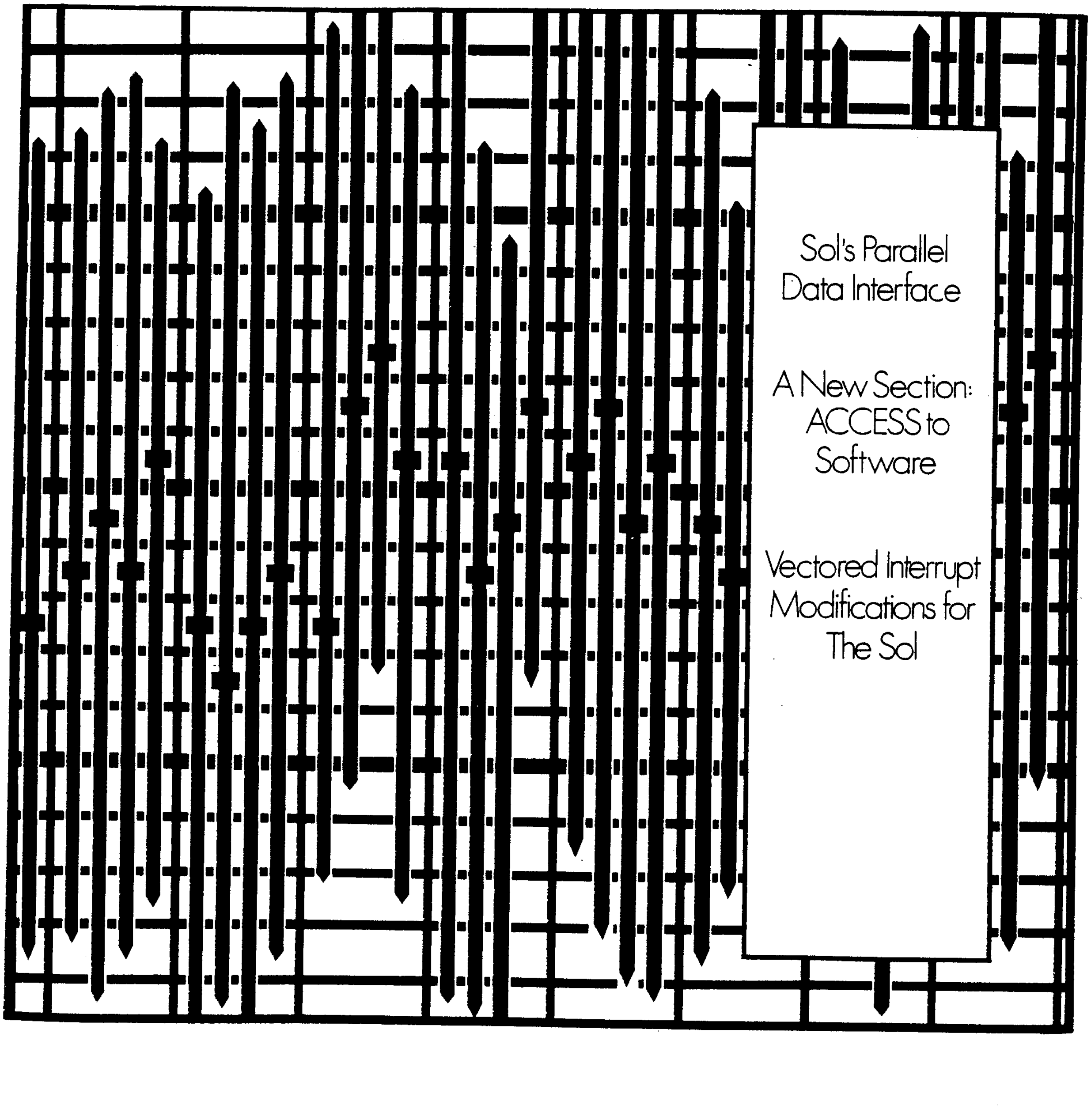
PROCESSOR TECHNOLOGY ACCESS.

75¢ per issue

Published by Processor Technology Corp., Pleasanton. Volume Two, Number One, March 1978



Sol's Parallel Data Interface

A New Section: ACCESS to Software

Vectored Interrupt Modifications for The Sol

A Letter From The [Outgoing] Editor

"Alas poor Yorick, for I knew him well." Willie the Shake

For those of you who don't already know, this is to inform you of my departure from PTC and my abdication of the editorship of ACCESS. The reason for this change is fairly simple: I'm going to work for myself as an independent consultant (free-lance technician in plebian terminology). Sorry I can't provide any scandalous gossip about my reasons for leaving. But, if you really must ask, well, let me tell you . . .

Obviously I couldn't have gotten to the position of being my own boss if it hadn't been for the training, experience and contacts (not to mention confidence) I gained at PTC.

Leaving definitely creates an empty spot in my life, having grown with the company (2 years). I feel like a teenager leaving home for the first time (sob, sob!).

But enough being maudlin and on to the corn. I think old Ralph should be able to take the helm and steer the ship-of-state through the uncharted waters of the future, in a manner and style befitting one of this stature.

My God! Somebody shut this fool up. He sounds like journalistic Muzak!!

Why should I shut up, dammitt. I'm the Editor. I can say anything.

Not anymore you can't. You're just the Outgoing Editor. You don't count any more!

Well, as you can see, there are good reasons for me being locked up in this place. And, what the hell, the food's OK.

I guess it's time to go, so I'll close this chapter of history with a sentimental goodbye to y'all. And expecially to Les Solomon who could be a guru. But whoever heard of a Jewish guru?

Love and kisses to all,

Aram Attarian

Aram Attarian II
(415) 534-6628
1924-10th Ave.
Oakland, CA 94606

PROCESSOR TECHNOLOGY
ACCESS.

Volume Two,
Number One
March 1978

Publisher
Processor Technology Corp.

Editor
Ralph I. Palsson

Staff Writers
Lee Felsenstein
Drew Rogge

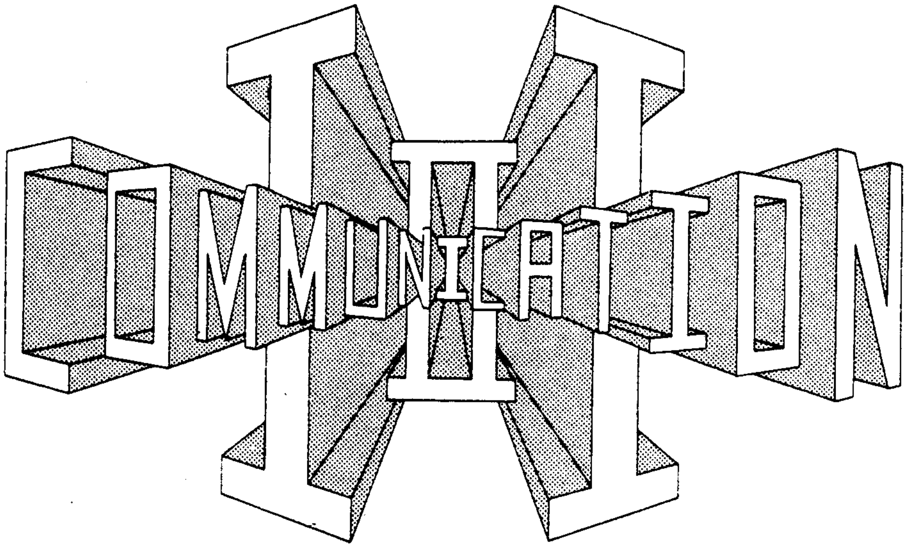
Design/ Production
Wible/Rampton Advertising

ACCESS is published approximately every six weeks. Subscription rate: $4 per year, from Processor Technology Corp., 7100 Johnson Industrial Way, Pleasanton, CA 94566.

Table of Contents

Have YOU Moved?

Please notify us of your change of address. Here's a handy form:
NAME ______________________________
NEW ADDRESS ______________________
____
____



A Letter From The [New] Editor

That's me, Ralph Palsson. Remember me from One-to-One Communication? I'm changing hats because Aram Attarian II, former ACCESS editor, lead Warranty Repair Applications Engineer, customer interfacer, and all-round good guy, has succumbed to the siren song of "making it" freelance. He's setting up an independent systems consultant and repair technician, and we're going to miss his red suspenders around here. He devoted a lot of creative energy and talent to making ACCESS what it is today, so I'll do my best to keep up the good work. For Aram's last words of wisdom to us, see his outgoing editorial on p. 2.

Now that I have this column to sound off in, we'll probably discontinue One-to-One Communication as separate feature. The original idea behind ACCESS was to make it a vehicle for one-to-one communication in its entirety, and that's beginning to happen, as you'll see from the many excellent reader contributions in this issue. I'll be laying it pretty heavy on you readers to keep up the good work-send us your program listings, letters, hardware modifications, applications, anything you think is worthwhile and would like to share.

The main change I initially hope to make in ACCESS is getting it on a somewhat more regular publication schedule, aiming at every 6 (±2) weeks. And I promise that everyone who had paid $4.00 for a year's subscription will actually receive 8 issues, regardless of how many leaves get torn off the calendar. As for the new format- well, you people seem to be as enthusiastic about it as we are, and we'll keep adding whatever improvements we think of. Again, your suggestions are valuable, like the punched-hole idea several readers have mentioned.

I like to think of ACCESS as one proof of the pride Processor Technology takes in providing customer support. So far, we have a pretty good track record, but we aim to make it even better. We have a very user-oriented Customer Service Dept., plus the more technically inclined applications and product support engineers regularly available to help you with debugging and troubleshooting your kits. They're also good with advice on how to interface a Sol or 3P+S to a peripheral device. And we even have a customer service software consultant available for phone discussions. But there's one thing none of these people can do, and that's help you with support for XYZ company's floppy disk or ABC's 16K board. No matter how expert they are, our engineers can't know the personal design quirks of a product they may never have seen, and really your guess is as good as theirs many times. We'll do as much "hand-holding" as we can by phone or letter, but please understand the distinction between help with interfacing and help with someone else's product.

Our Customer Service technicians are currently available daily from 1:30 to 4:00 p. m. PST for phone consultation. The number again is (415) 829-2600. We'll increase the hours as soon as we can add more trained personnel who can give you straight answers you deserve.

I'll end with my usual plug for Processor Technology dealers. These guys can provide you with a wealth of information and assistance without running up your phone bill. Our intentions are good, and we back them up with action as much as possible, but it's a fact of life that we're way out here in Pleasanton, CA. And if you're in Manhattan, Bloomington, or even Lompoc, you will probably get more personal service from the dealer who's helped you plan your system in the first place than you can get from a voice on the phone. The idea is to give you the best support possible, wherever it comes from.

Well, hope you enjoy ACCESS # 5, and in the words of AA II, y'all have a good time, y'heah?!

Ralph I. Palsson

READER FEEDBACK

Dear Editor:

HISTOGRAMS IN BASIC
(Or how to display multiple cursors)

Attempting to use Basic 5 for plotting the results of stock analysis, I wanted to plot a graph of the derived results. An approach has been developed, using the TAB function to start a plot, and a calculated value of a variable to end the plot. I also wanted to use reverse video to highlight data or to plot a bar made up of (a0 hex) "cursors:" Initially this was difficult, since many 7F masks were used in the VDM driver within CUTER to maintain a clean display. Therefore the VDM driver cannot be entered with a CALL to its entry address, since bit 7 is cleanly stripped off. The solution to this problem lies in addressing that portion of the VDM driver starting at location C0BB after having put the A0 into register B. That can be done as an immediate move in machine code necessitating a call from Basic to your machine code which in turn calls the driver, or by using the ARG command in basic before the call to the driver. This latter approach has an advantage in that the value of the ARG can also be calculated thus giving you control over the character chosen for the plot. (Incidentally, the 7F, delete character makes a nice chart due to the spacing between bars.) These ARG and CALL functions are used within a FOR-Next loop which is set up as described in the beginning of this description. That looping puts out the desired number of repeated characters and leaves VDM screen management control within the Basic/Cuter/VDM driver software management system.

