PROCESSOR TECHNOLOGY

ACCESS.

75C per issue

Published by Wible/Rampton Advertising, San Francisco. Volume One, Number One, February 1977

A Letter from the Editor

"The flowers, the gorgeous, mystic multi-colored flowers are not the flowers of life, but people, yes people are the true flowers of life: and it has been a most precious pleasure to have temporarily strolled in your garden."

Lord Buckley

No, this isn't a poetry journal or philosophical tract, but we do feel that Lord Buckley elucidates the concept behind the Processor Technology newsletter: its contents will be the kind of information that allows you to expand the creative applications of your personal computing system.

ACCESS is dedicated to the creation of a personal dialog between us at PTC and you, the people using our products, because we believe that it's important for you to understand your present system well enough to use it effectively, before you become trapped in the "bigger means better" syndrome.

Hence there *will* be no advertising of any kind in ACCESS, and no thinly disguised pitches aimed at getting you to part with your hard earned money for the unneccessary duplication of existing hardware.

Each issue will contain articles, engineering/application bulletins, and software (PROGRAM) listings which we think will help you get more fun out of owning your computer.

All we expect from you in return is what's known in technical and psychological circles as feedback. If you have devised a neat hardware modification or written a nifty program, send it to us and you'll get no financial remuneration. You'll just get the glory of seeing your name in print. Very spiritual, eh what?

Aram Attarian II

Subscription Information

ACCESS will be published every six weeks. This first issue is being sent free to all of our customers. 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, 6200 Hollis Street, Emeryville, CA 94608.

One to One Communication

Open two-way communication is our goal, and it would be great to have unlimited time to rap with you over the phone. But we can't manage that logistically, so we are instituting procedures that will get your questions answered efficiently, without eliminating the personal touch.

You can get through to our engineers and technical staff any time between 9:30 and noon or 1:30 and 4, Monday through Friday. The number is (415) 652-8080. They'll try to answer your technical questions and provide more detailed product information than you may glean from our literature. It helps a lot if you think out your questions before you call, maybe make a few notes. It helps, too, if you have the manual and/or equipment handy to the phone. The easier it is for you to figure out exactly what we're talking about (and vice versa), the more help we can give in one phone call and the more phone calls we can handle. We'll also be happy to answer your questions if you drop us a letter.

There are two or three situations in which calling us isn't the fastest way to get help. If you have a problem with a **Processor Technology product which you purchased from a local dealer,** try your dealer first. They're all authorized to service the products they sell, precisely because they can give you faster and more personalized service than anybody can by mail. Our dealers all have diagnostic tapes available now, so if you need a PTC product tested it can be handled on the spot.

If you need a defective part replaced, just mail it to us with a note of explanation. There's nothing we can do about it over the phone anyway, so you might as well save yourself the time. (Naturally, it helps if your explanation is as clear and concise as possible. but we can't all he Hemingways.) If you need service that isn't covered by your warranty, please enclose a check for the $20.00 minimum service charge. If the cost exceeds that (heaven forbid!) you will be notified before we proceed with the service.

If you do send in one of our products for repair, **please send only the defective board.** For obvious reasons, we cannot be responsible for the care and feeding of your monitors, keyboards or other products from other manufacturers.

MATCHMAKING-Software Division

Since most important programs are written for a particular system, modifications are sometimes in order to improve compatibility with your system. We'll be passing on more good ideas as we come across them; here are two dealing with MITS BASIC and the VDM.

MITS 12K Extended Disk and the VDM

If you don't have a 3P+S interface, you can still use the BASIC program in your VDM manual to link MITS 12K Extended Disc to your VDM driver. Simply change these six lines in the listing of VDM-1 to BASIC link appearing on page AV-18 of your VDM manual, and then run the program as explained there. Change the statements to read as follows:

```
0106 FOR Y=L TO L+419
0140 POKE G,195:POKE G+1,0:POKE G+2,P:POKE G+3,0
0160 DATA195,151,1001,195,156,1001,245,229,213,197
0162 DATA205,58,1000,193,209,195,147,1001,245,58
0232 DATA0,0,6,225,241,193,201,254,26,195,142
0234 DATA13,254,127,202,142,13,195,46,1000
```

MITS 8K or 12K BASIC and the VDM

If you have I/O ports that are assigned 20 and 21 instead of 0 and 1, you'll need to make five statement changes in the VDM-1 to BASIC link program in your VDM manual. The listing appears on pp. AV-17 to AV-19 of the manual, and should be modified as follows:

```
0078 IF A=219 AND 11=16 GOTO 92
0080 IF A=219 AND B=17 THEN I=K:GOTO 106
0152 DATA219,255,31,210,13,1000,219,16,230,2000
0224 DATA201,219,16,230,6000,201,58,140,1001,254
0228 DATA201,219,17,230,127,254,1,202,46,1000
```

