY! WANT TO SHOOT SOME MORE"
MESS7 ASC "YOUR SHOT?"
MESS8 ASC "THE UNIVERSE."
MESS9 ASC "YOU SHOOT A STAR, (NOT A BLACK HOLE) BY TYPING "
MESS10 ASC "ITS NUMBER."
MESS11 ASC "1 2 3"
MESS12 ASC "4 5 6"
MESS13 ASC "7 8 9"
MESS14 ASC "* * *
MESS15 ASC "* * *
MESS16 ASC "* * *
MESS17 ASC "* * *
MESS18 ASC "YOU WIN IF YOU GET THE PATTERN IN THE MIDDLE."
MESS19 ASC "YOU LOSE IF YOU GET THE PATTERN ON THE RIGHT."
MESS20 ASC "READY TO PLAY."
MESS21 ASC "TYPE ANY KEY TO START. GOOD LUCK!"
0308 *
0309 *
0310 *
0311 *
0312 MASK DW 0131 MASKS FOR CHANGING UNIVERSE
0313 DW 010BH
0314 DW 0232H
0315 DW 0007H
0316 DW 0433H
0317 DW 0116H
0318 DW 0834H
0319 DW 0029H
0320 DW 0035H
0321 DW 015AH
0322 DW 1036H
0323 DW 0094H
0324 DW 2037H
0325 DW 0168H
0326 DW 4038H
0327 DW 00E0H
0328 DW 8039H
0329 DW 01D0H
0330 VDM1 EQU 0CC00H
0331 SCRBT EQU 0CF00H
0332 VDM EQU 0FEH CHANGE TO 0C8H FOR USE WITH VDM-1
0333 STR1 EQU 0CD1BH STAR LOCATIONS
0334 STR2 EQU 0CD20H
0335 STR3 EQU 0CD25H
0336 STR4 EQU 0CD9BH
0337 STR5 EQU 0CDA0H
0338 STR6 EQU 0CDA5H
0339 STR7 EQU 0CE1BH
0340 STR8 EQU 0CE20H
0341 STR9 EQU 0CE25H
0342 STAT EQU 0FAH KEYBOARD STATUS PORT
0343 DATA EQU 0FCH KEYBOARD DATA PORT
0344 DAVM EQU 01H DATA AVAILABLE MASK
0345 BOTT EQU 0D0H
0346 EM EQU 01H
0347 ESC EQU 1BH
Guess Who??