In closing, let me send my praise and thanks for your product quality and utility. The PTCO additions have really enhanced my IMSAI.

Donald F. Petrie
831 Ponderosa Dr.
So. Daytona, Fla. 32019

Donald also enclosed a carbon of a letter he wrote in response to Dave Wrobel's query in the last ACCESS:

January 10, 1978

Dear Dave:

Don't despair! Inverse video is attainable. I noticed your letter and Mr. Attarian's response in issue number four of Processor Technology ACCESS.

I too was interested in the use of inverse video with my IMSAI/ VDM/CUTER configuration and was frustrated when using the SOUT entry point of the VDM driver. When I use an ANI 7F in the code, a cleanup, stripping off of the parity bit (in our case the VDM inverse video bit) is performed.

There is a simple solution to the problem, however, if you will use a machine language subroutine (callable from Basic, if desired) which calls the VDM driver after the housekeeping stripping referred to above. This driver routine, identified as "OCHAR," is at CUTER location C0BBH and in SOLOS at location C098H. Calling the driver at OCHAR with bit 7 high will result in the desired inverse video display.

You have a number of options in approaching this problem. In machine code, put the character into Register B, with bit 7 high and then call OCHAR. In BASIC/5, establish the character as the value of an ARG function and use the Basic ARG/CALL routines to call OCHAR. And if you want keyboard on-off toggling control of this feature, a simple custom output driver invoked by the SOLOS/CUTER pseudo-port 3 routine can be written to sense your command choice; then either add or strip-off bit 7 as appropriate and put the character in register B before calling OCHAR and returning to your system.

I hope one of these suggestions will be of use to you if you haven't already found a solution to your desire for inverse video display.

D. F Petrie

cc: Mr. Aram Attarian Editor, Processor Technology Access

Dear Donald,

Yes, Virginia, Inverse Video is attainable in Sol BASIC/5, thanks to your information. Reader Tom Digate has also sent us a contribution on this subject, a cassette written in BASIC15 demonstrating the uses of SOLO's VDM driver escape sequence. A printout of his program appears in the ACCESS to Software: section of this issue. Many thanks to you, too, Tom.

Aram is passionately hardware oriented and doesn't always realize that anything's possible with software!

Ralph

Dear Editor:

Enclosed you will find a check for $4.00 for my 1-year subscription to ACCESS. I was at my local computer store, (Itty Bitty Machine Company, Lombard, ILL), and they gave me a copy of ACCESS to look at. I thoroughly enjoyed reading the articles in the November issue. Congratulations on your fine publication. I have a question that you or maybe your readers could help me on. I have a IMSAI 8080 Micro Computer with the following boards: a 16K Processor Tech. Memory, VDM board by PTC, and my problem, 1 Tarbell Cassette interface board. I am going to purchase your (Sol) Basic Computer and I would like to know if there is a program to convert the CUTS format to Tarbell format for my cassette interface. I would like my IMSAI to run Sol software. If anyone can help, please do.

Thank you.

Robert Bennett
2046 Louis Street
Melrose Park, ILL 60164

Dear Bob,

Thanks for your letter; an ACCESS subscription has been entered for you and hopefully you're now reading this in your first copy.

Note that Sol BASIC/5 is an interpreter, not a compiler. If you're feeling any confusion about it, I'm sure your dealer can help.

About the CUTS/Tarbell matter. PTC doesn't support the Tarbell cassette interface with hardware or software, because we feel that CUTS is a superior system for several reasons. The CUTS (Computer Users Tape Systems) recording method was derived from and is compatible with the Byte (Kansas City) standard recording method, a system that PTC helped develop at the 1975 Kansas City meeting. We believe strongly in promoting software standardization and have put a lot of effort into the cause. CUTS was especially designed to permit widespread dissemination and exchange of low cost software. To that end, CUTS tapes are less expensive and easier to produce than Tarbell; PTC has shipped over 30,000 of them to date with less 1 % failure rate-not bad, if we do say so ourselves! And although CUTS is slightly slower than Tarbell, it requires less adjustment to work with a greater variety of tapes: and cassette recorders-again facilitating standardization and exchange.

So while we can't be too encouraging about your conversion project, we hope you will continue your interest in CUTS and Sol software. We think you'll end up very happy with it.

Ralph

Gentlemen,

I am enclosing a listing of a game written in BASIC called Hammurabi. I found it in one of the hobbyist magazines, Personal Computing, I think. I modified it to run in Basic/5 and I and my two boys have enjoyed playing it. I thought perhaps other Sol owners might also enjoy it. It will run in 8K and I run it on the CRT. I have enclosed a printout of a typical "year" of the game ending with a deliberate error which ends the game with a loss of kingdom. The entire successful game would have been too long a printout and I think the short example will give anyone looking at it the idea of how it's played.

I enjoyed the first issue of the new format of Access. Keep up the good work.

I visited your booth at the New York Coliseum and was very taken with the Sol set up to record a mailing list on tape which was open to public entry. If a copy of this program is available, I would very much like to obtain one.

Best wishes for your continuing success.

John Dowd
Lynn's Hair Fashions
New Road, Central Ave.
Linwood, N.J. 08221

Dear John,

Thanks for your comments and contributions. The mailing list program you saw in New York was just a demo version and not availavailable in its present form. However, our software development is now shifting to an emphasis on applications, and you'll find many exciting programs forthcoming in the near future.

To our readers,

Besides the Hammurabi program, John also sent us two others, PAROL and COMM. All three appear in the ACCESS to Software section-try them out!

Ralph

[redacted]

RE: So120

Gentlemen:

I purchased my So120 mid December 1977. After several hours of building (72) the Sol, to explicit and well defined hardware assembly instructions, I had a letter perfect "turn on:" My compliments to all connected with this portion of the manual and hardware parts staff.

I have encountered one problem which I have since corrected. This was a very heavy smearing and unstable effect on my video monitor Distortion severe enough to force shut down manually and of course interrupting work in process. I could still "SAVE" etc. But it was very difficult to read monitor and inputs, when instability occurred. Being more familiar with R.F., video and communications equipment than logic, I found a quick and normal solution to the problem, one which is a standard practice in the R.F. field.

Solution:

The clock generator at "14.31818 MHz" is fairly open to external field excitation, by nearby C.B. and other transmitter frequencies. This unprotected portion of your circuit tends to wander if excited by an external signal and does not settle down. The reason is the XTAL can itself, floating above ground. A short jumper between the can and ground corrected the problem immediately.

I hope this assists other Sol users.

Now the bad news. I realize most people purchasing Sol's are probably more software oriented than I am. Therefore, your company did a painstaking task on the hardware. But your software writeup for a beginner (28 years communications) in computer talk and functions is very poor.

I know it is difficult and a lot of self study must be done, which I am now doing, but!

Might I suggest you acquire some unsuspecting person with no software knowledge, sit him/her in front of the machine, hand him "Console Manual" and "Solos Manual" and watch the confusion. If it is me, I apologize. But after showing your book to 3 top software experts, they agreed, it was written by a software expert.

Example:

After 10 hours of reading, re-reading, 1 trip to vendor (80 miles), no luck, re-reading, etc., trying to work out the CRC computation. Nowhere could I see it written.

"When you desire to save a file. Input SAVE (title) 5 words and press "RETURN:"

N.B. "Making sure recorder is in record mode:"

But most important, "File header is then automatic:"

I guess I'm saying try to use layman's language, if possible. If you re-read "SOLOS" I am sure you will agree.

I am very satisfied with the unit and want to congratulate those who worked hard on design and appearance. Well done.

B. W. Poste
17 Lonsdale Place,
Barrie, Ontario

Dear B. W.,

Thanks for the compliments, and for the hardware fix. Many of our readers will find your contribution from the RF/analog world extremely useful.

Your critical points on documentation and software are very worthwhile. PTC's Technical Documentation Dept. is always looking for input that will help future publications be more clearly presented, on the appropriate level of expertise. Meanwhile, we'll try to provide some concrete follow through in the next issue of ACCESS.

Ralph

Gentlemen:

Loved the new ACCESS format, but tell me, will my one-year subscription be extended to cover the first 8 issues (since you nominally publish every 6 weeks)? You haven't quite been able to keep up with the announced schedule for either ACCESS or ALS-8 users notes, (though the latter are not officially a product of P.T.). Also, (this may sound like a dumb question), is there any chance of re-publishing the 1st 3 issues in the new format? Even if it costs $2-$3 per copy? The new format is so much better, especially for my only good source of a CUTTER (or SOLOS) listing* Thanks.

Barry Watzman
2330 Millenium Lane
Reston, VA 22091

*Besides, then I could keep a complete collection of ACCESS in a 3-ring binder!

Dear Barry;

Yes. We'll make sure you (and other ACCESS subscribers) receive 8 issues before you come up for second year billing. Also, we aim to put the journal back onto a more reasonable schedule.

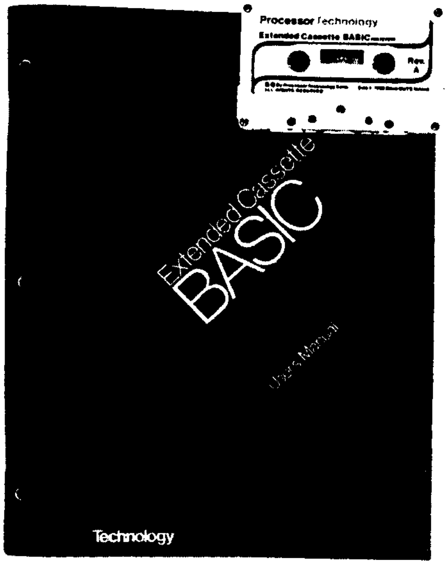
As you might guess, costs of republishing out-of-print ACCESS

issues are prohibitive. However, I'm thinking of possibly doing a collection of the "Best of ACCESS" as a separate issue or reprinting the most requested articles from past issues. How do you other readers

feel? (NOTE. We still have a lot of our 2nd issue on hand; numbers 1 and 3 are out of print.)

Ralph

Extended Cassette BASIC Is Here!



Our long-promised Extended Cassette BASIC has been released and is now at your local PTC dealer. It's a powerful language, and we'll be publishing assorted program listings in future ACCESS issues to stimulate your imagination on its applications. Just for starters, note that Extended Cassette BASIC features complete string functions and matrix operations. And the documentation is excellent.

It's rumored there may be some bootleg copies floating around. Assure yourself of getting a quality tape and complete package by buying only if the cassette and manual are identical to those shown in the photograph here.

There's one slight bug we've discovered in using manual FOR/NEXT loops. The bug occurs when there is a FOR/NEXT loop within a FOR/NEXT loop which will not allow the execution of the inner loop. The squasher allows the execution of the internal loop rather than searching for a loop elsewhere in the program. It's easily fixed by following this 5-step procedure:

1) GET BASIC from the tape. Note the load address and length.
2) Type EN B3E in SOLOS. (CR)
3) Type the following numbers, separated by a space:
   C1 CA 2E 0B/(CR)
4) SAVE BASIC (refer to Section 3.4 in the BASIC User's Manual). Let the tape run in the RECORD mode for 10-15 seconds before re-recording. You can use a new tape if you wish, but the original one will work just as well. Use the load address and length information you obtained in Step 1 of this procedure, and make the auto start address 0 and ending address 3F84.

If you're using the original cassette, affix scotch tape over the two holes on the back of the cartridge. Remove tape when finished; this will afford you "write protection."

The Sol Users' Society Is Up And Running

Last issue we reported on the formation and first meeting of SOLUS, the Sol Users' Society. It's an organization for owners of Sol computers or of Sol-type computers compatible with other PTC products. Specifically, if you have an 8080 or Z-80 microcomputer you're using with SOLOS, CUTER, SOL/ CUTS cassette interface, or any functionally equivalent operating system, you might like to join. The membership is world-wide.