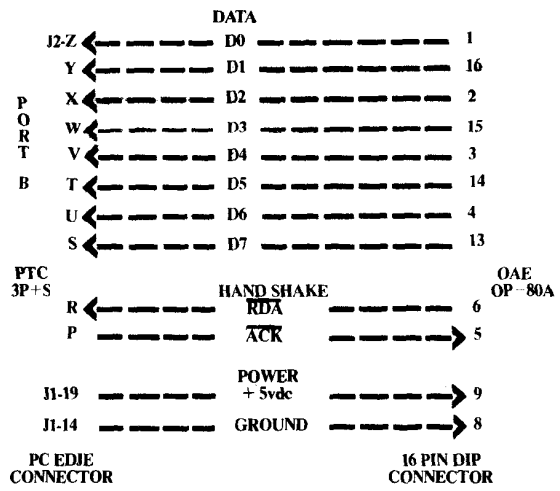
MATCHMAKING-Hardware Division

When you're combining equipment from different manufacturers, optimum performance often depends on specific information about the idiosyncracies of their connections. Owner's manuals can't cover every permutation and combination, so think of this feature as a running supplement that gets down to cases.

3P+S and the OP 80A

We've had a lot of inquiries about using the OP-80A high speed manual punchpaper tape reader with the 3P+S, so we asked Oliver Audio Engineering for the straight scoop on interconnecting it with the 3P+S interface. Note that the OP-80A acknowledge line must be programmed for a negative true (ack) signal. We've also heard from many of our customers that the use of black tape and rolled tape (instead of fanfold) produces more accurate data transfer, using the OP-80.

3P+S/OP-80A INTERFACE



Announcing the SOL USERS GROUP!

The Sol Users Group was recently organized by members of the Homebrew Computer Club in Palo Alto; SUG is not affiliated in any way with Processor Tech-nology. The purposes of SUG are to exchange software and other applications, and to create standards. If plenty of interest is shown, a Sol Newsletter will be published and sent to members.

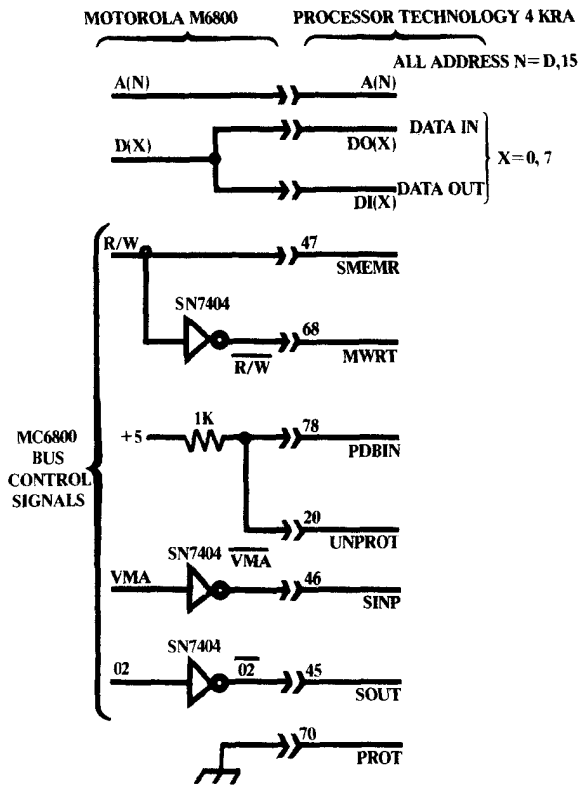
If you own or have ordered a Sol, send your name, address, phone number and ideas to:

Bill Burns
4190 Maybell Way
Palo Alto, CA 94306
(Please don't call.)

PT 4KRA Memory Boards and the Motorola M6800

The following diagram should help you expand your M6800 system with minimum headaches. Note that the M6800 can drive 2 4KRA cards at reduced clock rates without bus drivers; however, full buffering is recommended.

CONNECTING PROCESSOR TECHNOLOGY 4KRA MEMORY CARD FOR USE IN MOTOROLA M6800 SYSTEMS



VDM-1 and the 6800 Microprocessor

The VDM-1 can live quite happily with your 6800 if a small amount of signal processing is performed externally.