Currently, SOLUS has several good services going, beginning with their own bi-monthly (approximately!) newsletter to keep members informed on hardware, software, new products, bugs, local chapter meetings, and other items of interest. They maintain a Software Library which collects and distributes public domain programs for a nominal charge. Local chapters hold meetings to exchange software and ideas, and the club headquarters, in convenient proximity to PTC, keeps a close communication link going with us. Finally, qualified SOLUS volunteers are testing products for Sol compatibility and reporting their experiences in the newsletter.

To refresh you on the club's goal and organization: The stated goals are to facilitate communication among SOLUS members, to provide a mechanism of exchange for Sol-compatible software, to give PTC feedback from SOLUS members, and to encourage the development and testing of Sol-compatible hardware and software produced by independent sources. Their relationship with PTC and other manufacturers is co-operative but independent. SOLUS is primarily a personal computer users' group, but special interest groups can be formed within the club on subjects like commercial use, health care, education, scientific applications, etc. The society is supported by dues and volunteer efforts.

To join SOLUS: If you live in the U.S., Canada or Mexico, send $10.00 in check or money order to SOLUS, P.O. Box 23471, San Jose, CA 95153. If you live anywhere else, send $15.00. Dealers and manufacturers of Sol-compatible equipment or software should contact SOLUS at the above address for details on special memberships.

What To Do With Your BASIC/5 Programs Now That You Have Extended Cassette BASIC

Unfortunately, programs written in Sol BASIC/5 won't run with the new Extended Cassette (formerly called 8K) BASIC interpreter. But you can save yourself several thousand tedious keystrokes by using the following assembly language program to save your BASIC/5 programs in text form on CUTS cassette. Then it will be easy to get and run a text file on the new interpreter whenever you wish.

The program uses the byte access tape routines in the SOLOS/ CUTER monitor. CLIST is written to be used as custom output diver in conjunction with pseudo port 3.

When you use the program, load the machine code at the addresses indicated in the assembly listing below. Set the custom output port address to the beginning address of this program. Get and execute BASIC/5. Get the program you want to write to the tape. From BASIC/5, set the output port to 3 and type LIST.

After your program has been saved, this routine will set the current output port back to 0.

```
      000D              CR    EQU   0DH   ASCII carriage return
      000A              LF    EQU   0AH   ASCII line feed
                        *
                        *  Solos equates
                        *
      C01C              ROUT  EQU   0C01CH
      C007              FOPEN EQU   0C007H
      CODA              FCLOS EQU   0C00AH
      C010              WRBYT EQU   0C010H
      C807              OPORT EQU   0C807H
                        *
                        *  The program starts here.
                        *  This is also the entry point.
                        *
CB00                          ORG   0CB00H
      CB00              CLIST EQU   $
CB00  C5                      PUSH  B     Save registers
CB01  D5                      PUSH  D
CB02  E5                      PUSH  H
                        *
CB03  78                      MOV   A,B   get character
                        *
CB04  FE OD                   CPI   CR    Carriage return?
CB06  C2 26 CB                JNZ   CL2
                        *
CB09  21 73 CB                LXI   H,LCHAR
CB0C  7E                      MOV   A,M   Get last character
CB0D  FE 0D                   CPI   CR    If not two carriage returns in
CB0F  C2 26 CB                JNZ   CL2   ... in a row then nothing special
CB12  23                      INX   H     Else check if first set
CB13  7E                      MOV   A,M   ...of carriage returns
CB14  B7                      ORA   A     If not then close file and
CB15  C2 4B CB                JNZ   CLOOP reset program
```

```
CB18  3D                      DCR   A     Else say we already
CB19  77                      MOV   M,A   had first time through
CB1A  3A 62 CB                LDA   UNIT  Get unit to write to
CB1D  21 63 CB                LXI   H,HEADR     Point to tape header
CB20  CD 5E CB                CALL  OPNOP Open the tile
CB23  C3 47 CB                JMP   GBACK
CB26  78                CL2   MOV   A,B
CB27  FE 0B                   CPI   LF+1
CB29  DA 2F CB                JC    CL3
CB2C  32 73 CB                STA   LCHAR
                        *
CB2F  3E 00             CL3   MVI   A,0   Reflect character to screen
CB31  CD 1C C0                CALL  AOUT
                        *
CB34  3A 74 CB                LDA   FIRST Check if writing yet
CB37  B7                      ORA   A
CB38  CA 47 CB                JZ    GBACK Return if not
                        *
CB3B  78                      MOV   A,B
CB3C  FE 0B                   CPI   LF+1
CB3E  DA 47 CB                JC    GBACK
                        *
CB41  3A 62 CB                LDA   UNIT  Get unit to write to
CB44  CD 10 C0                CALL  WRBYT Write the byte
                        *
CB47  E1                GBACK POP   H     Restore registers
CB48  D1                      POP   D
CB49  C1                      POP   B
                        *
CB4A  C9                      RET   .     ...to Basic/5
      CB4B              CLOOP EQU   $
                        *
CB4B  3A 62 CB                LDA   UNIT
CB4E  CD 0A C0                CALL  FCLOS Close file
CB51  AF                      XRA   A
CB52  32 73 CB                STA   LCHAR Reset program for next
CB55  32 74 CB                STA   FIRST ...time through
CB58  32 07 C8                STA   OPORT Change output port to 0
CB5B  C3 47 CB                JMP   GBACK
                        *
                        *
      CB5E              OPNOP EQU   $
                        *
CB5E  CD 07 C0                CALL  FOPEN
CB61  C9                      RET
                        *
CB62  01                UNIT  DB    1     Change to 2 to use tape unit 2
                        *
CB63  43 4C 49 53       HEADR ASC   'CLIST'
      54
CB68  00                      DB    0
CB69  54                      DB    'T'   for TEXT
CB6A  00 01                   DW    256   Length of block
CB6C  00 00                   DW    0
CB6E  00 00                   DW    0
CB70  00 00                   DW    0 Three spares
CB72  00                      DB    0
                        *
CB73  00                LCHAR DB    0
CB74  00                FIRST DB    0
```

```
CB75  00                    INFLG DB    0
                            *
```

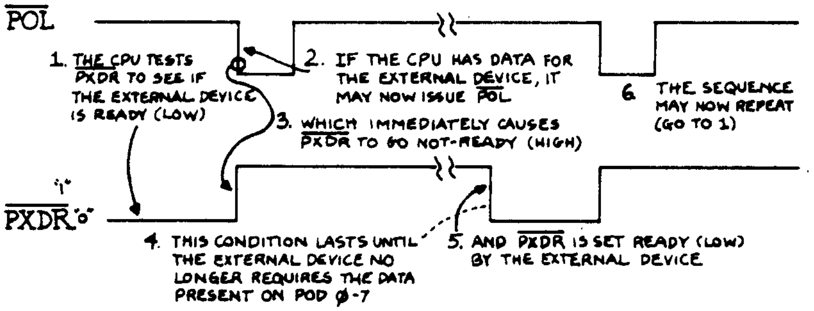
Interfacing To The Sol's Parallel Port

by Lee Felsenstein

The parallel input/output port on the Sol, J2, is intended to allow highspeed transfer of data between the CPU and outside devices. The circuitry was designed to be symmetrical, so that two devices having identical interfaces can communicate with each other. I n contrast, the 3P+S circuit puts out a positive-going data strobe pulse but requires either a negative-going data strobe pulse or a steady level in the return direction. As a result, two 3P+S boards cannot communicate without additional circuitry, while two Sols can communicate through their parallel ports with nothing but wires between them.

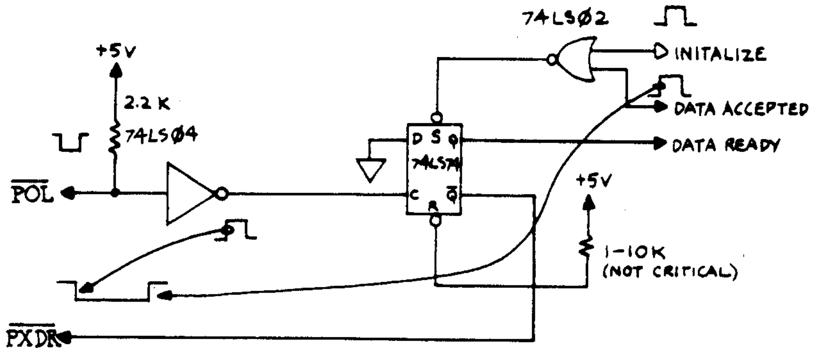
Data is transferred on eight-bit-wide paths into and out of the Sol, through the pins of J2 labelled PID0-7 (data into the CPU) and POD0-7 (data out of the processor). Polarity of the data on these pins is positive---that is, a "high" logical level is equivalent to a binary "1." Data is latched at the output side of the interface and is not latched at the input side. This means that after an output instruction to the parallel port (port FD) is issued, the new data remains present at the output pins of the connector indefinitely, until the next output instruction is issued. Similarly, the external device which provides data to the Sol through J2 must keep its data present on the PID pins long enough for it to be "picked up" by the Sol CPU.

FIG 1.
HANDSHAKING SEQUENCE



POL
1. THE CPU TESTS PXDR TO SEE IF THE EXTERNAL DEVICE IS READY (LOW)
2. IF THE CPU HAS DATA FOR THE EXTERNAL DEVICE, IT MAY NOW ISSUE POL
3. WHICH IMMEDIATELY CAUSES PXDR TO GO NOT-READY (HIGH)
6. THE SEQUENCE MAY NOW REPEAT (GO TO 1)
"1"
PXDR "0"
4. THIS CONDITION LASTS UNTIL THE EXTERNAL DEVICE NO LONGER REQUIRES THE DATA PRESENT ON POD Ø-7
5. AND PXDR IS SET READY (LOW) BY THE EXTERNAL DEVICE

FIG. 2.
CIRCUITRY FOR HANDSHAKING



Supervisory signals, known as "handshaking" signals, are provided in both directions to allow the devices on each side of the interface to know when data has been accepted and when each may present the next eight bits of data. Each side issues a negative-going "strobe pulse" when its data is ready and latched. In the direction going out from the Sol, this signal is called POL, for Parallel Output Load. The line over the signal name signifies that the signal is "active low." All of the handshaking signals are active low, so that a disconnected plug automatically results in a "not ready" condition. Resistors connected from the in-coming handshaking signals to +5 volts will "pull up" these pins to a high level automatically when this happens. Due to the pecularities of TTL logic, it is almost impossible to "pull down" a disconnected input.

The external device sees the POL signal and immediately acknowledges it with a high level on the PXDR pin, (Parallel External Device Ready). The Sol CPU will test the level of this signal before issuing a new POL. The external device sets the PXDR low again as soon as it has accepted the data and no longer requires its presence on the POD pins. Fig. 1 shows the sequence relationship between these signals.

In the incoming direction to the Sol CPU, the signals and the sequence are the same, with only the names and pin numbers changed. The data strobe fed to the Sol is PDR, (Parallel Data Ready). The Sol circuitry immediately responds with a high level on its PIAK pin (Parallel Input Acknowledge), which continues until the CPU has accepted the data presented at the PID pins.

Fig. 2 shows a suggested circuit for providing the handshaking signals from an external device. Note the provision for an "Initialize" pulse. This prevents the circuit from starting up in a "not ready" condition and "hanging up" the Sol CPU. Some external devices will not require this signal, but as an interface designer or troubleshooter, you should understand the possible pitfall. 74LS series logic is shown, but any TTL logic may be used. Each input of the parallel interface should present no more than one standard TTL equivalent load (1.6 maximum current source).