Signals which may be directly used by the 6800 system are:

ADR0-ADR15 High-active address

DO0-DO7, DI0-DI7 High-active data lines. They may be connected in parallel to form a bi-directional data bus if necessary.

Signals which require conversion or re-interpretation:

PSYNC Connect to high-active VMA (Valid Memory Address)

Φ2 Connect to Φ1 clock

Pin 4 of IC 18 (74LS132) Break connection to Pin 3 and connect to Pin 2

SINP, SOUT Connect to the highest-order address bits which are "1" when registers are addressed. (Bits 14 and 15 are suggested.) Presence of a "1" on either line will cause the address decoder to switch its comparison to to the status port address. The status port will therefore respond to any address whose top six bits are set by jumpers (see your VDM-1 manual), and whose bits 8 and 9 are zero. The low-order 8 bits are not decoded during status port response.

PWR Externally generated signal consisting of VMA • Φ2 • R/W

MWRITE Inverted PWR

PDBIN Externally generated signal consisting of VMA • Φ2 • R/W

XRDY, PRDY Not used

Note that +8V, +16V and -16V are still required to operate the board, and that 6800 systems designed for all 5-volt operation may have to be augmented. Simple unregulated supplies will perform well, but care should be taken not to exceed +10V on the +8V line to avoid excessive dissipation in the +5V regulator.

What's the Best Monitor for your Sol or VDM-1?

That's one of the questions we're asked most frequently. We always recommend a black & white monitor designed for use with closed circuit TV systems or videotape recorders. Check your local Processor Technology dealer for good sources; they're up on that sort of thing.

If you want to use a regular black & white TV, try for a solid state model with an isolation transformer. In either case, remember that you don't have to spend a lot of money to get a decent picture, so let the rest of the family enjoy that big screen super color set.

PROCESSOR TECHNOLOGY
ACCESS.

Editor: Aram Attarian II
Publisher: Wible/Rampton Advertising,
727 15th Avenue, San Francisco, CA 94118
ACCESS is published approximately every six weeks. Subscription rate: $4 per year, from Processor Technology Corp., 6800 Hollis St., Emeryville, CA 94608.

Interfacing a Keyboard and VDM with ALS-8, using the 3P+S.

To get your keyboard connection up and running, you'll need to connect 7 data lines, the keyboard strobe, and ground to the J2 connector of your 3P+S. Figure 1 shows the typical keyboard connections.

USING A KEYBOARD AND THE VDM WITH THE ALS-8

KEYBOARD INTERFACE

TYPICAL KEYBOARD CONNECTIONS

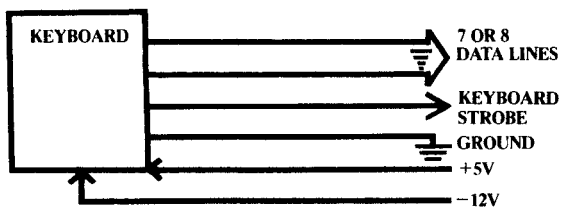


Fig. 1

The keyboard strobe. This circuit tells the processor when a key has been pressed. You'll need to install an additional IC (preferably in a 16 pin DIP socket) in the unused IC pad in the lower right corner of your 3P+S, and connect it to pin 7 of J2. It's used to condition the strobe. Use a 74LS109 or 74109. (Figure 2)

STROBE CONDITIONING CIRCUIT

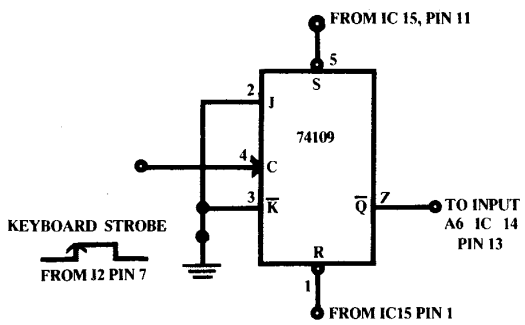


Fig. 2

To wire the 74109:

1) Connect pins 2 and 3 to ground.
2) Pins 16 (VCC) and 8 (GND) are already connected on board.
3) Connect pin 1 to pin 1 of IC 15 to provide a pull-up connection.
4) Connect pin 5 to pin 11 of IC 15 to reset the flip-flop when data has been accepted.
5) Connect pin 7 to a point on 3P+S leading to J2 pin 12. This point will go low when the strobe occurs.
6) Connect pin 4 to a point on 3P+S leading to J2 pin 7, the point where the strobe from the keyboard will be connected.

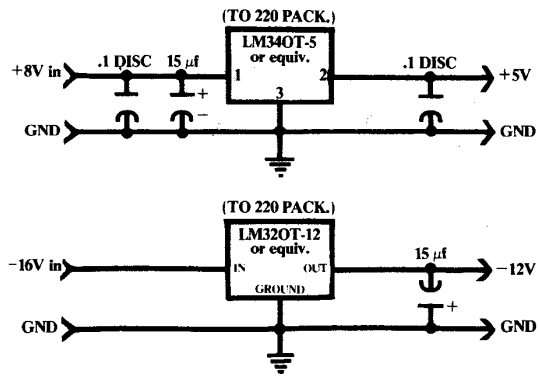
The data lines. Either seven or eight data lines are used to transmit the ASCII code for the key being pressed to the computer. They're connected to the B port inputs of the 3P+S in the following manner:

Keyboard Signal	3P+S J2 Connector	Data Bit
Bit 1	Z	D0
Bit 2	Y	D1
Bit 3	X	D2
Bit 4	W	D3
Bit 5	V	D4
Bit 6	U	D5
Bit 7	T	D6
Bit 8*	S	D7

*Pin J2-S is left unconnected if your keyboard doesn't have an eighth bit.

Ground, +5V, -12V. These provide power for the circuitry of the keyboard. Ground is simply connected to pin 12 of J2. +5V and -12V (regulated) should be provided by a separate power supply. Some KYBDS only require +5V Figure 3 shows an example of one, assembled on a small piece of perf board and attached to the keyboard or main frame. Jumper it to the keyboard connector.

ASSEMBLE ON SMALL PIECE OF PERF BOARD AND ATTACH TO KYBD OR MAINFRAME AND JUMPER TO KYBD CONNECTOR



Jumper areas. Only two are important for the keyboard interface: Area A must be jumpered for address 0 (i.e., all to ground), and Area B must be jumpered from left to center, to set port A at 0 (status) and port B at 1 (data) to correspond to Processor Technology software.

Testing the interface with ALS-8.

1) Turn on the computer and examine location E000H. When you hit Run, the address lights should look like this:

```
A15 A14 A13 A12 All A10 A9 A8
 •   •   o   •   o   o   o  •
```

• = Light on o = Light out

2) Be sure the keyboard is sending upper case characters. Type EXEC E024, then hit the carriage return key. Address light A13 should blink instantaneously at this point.
3) Type IODR/SYSIO/0 FE77, then hit the return key. Again, address light A13 should blink.
4) Now hit Space, Space, Return; wait a moment and then hit Control Z. The screen should now be blank.
5) Hit Control S. The word SPEED? should now appear in the lower left corner of the screen. If it doesn't, you may have to adjust the horizontal and vertical controls on the VDM to get this message onto the visible portion of the screen.
6) Type 1; you should get an automatic carriage return, then type DUMP 0 FFFF and hit Return. Memory will now dump on the VDM display. You can hit the Escape key to stop the dump.
7) Now you're ready to start programming with the ALS-8! Try some of the examples from the ALS-8 manual to familiarize yourself with its operation.

Hoare's Law of Large Programs

Inside every large program is a small program struggling to get out.

Good News for our Customers in Europe: VDM-1 and the European 50 Hz Standard

The European television standard maintains the same horizontal rate as the U.S. (15,750 Hz), but it defines a raster of 625 lines at a field rate of 50 Hz. The effect is to increase the number of scan lines on the screen.

It's quite easy to modify your VDM-1 to work on this standard. Simply disconnect pin 5 of IC 8 from pin 6, and reconnect it to ground (pin 4). This increases the modulus of the counter to 8 in the VDISP time, resulting in 4 extra character lines (52 scan lines) on the display. The total is now 312 scan lines per field, the equivalent of 624 per frame.

The field rate should be close enough to 50 Hz to reduce any swim effects to less than 0.1 Hz. You may have some trouble centering the display in the frame, because the standoff time to VSYNC from the bottom of the display is still on the 60 Hz standard. If the effect is objectionable, increase Resistor R 34 in series with the VPOS control.

Rule of Accuracy

When working toward the solution of a problem it always helps you to know the answer.

Sattinger's Law

It works better if you plug it in.

Newett Awl's Choo Choo Train

Or, Idle Fantasies on a VDM Screen

Once upon a time, in a curious little place, there was a Sol system and a programmer sitting around with nothing to do. So Newett Awl decided to tell his computer a bedtime story, and VDM-1 helped him out by drawing the pictures.