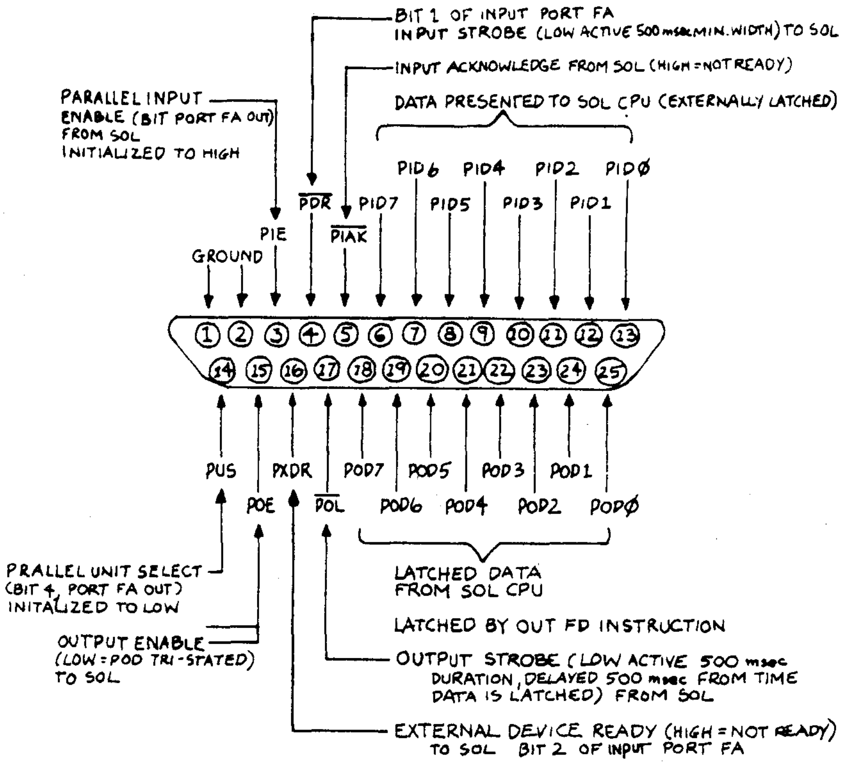
Three signals remain unexplained so far. These are PIE (Parallel Input Enable), POE (Parallel Output Enable), and PUS (Parallel Unit Select). These are direction-control signals to allow sharing of data lines among several devices. The latches in the Sol have "tri-state" output drivers, which can be "turned off" so that another device can take over the same data lines with its own tri-state drivers. The signal which turns off the Sol drivers is POE (Parallel Output Enable). This signal is high-active, so that a low level on this pin will cause the output drivers to release. It will therefore keep the output drivers on when no connection is made to that pin of J2.

The Sol provides a signal which is intended to control the output-enable signal of an external device. This is PIE, a latched bit which is initialized to a logical "high" level and which can be set be setting bit 3 of output port FA. (The remaining bits of FA control other things, so care must be taken to keep them in the proper pattern when such an output is performed.) PIE may be used as a "direction control" bit if you want to share the data lines in both directions: for example, if you need to save wire or to use the interleaving wires of a ribbon cable for grounds to provide shielding on long runs. PIE would then be connected to POE of the Sol and to a similar signal on the external device which recognized a "low" level as its output-enable signal. The PID and the POD lines may now use the same eight wires.

Note that the layout of the pins on E revision Sols and above (F, etc.) is such that the PID 0 and POD 0 lines are physically adjacent. This allows easy cross-connection of these lines in parallel when using a ribbon cable. If a 25-wire ribbon cable is used, the interleaving wires are free for use as grounds, helping to reduce the problems of "glitches" which often plague high-speed parallel interfaces.

The PUS (Parallel Unit Select) pin is another latched bit, bit 4 of output port FA, which is initialized to a logical "low" level. It may be used to select two different external devices to receive or transmit data. These devices may share the same PID lines if their outputs are tri-state. Together with the PIE bit, the PUS bit allows the CPU to select four external devices, providing that appropriate decoding and response circuitry is included in the external device.

FIG 3
J2-VIEWED FROM REAR OF SOL


BIT 1 OF INPUT PORT FA
INPUT STROBE (LOW ACTIVE 500 msec MIN. WIDTH) TO SOL
INPUT ACKNOWLEDGE FROM SOL (HIGH = NOT READY)
PARALLEL INPUT ENABLE (BIT PORT FA OUT) FROM SOL INITIALIZED TO HIGH
DATA PRESENTED TO SOL CPU (EXTERNALLY LATCHED)
PID6
PID4
PID2
PIDØ
$\overline{PDR}$
PID7
PID5
PID3
PID1
PIE
$\overline{PIAK}$
GROUND
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
PUS
PXDR
POD7
POD5
POD3
POD1
POE
$\overline{POL}$
POD6
POD4
POD2
PODØ
PRALLEL UNIT SELECT (BIT 4, PORT FA OUT) INITALIZED TO LOW
LATCHED DATA FROM SOL CPU
LATCHED BY OUT FD INSTRUCTION
OUTPUT ENABLE (LOW = POD TRI-STATED) TO SOL
OUTPUT STROBE (LOW ACTIVE 500 msec DURATION, DELAYED 500 msec FROM TIME DATA IS LATCHED) FROM SOL
EXTERNAL DEVICE READY (HIGH = NOT READY) TO SOL BIT 2 OF INPUT PORT FA

Like all logic-level signals, the parallel interface signals should be kept away from high voltages such as AC and static electricity. Cables longer than 10 feet should not be used; or if they are, precautions should be taken to accommodate the "ringing" effects which will result during signal transitions. The most typical such precaution is "terminating" the line with a resistor, usually about 120 ohms connected to +5 volts. The output circuits of the Sol parallel port cannot drive a load of less than 330 ohms, so additional drivers may be necessary for such terminated lines. It is also a poor practice to connect edge-sensitive inputs such as the clock or trigger input of a flip-flop to the cable directly. Use an inverter or a Schmitt trigger circuit such as the 74LS14, 8837, or 8836 (8T380). Edge effects can be impossible to see on an oscilloscope, so precautions are usually advisable when attempting to "debug" a cranky interface.

The Sol parallel interface provides many opportunities for imaginatvie and economical interfacing. we would be interested in hearing your results in applications which you consider noteworthy. Good luck!

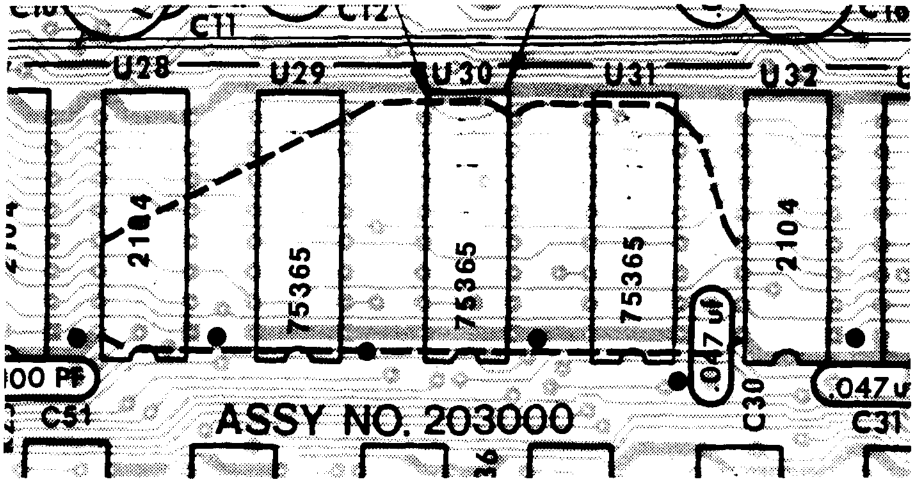
Important Notice To 16KRA Owners

We're currently up to Revision Level J on the 16KRA memory board in our tireless quest to improve the reliability of the circuit. Most of the earlier boards work fine, and if yours isn't exhibiting any intermittent failures, you can stop reading now. We don't recommend making the modification unless you are having trouble.

However, here's the modification procedure for those of you who need it. The assembly in question is Number 203000, shown on page VI-1 of the 16KRA manual. The portion of the drawing reproduced here, Detail A, shows the modification to increase reliability. Five jumper wires have been added to the trace side of the board, all at ground potential, in order to improve ground return paths. U30 is shown as a 75365-the alternate part 3207 should not be used at this location. To make the changes:

1) Connect pin 5 of U28 to pin 9 of U29.
2) Connect pin 9 of U29 to pin 9 of U30.
3) Connect pin 9 of U30 to pin 9 of U31.
4) Connect pin 9 of U31 to pin 5 of U32.
5) Connect pin 8 of U28 to pin 1 of U32.

These changes don't require any change in the schematic shown on page VI-2, or in the Theory of Operation, Section V.



Vectored Interrupt Capability For Sol

Although the 8080A microprocessor used in the Sol provides vectored interrupt capability, neither Sol nor Processor Technology's other S100 modules require use of this capability. However, Sol does provide a means for implementing vectored interrupt if the interrupt signal is made available by a circuit board inserted in the S100 bus. This board is connected to S100 bus pin 96, SINTA. If you want this board to generate interrupts, you'll need to add two jumpers, shown below, to the Sol P.C. These jumpers enable the SINTA signal to reach the memory decoder circuit; they can be added after assembling the Sol P.C., or even after the entire Sol has been assembled and tested.

The jumpers should be made from #24 solid insulated wire (not provided). Their electrical effects can be seen in Drawing X-16. They can be left in place even if no S100 board generates interrupts. and S100 bus pin 96 may float with no interference.

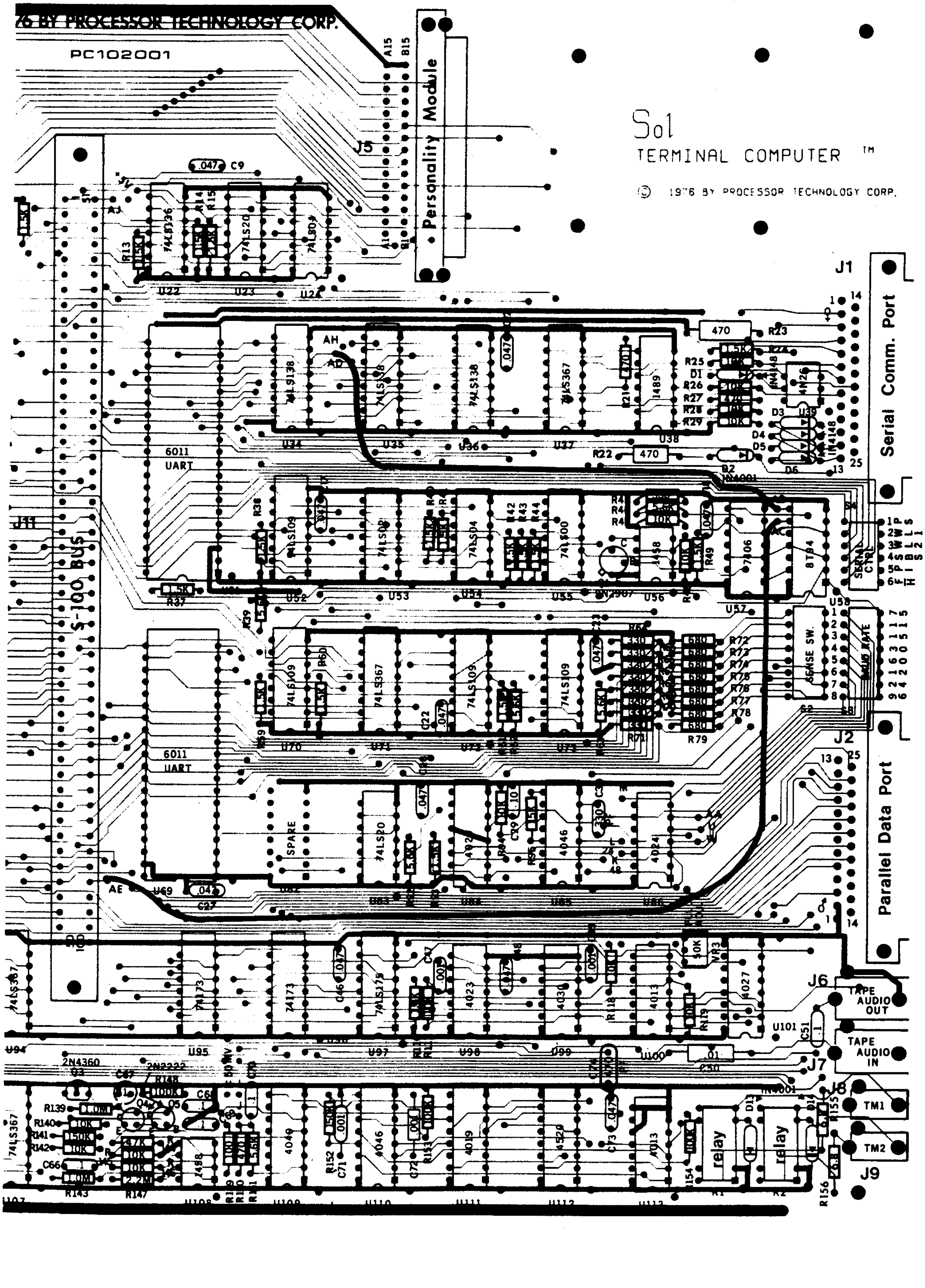
The necessary modifications for both Revision D and Revision E level circuit boards are as follows:

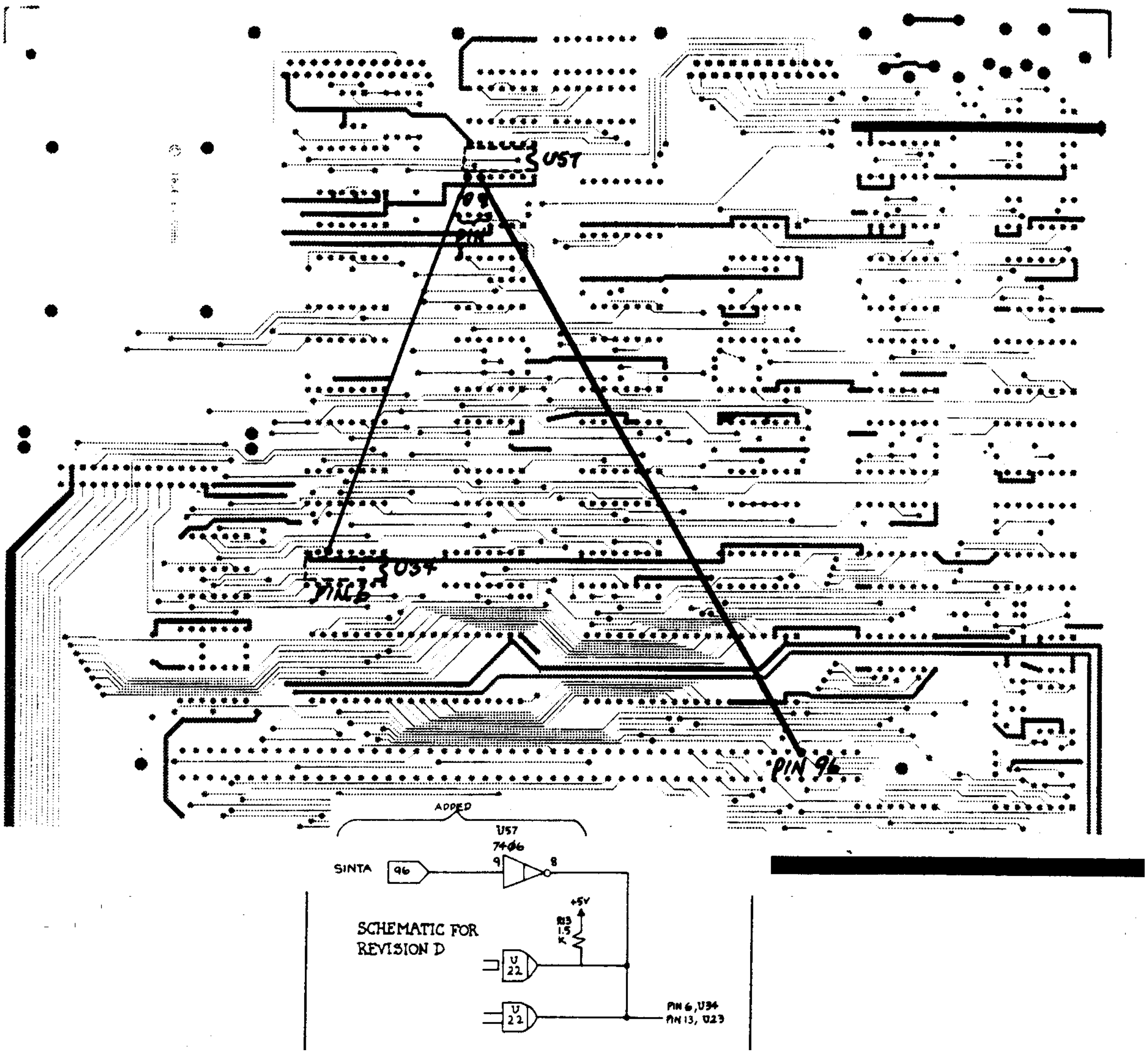
Modification to Revision E board. Sol-PCB:

1) Strip .1" of insulation from one end of two 8" lengths of wire. Insert the stripped ends into pads AC and AB (near U58), from the component side of the board. Solder and check for solder bridges.
2) Dress the wires as shown in the accompanying diagram, trim to the correct length, and strip .1" from the loose ends. Insert the wire from pad AB into pad AD, and the wire from pad AC into pad AE. Again, solder and inspect for solder bridges.
3) Attach the long runs of wire to the board with silicone compound or tape.

Modification to Revision D Board, Sol-PCB:

1) Strip.1" insulation from each end of a 6.25" length of wire. Connect one end to pin 96 of J10 (S100 connector); the other end goes to pin 9 of U57.
2) Strip .1" of insulation from each end of a 4" wire. Connect pin 8 of U57 to pin 6 of U34.
3) Attach the long runs of wire to the board with silicone compound or tape.


76 BY PROCESSOR TECHNOLOGY CORP.
PC102001
Sol
TERMINAL COMPUTER ™
© 1976 BY PROCESSOR TECHNOLOGY CORP.
Personality Module
J5
A15
B15
J1
Serial Comm. Port
J2
Parallel Data Port
J11
S-100 BUS
6011 UART
6011 UART
SPARE
SENSE SW.
BAUD RATE
SERIAL CTRL
J6
TAPE AUDIO OUT
J7
TAPE AUDIO IN
J8
TM1
J9
TM2
relay
relay
74LS136
74LS20
74LS04
74LS138
74LS08
74LS138
74LS367
1489
74LS109
74LS00
74LS04
74LS00
1458
7406
8T34
74LS109
74LS367
74LS109
74LS109
74LS20
402
4046
4024
74LS367
74173
74173
74LS175
4023
4030
4013
4027
4049
4046
4019
4520
4013
1488
U22
U23
U24
U34
U35
U36
U37
U38
U39
U51
U52
U53
U54
U55
U56
U57
U58
U69
U70
U71
U72
U73
U82
U83
U84
U85
U86
U94
U95
U96
U97
U98
U99
U100
U101
U107
U108
U109
U110
U111
U112
U113
2N4360
2N2222
2N2907
1N4001
1N4148
C9
C27
C50
C51
C73
R13
R23
R22
R37
R39
R72
R73
R74
R75
R76
R77
R78
R79
R139
R140
R141
R142
R143
R147
R148
R154
R155
R156
470
470
.047
.01
1.0M
100K
150K
10K
2.2M
47K
680
330
1.5K
5.6K
AH
AD
AE
S1
S2
S3
S4
1P
2WP
3SW
4SL
5PL
6HS



The Sol-20 And The OP-80A Paper Tape Reader

by Drew Rogge

Oliver Audio Engineering's excellent and inexpensive OP-80A Paper Tape reader is one of the most popular peripherals in use with the Sol-20 and Sol User's Manual does contain some information about interfacing it. However, there's an error in the manual, and hardware interfacing can be a tricky chore in any case. Especially when it comes to using the Sol's 8-bit parallel output port. (See Lee Felsenstein's article on the parallel port else where in this issue.) So here's what to do about the manual error, and also some other useful tidbits.

Section VII (Operating Procedures) of the Sol Users' Manual discusses the OP-80A. The error is on page 31: pin 5 of J2 (Sol PDI connector) should be labeled IAK, *not* IAK. Therefore, the "input Acknowledge" jumper on the paper tape reader should be connected to ACK, not ACK.

The intensity of the light source is also an important consideration in bringing the OP-80A up. You can allow just enough brightness to operate the tape reader, yet avoid saturating the tape, if you start by placing the light source far enough above the reader so that the SP LED indicator remains off. Then lower the light source just far enough to activate the LED, and keep it there.

Pat Tuell, of the San Jose Byte Shop, has contributed a driver routine for using the OP-80A, which we've included along with this article.

OP-80/ Sol Driver

by Pat Tuell

Enter the routine from keyboard or cassette; hook up the reader as specified. Execute C900, and start pulling the tape through. Intel-formatted paper tapes will automatically be loaded into the correct memory locations and control will automatically return to SOLOS.

Have fun getting your data loaded!

Note: In case you haven't already made the correction, please note that in ACCESS #4 there is a routine to load Intel-formatted tapes from a teletype connected to the serial port. Memory location 1F0A should be "3A" instead of "3B."

```
C900             0010 *
C900             0020 *     THIS WILL LOAD INTEL FORMAT PAPER TAPE
C900             0030 *INTO SOL THRU OP80 READER CONNECTED TO SOL
C900             0040 *PARALLEL DATA INTERFACE
C900             0050 *
C900             0060 *
C900 3E 10       0070 START  MVI    A,010H    STARTUP
C902 D3 FA       0080        OUT    0FAH
C904 CD 93 C9    0090 CALL   DAZL
C907             0100 *
C907 DB FA       0110 PTIN   IN     0FAH      INPUT FROM OP80 ASC.
C909 E6 02       0120        ANI    002H
C90B C2 07 C9    0130        JNZ    PTIN
C90E 3E 00       0140        MVI    A,000H
C910 D3 FA       0150        OUT    0FAH      TOGGLE ACK
C912 3E 10       0160        MVI    A,010H
C914 D3 FA       0170        OUT    0FAH
C916 DB FD       0180        IN     0FDH
C918 E6 7F       0190        ANI    07FH
C91A C9          0200        RET
C918             0210 *
C91B D6 30       0220 HEX    SUI    030H      CONVERT ASC TO HEX
C91D FE OA       0230        CPI    00AH
C91F D8          0240        RC
C920 D6 07       0250        SUI    007H
C922 C9          0260        RET
C923             0270 *
C923 CD 07 C9    0280 CHAR   CALL   PTIN      ASSEMBLE BYTE
C926 CD 1B C9    0290        CALL   HEX
C929 CD 35 C9    0300        CALL   ROLL
C92C 5F          0310        MOV    E,A
C92D CD 07 C9    0320        CALL   PTIN
C930 CD 1B C9    0330        CALL   HEX
C933 83          0340        ADD    E
C934 C9          0350        RET
C935             0360 *
C935 07          0370 ROLL   RLC
C936 17          0380        RAL
C937 17          0390        RAL
C938 17          0400        RAL
C939 C9          0410        RET
C93A             0420 *
C93A CD 07 C9    0430 CARD   CALL   PTIN      ASSM BYTE, DISPLAY
C93D F5          0440        PUSH   006H
C93E 47          0450        MOV    B,A
C93F CD 19 C0    0460        CALL   SOUT
C942 F1          0470        POP    006H
C943 CD 1B C9    0480        CALL   HEX
```

```
C946 CD 35 C9    0490        CALL   ROLL
C949 5F          0500        MOV    E,A
C94A CD 07 C9    0510        CALL   PTIN
C94D F5          0520        PUSH   006H
C94E 47          0530        MOV    B,A
C94F CD 19 CO    0540        CALL   SOUT
C952 F1          0550        POP    006H
C953 CD 1B C9    0560        CALL   HEX
C956 83          0570        ADD    E
C957 C9          0580        RET
C958             0590 *
C958 CD 23 C9    0600 LOOP   CALL   CHAR       INPUT DATA
C95B 77          0610        MOV    M,A
C95C 23          0620        INX    H
C95D 15          0630        DCR    D
C95E C2 58 C9    0640        JNZ    LOOP
C961 C3 64 C9    0650        JMP    READ
C964             0660 *
C964 CD 07 C9    0670 READ   CALL   PTIN       LOOK FOR A BLOCK
C967 FE 3A       0680        CPI    03AH
C969 C2 64 C9    0690        JNZ    READ
C96C 06 OA       0700 RD1    MVI    B,00AH
C96E CD 19 C0    0710        CALL   SOUT
C971 06 OD       0720        MVI    B,00DH
C973 CD 19 C0    0730        CALL   SOUT
C976 CD 3A C9    0740        CALL   CARD
C979 57          0750        MOV    D,A
C97A C8          0760        RZ
C97B 06 OA       0770        MVI    B,00AH
C97D CD 19 C0    0780        CALL   SOUT
C980 06 0D       0790        MVI    B,00DH
C982 CD 19 CO    0800        CALL   SOUT
C985 CD 3A C9    0810        CALL   CARD
C988 67          0820        MOV    H,A
C989 CD 3A C9    0830        CALL   CARD
C98C 6F          0840        MOV    L,A
C98D CD 23 C9    0850        CALL   CHAR
C990 C3 58 C9    0860        JMP    LOOP
C993             0870 *
C993 CD 07 C9    0880 DAZL   CALL   PTIN
C996 FE 3A       0890        CPI    03AH
C998 C2 9E C9    0900        JNZ    WAIT
C99B C3 6C C9    0910        JMP    RD1
C99E 47          0920 WAIT   MOV    B,A
C99F CD 19 CO    0930        CALL   SOUT
C9A2 C3 93 C9    0940        JMP    DAZL
C9A5             0950 *
C9A5             0960 SOUT   EQU    0C019H
C9A5             0970 *
C9A5             0980 *
C9A5             0990 *
C9A5             1000 *      CABLE PINOUT
C9A5             1010 *
C9A5             1020 *    SOL                 OP80
C9A5             1030 *REV D   REV E      PIN     SIGNAL
C9A5             1040 * 6      13           1     ID0
C9A5             1050 * 7      12          16     ID1
C9A5             1060 * 8      11           2     ID2
C9A5             1070 * 9      10          15     ID3
C9A5             1080 *10       9           3     ID4
```

```
C9A5                1090 *11       8           14      ID5
C9A5                1100 *12       7            4      ID6
C9A5                1110 *13       6           13      ID7
C9A5                1120 *
C9A5                1130 *14      14           5       ACK-
C9A5                1140 * 4       4           6       RDA-
C9A5                1150 * 2       2           8       GND
C9A5                1160 *                     9       +5V
C9A5                1170 *
C9A5                1180 *
C9A5                1190 *
C9A5                1200 * THE SOL SURVIVOR
C9A5                1210 *
```

We hope this section is going to become a regular feature of ACCESS, and we hope that you're going to write it! All of the programs in this issue were contributed by readers, many of them as submissions to the Applications Contest described last issue. Naturally, they're all very much in the running we're just printing them now because they seemed too good to keep to ourselves. Keep your contributions coming---if this selection is a fair sample of the quality we can expect, ACCESS to Software may prove the most popular part of the magazine.

You'll note the miniscule size of the listings. Well, the less space they take, the more we can run, and the more space we can devote to meatier prose in other sections. We're thinking that most users will only have to read the programs in once, and hoping that you'll become resigned to squinting your eyes as a one-time (per program), character-building experience, worth it for the rewards of acquiring the program.

Also check the reader feedback section of this issue for additional comments by the authors of these programs.

Again, you're encouraged to submit any and all programs you have. For purposes of accuracy and the suppression of typos, we encourage you to send either a cassette or photo-reproducible printout of the listing. If you send a cassette, we'll replace it with a blank one so you don't lose on the deal. Also try to provide as much support information as you can, like your uses for the program, where it originated, etc. It's a nice way to get your name in print!

Contributions to this issue:

Hammurabi (game, modified for BASIC/5)	John Dowd
PAROL (payroll records)	John Dowd
COMM (weekly sales/payroll computations)	John Dowd
Inverse Video in BASIC/5 (just that!)	Tom Digate
BASIC VDM driver fix (to work with MITS 4.0 BASIC)	Warren Saunders

Thanks, fellows!

BASIC VDM Driver Fix to Work with MITS 4.0 BASIC

This useful emendation was contributed by ACCESS reader Warren Saunders of Oakton, Virginia. Add the following statements to BASIC and you'll be in the running!

```
0106 FOR Y=L TO L+416
0154 DATA 195,4000,5000,241,245,230,127,254,32,210,154
0156 DATA 1001,254,7,202,6,1000,254,13,202,47,1000
0158 DATA 254,1,202,47,1000,254,26,202,47,1000,241
0160 DATA 201,0,0,245,229,213,197
0233 DATA 254,127,194,47,1000,241,201
```

Note: BASIC VDM DRIVER with these changes remains compatible with previous versions of MITS BASIC.

PAROL and COMM

John Dowd told us he adapted this program to work with Sol BASIC/5 from a version found *in Some Common BASIC Programs*, a book by Lon Poole and Mary Borcher. No data is stored on cassette --- only the programs. John uses them with a printer to keep records of beauty shop income (Lynne's Carrousel of Hair Fashions in Linwood, NJ), sales analysis, employee commission and payroll. Previous balances and sales totals are entered by keyboard from cash register tapes and payroll records. He says that using Sol has cut the time required for these jobs from 4 hours to 45 minutes!

```
LIST
10   PRINT "PAYROLL PROGRAM"
30   DATA 16,33,18,76,22,143,24,182,28,220,32,297,36,355
40   DATA 15,61,18,105,22,223,25,278,28,355,32,432,36,500
50   F1=.0605
60   W2=19.2
70   W1=14.4
90   DIM F1(28)
100  FOR I=1 TO 28
110     READ F1(I)
120  NEXT I
130  PRINT %Z2%
140  PRINT "MARITAL STATUS  (1=SINGLE, 2=MARRIED)"
150  INPUT S
160  PRINT "WITHHOLDING TAX EXEMPTIONS"
170  INPUT W
180  INPUT "REGULAR PAY     ",R
190  INPUT "COMMISSION     ",C
195  P=R+C
200  PRINT "YTD GROSS PAY"
210  INPUT Y
220  PRINT "YTD FICA"
230  INPUT Y2
240  PRINT "YTD FED. W/T"
250  INPUT Y1
260  PRINT "YTD N.J. W/T"
270  INPUT Y3
280  PRINT "YTD SUI"
290  INPUT Y4
300  PRINT "YTD NET"
310  INPUT Y5
311  O=Y-Y1-Y2-Y3-Y4
312  IF O=Y5 THEN 319
313  GOTO 130
319  P1=P1+P
320  PRINT "OPERATOR NUMBER"
330  INPUT O
```

```
340  O=P-W1*W
350  T1=0
360  FOR I=2 TO 7
370     X=2*I+14*(S-1)-1
380     IF G<=F1(X-1) THEN 460
390     IF G>F1(X+1) THEN 420
400     T1=T1+(G-F1(X-1))*F1(X-2)/100
410     GOTO 460
420     T1-T1+(F1(X+1)-F1(X-1))*F1(X-2)/120
430  NEXT I
440  T1=T1+(G-F1(X+1))*F1(X)/100
450  T1=T1+(G-F1(X+1))*F1(X)/100
460  T2=0
470  T2=INT((P*F1)*100+.5)/100
480  T3=0
481  T3=INT((P-W*W2)*2.2+.5)/100
482  IF T3>=0 THEN 489
483  T3=0
489  T4=0
490  T4=INT(P+.5)/100
495  T1=INT((T1*100)+.5)/100
500  N=P-T1-T2-T3-T4
510  PRINT "OPERATOR NUMBER",O
520  PRINT "GROSS PAY",P
530  PRINT "FICA",T2
540  PRINT "FED W/T",T1
550  PRINT "N.J. W/T",T3
560  PRINT "SUI",T4
570  PRINT "NET PAY",N
571  T6=T6+T1: T5=T5+T2: T7=T7+T3: T8=T8+T4: N1=N1+N
580  Y2=Y2+T2
590  Y1=Y1+T1
600  Y3=Y3+T3
610  Y4=Y4+T4
620  Y5=Y5+N
630  Y=Y+P
631  PRINT "YTD GROSS",Y
632  PRINT "YTD FICA",Y2
633  PRINT "YTD FED W/T",Y1
634  PRINT "YTD N.J. W/T",Y3
635  PRINT "YTD SUI",Y4
636  PRINT "YTD NET PAY",Y5
640  SET 0=2
642  PRINT TAB(5);"REGULAR PAY";TAB(25);R       This section prints out a
644  PRINT TAB(5);"COMMISSION";TAB(25);C        slip for each operator showing
646  PRINT TAB(5);"GROSS PAY";TAB(25);P         reg pay, commission, gross pay,
648  PRINT TAB(5);"FICA";TAB(25);T2             deductions, net pay.
650  PRINT TAB(5);"FEDERAL W/T";TAB(25);T1
652  PRINT TAB(5);"NEW JERSEY W/T";TAB(25);T3
654  PRINT TAB(5);"SUI";TAB(25);T4
656  PRINT : PRINT
658  PRINT TAB(5);"NET PAY";TAB(25);N
660  PRINT : PRINT : SET O=0
670  INPUT "AGAIN? (1=YES 0=NO)",A,
680  IF A=1 THEN 130
690  PRINT "TOTAL GROSS PAY",P1
700  PRINT "TOTAL FICA",T5                      This section gives payroll totals
710  PRINT "TOTAL FED W/T",T6                   for week and computes amount to
720  PRINT "TOTAL N.J. W/T",T7                  be reserved for payment of payroll
730  PRINT "TOTAL SUI",T8                       taxes, state & federal.
```

```
740  PRINT "TOTAL NET PAY",N1
750  R=INT((P1*2*F1)+(P1*.05)+1+T6+T7)
760  PRINT "RESERVE FOR WEEK",R
```

Comm

```
LIST
50   INPUT "YTD SALES",G
75   INPUT "YTD SALARIES",P9
100  INPUT "MONDAY TOTALS      ",M1,M2,M3,M4,M5,M6 -- Cash register totals for day.
110  PRINT "ENTRY ERROR?   ( 1,YES  0,NO )"
120  INPUT E
130  IF E=1 THEN 100
150  M=M1+M2+M3+M4+M5+M6
160  PRINT %Z2%
200  PRINT M1;M2;M3;M4;M5;M6,M
300  INPUT "TUESDAY TOTALS     ",T1,T2,T3,T4,T5,T6
312  PRINT "ENTRY ERROR?  (1,YES  0,NO )"
320  INPUT E
330  IF E=1 THEN 300
400  T=T1+T2+T3+T4+T5+t6
500  PRINT T1;T2;T3;T4;T5;T6,T
600  INPUT "WEDNESDAY TOTALS     ",W1,W2,W3,W4,W5,W6
610  PRINT "ENTRY ERROR?  (1,YES  0,NO )"
620  INPUT E
630  IF E=1 THEN 600
700  W=W1+W2+W3+W4+W5+W6
800  PRINT W1;W2;W3;W4;W5;W6,W
900  INPUT "THURSDAY TOTALS     ",H1,H2,H3,H4,H5,H6
910  PRINT "ENTRY ERROR?  (1,YES  0,NO )"
920  INPUT E
930  IF E=1 THEN 900
1000 H=H1+H2+H3+h4+H5+H6
1100 PRINT H1;H2;H3;H4;H5;H6,H
1200 INPUT "FRIDAY TOTALS     ",F1,F2,F3,F4,F5,F6
1210 PRINT "ENTRY ERROR?  (1,YES  0,NO )"
1220 INPUT E
1230 IF E=1 THEN 1200
1300 F=F1+F2+F3+F4+F5+F6
1400 PRINT F1;F2;F3;F4;F5;F6,F
1500 INPUT "SATURDAY TOTALS     "S1,S2,S3,S4,S5,S6
1510 PRINT "ENTRY ERROR?  ( 1,YES  0,NO )"
1520 INPUT E
1530 IF E=1 THEN 1500
1600 S=S1+S2+S3+S4+S5+S6
1700 PRINT S1;S2;S3;S4;S5;S6,S
1800 INPUT "DAYS WORKED     ",D1,D2,D3,D4,D5,D6
1810 PRINT "ENTRY ERROR?  ( 1,YES  0,NO )"
1820 INPUT
1830 IF E=1 THEN 1800
1900 PRINT "DAYS WORKED     ";D1;D2;D2;D4;D5;D6
2000 B1=M1+T1+W1+H1+F1+S1
2100 B2=M2+T2+W2+H2+F2+S2    -- Each operator's weekly sales total computed
2200 B3=M3+T3+W3+H3+F3+S3
2300 B4=M4+T4+W4+H4+F4+S4
2400 B5=M5+T5+W5+H5+F5+S5
2500 B6=M6+T6+W6+H6+F6+S6
2600 B=B1+B2+B3+B4+b5+B6     -- Overall weekly total
2700 G=G+B                   -- New YTD total
2800 C1=INT(((B1-(D)*44))*100)/4)/100      C1-C6-Operator's Commission
```

```
2810 IF C1>=0 THEN 290O
2820 C1=0
2900 C2=INT(((B2-(D2*40))*100)/4)/100
2910 IF C2>=0 THEN 3000
2920 C2=0
3000 C3=INT(((B3-(D3*44))*100)/4)/100
3010 IF C3>=0 THEN 3100
3020 C3=0
3100 C4=INT(((B4-(D4*44))*100)/4)/100
3110 IF C4>=4 THEN 3200
3120 C4=0
3200 C5=INT(((B5-(D5*44))*100)/4)/100
3210 IF C5>=0 THEN 3300
3220 C5=0
3300 C6=INT(((B6-(D6*44))*100/4)/100
3310 IF C6>=0 THEN 3400
3320 C6=0
3400 C=C1+C2+C3+C4+C5+C6
3500 P1=C1+(D1*22)
3689 P2=C2+(D2*20)
3708 P3=C3+(D3*20)   -- Each operator's pay for week
3800 P4=C4+(D4*22)
3900 P5=C5+(D5*22)
4000 P6=C6+(D6*22)
4100 P0=P2+P3+P4+P5+P6   -- Total pay for week excluding operator#1--my wife
4200 P=P0+(.12*P0)       -- Total pay & employer taxes (approximate)
4300 R=INT((P/B)*100)
4400 R1=INT((B1/B)*100)
4500 R2=INT((B2/B)*100)
4600 R3=INT((B3/B)*100)
4700 R4=INT((B4/B)*100)  -- % of total sales done by each operator
4800 R5=INT((B5/B)*100)
4900 R6=INT((B6/B)*100)
5100 PRINT "ENTER  CUSTOMER COUNT (I-VI)"  Enter # of customers served by each
5200 INPUT L1,L2,L3,L4,L5,L6               operator
5300 PRINT L1;L2;L3;L4;L5;L6
5400 K1=K1+L1: K2=K2+L2: K3=K3+L3: K4=K4+L4: K5=K5+L5: K6=K6+L6
5600 K=K1+K2+K3+K4+K5+K6    -- Total customers
5700 Ll=INT((K1/K)*100): L2=INT((K2/K)*100): L3=INT((K3/K)*100)  % of customers
5800 L4=INT((K4/K)*100): L5=INT((K5/K)*100): L6=INT((K6/K)*100)  served by each
5900 A=INT((B/K)*100)/100  Average $ val. of sale per customer   operator.
5940 IF K1>0 THEN 6000
5950 A1=0: GOTO 6005
6000 A1=INT((Bl/K1)*100)/100
6005 IF K2>0 THEN 6010
6006 A2=0: GOTO 6015
6010 A2=INT((B2/K2)*100)/100
6015 IF K3>0 THEN 6020             Each operator's average S val. of sale
6016 A3=0: GOTO 6025               per customer
6020 A3=INT((B3/K3)*100)/100
6025 IF K4>0 THEN 6030
6026 A4=0: GOTO 6035
6030 A4=INT((B4/K4)*100)/100
6035 IF K5>0 THEN 6040
6036 A5=0: GOTO 6045
6040 A5=INT((B5/K5)*100)/100
6045 IF K6>0 THEN 6050
6046 A6=0: GOTO 6055
6050 A6=INT((B6/K6)*100)/100
6100 SET 0=2    Output to expandor printer by parallel port
```

```
6200 PRINT "   I       II       III       IV       V       VI          TOT"
6300 PRINT TAB(1);M1;TAB(8);M2;TAB(17);M3;TAB(28);M4;TAB(38);M5;TAB(46);M6,
6400 PRINT TAB(55);M
6500 PRINT TAB(1);T1;TAB(8);T2;TAB(17);T3;TAB(28);T4;TAB(38);T5;TAB(46);T6,
6600 PRINT TAB(55);T
6700 PRINT TAB(1);W1;TAB(8);W2;TAB(17);W3;TAB(28);W4;TAB(38);W5;TAB(46);W6,
6800 PRINT TAB(55);W
6800 PRINT TAB(1);H1;TAB(8);H2;TAB(17);H3;TAB(28);H4;TAB(38);H5;TAB(46);H6,
7000 PRINT TAB(55);H
7100 PRINT TAB(1);F1;TAB(8);F2;TAB(17);F3;TAB(28);F4;TAB(38);F5;TAB(46);F6,
7200 PRINT TAB(55);F
7300 PRINT TAB(1);S1;TAB(8);S2;TAB(17);S3;TAB(28);S4;TAB(38);S5;TAB(46);S6,
7400 PRINT TAB(55);S
7450 PRINT
7500 PRINT TAB(0);B1;TAB(8);B2;TAB(17);B3;TAB(28);B4;TAB(38);B5;TAB(46);B6,
7600 PRINT TAB(55);B
7700 PRINT "COMMISSIONS"
7800 PRINT TAB(0);C1;TAB(8);C2;TAB(17);C3;TAB(28);C4;TAB(38);C5;TAB(46);C6
7820 PRINT " INDIVIDUAL SALARIES INCLUDING COMMISSION."
7830 PRINT TAB(0);P1;TAB(8);P2;TAB(17);P3;TAB(28);P4:TAB(38);P5;TAB(46);P6
7840 PRINT "                  TOTAL PAY INCLUDING TAXES    PERCENT OF INCOME"
7900 PRINT TAB(30);P,: PRINT TAB(55);R;"%"
7950 PRINT "PERCENTAGE OF TOTAL BUSINESS DONE BY EACH OPERATOR."
8000 PRINT TAB(0);R1;TAB(8);R2;TAB(17);R3;TAB(28);R4;TAB(38);R5;TAB(46);R6
8100 PRINT "   I       II       III       IV       V       VI "
8200 PRINT "PERCENTAGE OF WEEK'S CUSTOMERS SERVED BY EACH OPERATOR."
8300 PRINT TAB(0);L1;TAB(8);L2;TAB(17);L3;TAB(28);L4;TAB(38);L5;TAB(46);L6
8400 PRINT "OPERATOR'S AVERAGE SALE PER CUSTOMER"
8500 PRINT TAB(0);A1;TAB(8);A2;TAB(17);A3;TAB(28);A4;TAB(38);A5;TAB(46);A6
8600 PRINT "SHOP'S AVERAGE SALE PER CUSTOMER"
8700 PRINT TAB(8);A
8710 PRINT "OPERATOR'S % OF SALARY TO SALES"
8715 IF Bl>0 THEN 8720
8716 V1=0: GOTO 8721
8720 V1=INT(((P1+(.12*P1))/B1)*100):
8721 IF B2>0 THEN 8725                   Compute each operator's % of salary
8722 V2=0: GOTO 8730                     paid to sales for week.
8725 V2=INT(((P2+(.12*P2))/B2)*100)
8726 IF B3>0 THEN 8730
8727 V3=0: GOTO 8731
8730 V3=INT(((P3+(.12*P3))/B3)*100):
8731 IF B4>0 THEN 8735
8732 V4=0: GOTO 8736
8735 V4=INT(((P4+(.12*P4))/B4)*100)
8736 IF B5>0 THEN 8740
8737 V5=0: GOTO 8741
8740 V5=INT(((P5+(.12*P5))/B5)*100):
8741 IF B6>0 THEN 8745
8742 V6=0: GOTO 8750
8745 V6=INT(((P6+(.12*P6))/B6)*100)
8750 PRINT TAB(0);V1;TAB(8);V2;TAB(17);V3;TAB(28);V4;TAB(38);V5;TAB(46);V6
9000 PRINT : PRINT : PRINT
9100 P9=P9+P
9200 R0=INT(((P9/6)*100)+.5)         %salaries/sales YTD
9300 PRINT "YTD SALES",G
9400 PRINT "YTD SALARIES",P9
9500 PRINT "YTD PERCENT OF SALARIES/SALES",RO;"%"
9550 SET O=0
9600 END
```

Hammurabi

```
LIST
80   PRINT "TRY YOUR HAND AT GOVERNING ANCIENT SUMERIA"
85   PRINT "SUCCESSFULLY FOR A 10 YEAR TERM OF OFFICE.": PRINT
90   LET D1=0: LET P1=0
100  LET Z=0: LET P=95: LET S=2800: LET H=3000: LET E=H-S
110  LET Y=3: LET A=H/Y: LET I=5: LET Q=1
210  LET D=0
215  PRINT : PRINT : PRINT "HAMURABI:  I BEG TO REPORT TO YOU,": LET Z=Z+1
217  PRINT "IN YEAR";Z;",";D;" PEOPLE STARVED,";I;" CAME TO THE CITY."
218  LET P=P+I
227  IF Q>0 THEN 230
228  LET P=INT(P/2)
229  PRINT "A HORRIBLE PLAGUE STRUCK!  HALF THE PEOPLE DIED."
230  PRINT "POPULATION IS NOW",P,
232  PRINT "THE CITY NOW OWNS";A;" ACRES."
235  PRINT "YOU HARVESTED";Y;" BUSHELS PER ACRE."
250  PRINT "RATS ATE";E;" BUSHELS."
253  PRINT "YOU NOW HAVE";S;" BUSHELS IN STORE.": PRINT
270  IF Z=11 THEN 860
310  LET C=INT(10*RND(0)): LET Y=C+17
312  PRINT "LAND IS TRADING AT";Y;" BUSHELS PER ACRE."
320  PRINT "HOW MANY ACRES DO YOU WISH TO BUY?";
321  INPUT Q: IF Q<0 THEN 850
322  IF Y*Q<=S THEN 330
323  GOSUB 710
324  GOTO 320
330  IF Q=0 THEN 340
331  LET A=A+Q: LET S=S-Y*Q: LET C=0
334  GOTO 400
340  PRINT "HOW MANY ACRES DO YOU WISH TO SELL?";
341  INPUT Q: IF Q<0 THEN 350
342  IF Q<A THEN 350
343  GOSUB 720
344  GOTO 340
350  LET A=A-Q: LET S=S+Y*Q: LET C=0
400  PRINT
410  PRINT "HOW MANY BUSHELS DO YOU WISH TO FEED YOUR PEOPLE?";
411  INPUT Q
412  IF Q<0 THEN 850
420  IF Q<=S THEN 430
421  GOSUB 710
422  GOTO 410
430  LET S=S-Q: LET C=1: PRINT
440  PRINT "HOW MANY ACRES DO YOU WISH TO PLANT WITH SEED?";
441  INPUT D: IF D=0 THEN 511
442  IF D<0 THEN 850
445  IF D<=A THEN 450
446  GOSUB 720
447  GOTO 440
450  IF INT(D/2)<S THEN 455
452  GOSUB 710
453  GOTO 440
455  IF D<10*P THEN 510
460  PRINT "BUT YOU HAVE ONLY";P;" PEOPLE TO TEND THE FIELDS.  NOW THEN,"
470  GOTO 440
```

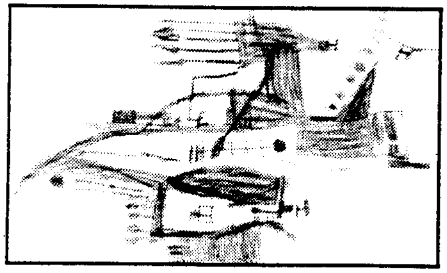
```
510  LET S=S-INT(D/2)
511  GOSUB 800
515  LET Y=C: LET H=D*Y: LET E=0
521  GOSUB 800
522  IF INT(C/2)<>C/2 THEN 530
525  LET E=INT(S/C)
530  LET S=S-E+H
531  GOSUB 800
533  LET I=INT(C*(20*A+S)/P/100+1)
540  LET C=INT(Q/20)
542  LET D=INT(10*(2*RND(0)-.3))
550  IF P<C THEN 210
552  LET D=P-C: IF D>.45*P THEN 560
553  LET Pl=((Z-1)*P1+D*100/P)/Z
555  LET P=C: LET D1=D1+D: GOTO 215
560  PRINT :PRINT "YOU STARVED";D" PEOPLE IN ONE YEAR!!!"
565  PRINT "DUE TO YOUR EXTREME MISMANAGEMENT YOU HAVE NOT ONLY"
566  PRINT "BEEN IMPEACHED AND THROWN OUT OF OFFICE, BUT YOU HAVE"
567  PRINT "ALSO SEEN DECLARED 'NATIONAL FINK!!": GOTO 990
710  PRINT "HAMURABI, THINK AGAIN.  YOU HAVE ONLY"
711  PRINT S;" BUSHELS OF GRAIN. NOW THEN,"
712  RETURN
720  PRINT "HAMURABI: THINK AGAIN. YOU OWN ONLY";A;" ACRES. NOW THEN,"
730  RETURN
800  LET C=INT(RND(0)*5)+1
801  RETURN
850  PRINT : PRINT "HAMURABI: I CANNOT DO WHAT YOU WISH."
855  PRINT "GET YOURSELF ANOTHER FLUNKY!!!!"
857  GOTO 880
860  PRINT "IN YOUR 10 YEAR TERM OF OFFICE,";P1;" PERCENT OF THE"
862  PRINT "POPULATION STARVED PER YEAR ON AVERAGE, I.E., A TOTAL OF"
865  PRINT D1;" PEOPLE DIED!!": LET L=A/P
870  PRINT "YOU STARTED WITH 10 ACRES PER PERSON AND ENDED WITH"
875  PRINT L;" ACRES PER PERSON.": PRINT
880  IF P1>33 THEN 565
885  IF L<7 THEN 565
890  IF P1>10 THEN 940
892  IF L<9 THEN 940
895  IF P1>3 THEN 960
896  IF L<10 THEN 960
900  PRINT "A FANTASTIC PERFORMANCE!!! CHARLEMAGNE, DISRAELI, AND"
905  PRINT "JEFFERSON COMBINED COULD NOT HAVE DONE BETTER!": GOTO 990
940  PRINT "YOUR HEAVY-HANDED PERFORMANCE SMACKS OF NERO AND IVAN IV."
945  PRINT "THE PEOPLE (REMAINING) FIND YOU AN UNPLEASANT RULER, AND,"
950  PRINT "FRANKLY, HATE YOUR GUTS!": GOTO 990
960  PRINT "YOUR PERFORMANCE COULD HAVE BEEN SOMEWHAT BETTER, BUT"
965  PRINT "REALLY WASN'T BAD AT ALL.";INT(P*.8*RND(0));" PEOPLE WOULD"
970  PRINT "DEARLY LIKE TO SEE YOU ASSASSINATED BUT WE ALL HAVE OUR"
975  PRINT "TRIVIAL PROBLEMS."
990  GOTO 995
995  PRINT "SO LONG FOR NOW.": PRINT
999  END
1000 PRINT %F%

RUN


TRY YOUR HAND AT GOVERNING ANCIENT SUMERIA
SUCCESSFULLY FOR A 10 YEAR TERM OF OFFICE.
```

```
HAMURABI: I BEG TO REPORT TO YOU.
IN YEAR 1, 0 PEOPLE STARVED, 5 CAME TO THE CITY.
POPULATION IS NOW 100     THE CITY N0W OWNS 1000 ACRES.
YOU HARVESTED 3 BUSHESLS PER ACRE.
RATS ATE 200 BUSHELS.
YOU NOW HAVE 2000 BUSHELS IN STORE.

LAND IS TRADING AT 21 BUSHELS PER ACRE.


HOW MANY ACRES DO YOU WISH TO BUY?
?10

HOW MANY BUSHELS DO YOU WISH T0 FEED YOUR PEOPLE?
?2000

HOW MANY ACRES D0 YOU WISH TO PLANT WITH SEED?
?   999


HAMURABI:  I BEG TO REPORT TO YOU.
IN YEAR 2, 0 PEOPLE STARVED, 3 CAME TO THE CITY.
A HORRIBLE PLAGUE STRUCK!  HALF THE PEOPLE DIED.
POPULATION IS NOW 51      THE CITY NOW OWNS 1010 ACRES.
YOU HARVESTED 1 BUSHELS PER ACRE.
RATS ATE 45 BUSHELS.
YOU NOW HAVE 1045 BUSHELS IN STORE.

LAND IS TRADING AT 23 BUSHELS PER ACRE.
HOW MANY ACRES DO YOU WISH TO BUY?
?0

HOW MANY ACRES DO YOU WISH TO SELL?
?10

HOW MANY BUSHELS D0 YOU WISH TO FEED YOUR PEOPLE?
?500

HOW MANY ACRES DO YOU WISH TO PLANT WITH SEED?
?1010
HAMURABI: THINK AGAIN.  YOU OWN ONLY 1000 ACRES.  NOW THEN,
HOW MANY ACRES D0 YOU WISH TO PLANT WITH SEED?
?1000 BUT YOU HAVE ONLY 51 PEOPLE TO TEND THE FIELDS.  NOW THEN,
HOW MANY ACRES D0 YOU WISH TO PLANT WITH SEED?
?500

YOU STARVED 26 PEOPLE IN ONE YEAR!!!
DUE TO YOUR EXTREME MISMANAGEMENT YOU HAVE NOT ONLY
BEEN IMPEACHED AND THROWN OUT OF OFFICE, BUT YOU HAVE
ALSO BEEN DECLARED 'NATIONAL FINK'!!
SO LONG FOR NOW.


SOL BASIC5

READY
```

Inverse Video in BASIC 5

```
10   REM * SOL VDM DEMONSTRATION PROGRAM
20   REM * THIS PROGRAM WILL DEMONSTRATE
30   REM * HOW TO USE THE ESCAPE SEQUENCES IN
40   REM * THE SOLOS VDM DRIVER.
50   REM * WRITTEN BY TOM DIGATE
55   REM * DECEMBER, 1977
60   REM
7J   REM * INITIALIZE THE MESSAGE DATA TABLE
80   DATA 201,238,246,229,242,244,229,228
90   DATA 160,214,233,228,229,239
91   REM * INITIALIZE VDMOUT DRIVER ADDRESS AND
92   REM * SHIFT VARIABLE FOR ESCAPE CODES.
93   V0=49177: V1=256: V2=27: V3=5
95   DIM J(30)
100  REM * CLEAR SCREEN
110  PRINT "&K": SET S=0
120  PRINT "This is a demonstration of the capabilities of"
130  PRINT "Sol Basic5 output display formatting."
135  PRINT : PRINT
140  FOR I=1 TO 14: READ J(I): NEXT
150  REM * PRINT INVERTED MESSAGE
160  PRINT "The Sol can print in upper and lower case and"
170  PRINT "can even print ";
180  FOR I=1 TO 14
190     C=J(I): GOSUB 65010
200  NEXT I
205  PRINT "."
210  REM * GET CURSOR BACK AT COLUMN 1
220  PRINT "&[&A&A"
230  PRINT : PRINT : PRINT
240  REM * DEMONSTRATE SPEED CONTROL
250  PRINT "The Sol can even change display";
260  SET S=10: PRINT " speeds";: SET S=75: PRINT " dynamically!"
270  SET S=0
280  PRINT : PRINT "Pretty neat, huh!"
285  SET S=15
290  REM * FINISH UP DEMONSTRATION
300  REM * BY SHOWING SPECIAL CHARACTER SET
301  PRINT : PRINT : PRINT "The Sol also has a special set of limited"
302  PRINT "graphics characters. They are shown below in both"
303  PRINT "regular and inverted video. This gives a Basic programmer"
304  PRINT "very good flexibility when designing display formats."
305  PRINT
306  REM * PRINT SPECIAL CHARACTER SET FOR VDM
307  SET S=0
310  FOR I=0 TO 31
320     C=I
330     GOSUB 65010
335     PRINT " ";
340  NEXT I
350  PRINT : PRINT
355  REM * PRINT SPECIAL CHARACTER SET FOR VDM IN INVERTED VIDEO
360  FOR I=0 TO 31
370     C=I+128: GOSUB 65010: C=160: GOSUB 65010
400  NEXT
410  PRINT
420  SET S=18
```

```
430  PRINT "This concludes my demonstration of special display"
440  PRINT "features available in BASIC5. Happy computing!!!"
450  PRINT : PRINT : PRINT : SET S=0
460  END

64998 REM ****************************************************
64999 REM *
65000 REM * OUTPUT INVERSE VIDEO SUBROUTINE USING SOLOS VDM
65001 REM * PASS DECIMAL VALUE OF CHARACTER
65003 REM * IN VARIABLE 'C'. NOTE- 'B' IS DESTROYED.
65004 REM *
65005 REM ****************************************************
65006 REM *
65010 B=ARG(V2*V1): B=CALL(V0): B=ARG(V3*V1): B=CALL(V0)
65020 B=ARG(C*V1): B=CALL(VO): RETURN
65025 REM *
65030 REM * THE ROUTINE ABOVE USES THE FOLLOWING VARIABLES
65040 REM * V0   =ADDRESS OF VDM DRIVER IN SOLOS
65050 REM * V1   =256 TO SHIFT CHARACTER INTO B REGISTER
65060 REM * V2   =27 FOR ESCAPE CHARACTER
65070 REM * V3   =05 FOR ESCAPE CODE TO SEND CHAR. TO SCREEN
65080 REM *               WITHOUT CHANGING HIGH ORDER BIT.
```



Star Ship, by Brian Marsh, age 6

PROCESSOR TECHNOLOGY
ACCESS.

7100 Johnson Industrial Drive
Pleasanton, CA 94566