Try it on your system. Ol' Uncle Sol makes a great babysitter.

```
0000             0000
0000             0010
0000             0020
0000             0030
0000             0040
0000             0050
0000             0060
0000             0070
0000 AF          0071 TRAIN   XRA    A        ONCE UPON A TIME,
0001 D3 C8       0072         OUT    0C8H     IN A CURIOUS LITTLE PLACE
0003 21 84 01    0080         LXI    H,SHED   THERE WAS
0006 01 00 08    0090         XSI    B,2048   ..A TINY TRAIN
0009 36 20       0100 EMPTY   MVI    M,20H    ..AND IT STAYED IN A
000B 23          0110         INX    H        ..TINY SHED
000C 0B          0120         DCX    B        ..THAT WAS ALL EMPTY
000D AF          0130         XRA    A
000E A8          0140         XRA    B
000F C2 09 00    0150         JNZ    EMPTY
0012 21 F7 02    0160         LXI    H,CLOUD  .. EXCEPT FOR A HUGE
0015 11 00 00    0170         LXI    D,0
0018 06 0F       0180         MVI    B,15 B
001A CD 50 00    0190         CALL   SMO1   I
001D 11 2B 00    0200         LXI    D,43    L
0020 06 0D       0210         MVI    B,13     L
0022 CD 50 00    0220         CALL   SMO1      O
0025 11 30 00    0230         LXI    D,48       W
0028 06 09       0240         MVI    B,9          Y
002A CD 50 00    0250         CALL   SMO1
002D 11 34 00    0260         LXI    D,52          C
0030 06 04       0270         MVI    B,4            L
0032 CD 50 00    0280         CALL   SMO1             O
0035 11 3B 00    0290         LXI    D,59              U
0038 06 02       0300         MVI    B,2                 D
003A CD 50 00    0310         CALL   SMO1
003D 11 3C 00    0320         LXI    D,60                O
0040 06 01       0330         MVI    B,1                  F
0042 CD 50 00    0340         CALL   SMO1
0045 11 3E 00    0350         LXI    D,62                   S
0048 06 01       0360         MVI    B,1                     M
004A CD 50 00    0370         CALL   SMO1                      O
004D C3 59 00    0380         JMP    ENGIN                       K
0050             0390 .                                          E
0050 19          0400 SMO1    DAD    D
0051 36 6F       0410 SMO2    MVI    M,SMOKE   .. COUGH
0053 23          0420         INX    H
0054 05          0430         DCR    B              .. COUGH
0055 C2 51 00    0440         JNZ    SMO2
0058 C9          0450         RET    .      COMING OUT OF THE STACK OF
0059             0460 .
0059 21 A6 04    0470 ENGINE  LXI    H,SHED+322H  A TINY LOCOMOTIVE
005C 36 16       0480         MVI    M,16H  WITH A LITTLE SMOKESTAK
005E 21 A9 04    0490         LXI    H,SHED+325H  AND A LITTLE BELL
0061 36 07       0500         MVI    M,07H
0063 21 AB 04    0510         LXI    H,SHED+327H  ..AND A TINY DOME
0066 36 6E       0520         MVI    M,6EH
0068 2A 66 01    0530         LHLD   CAB1   ..AND A CAB
006B 22 AE 04    0540         SHLD   SHED+32AH WITH WINDOWS
006E 2A 68 01    0550         LHLD   CAB2   .. SO THAT YOU COULD
0071 22 B0 04    0560         SHLD   SHED+32CH SEE INTO WHERE THE
0074 2A 6A 01    0570         LHLD   CAB3   .. ENGINEER AND THE
0077 22 B2 04    0580         SHLD   SHED+32EH FIREMAN SAT.
007A 2A 6C 01    0590         LHLD   BOI1   .. IT HAD A BEAUTIFUL
007D 22 E5 04    0600         SHLD   SHED+361H POLISHED
0080 2A 6E 01    0610         LHLD   BOI2   .. BRASS
0083 22 E7 04    0620         SHLD   SHED+363H  BOILER
0086 2A 70 01    0630         LHLD   BOI3   ..  WITH
0089 22 E9 04    0640         SHLD   SHED+365H  THE NUMBER
008C 22 EB 04    0650         SHLD   SHED+367H  "99" ON THE
008F 2A 72 01    0660         LHLD   BOI4   ..  SIDE OF
0092 22 ED 04    0670         SHLD   SHED+369H  THE CAB
0095 2A 74 01    0680         LHLD   BOI5   ..  BUT YOU COULDN'T
0098 22 EF 04    0690         SHLD   SHED+36BH SEE EITHER THE
009B 2A 76 01    0700         LHLD   BOI6   ..  ENGINEER OR THE
```

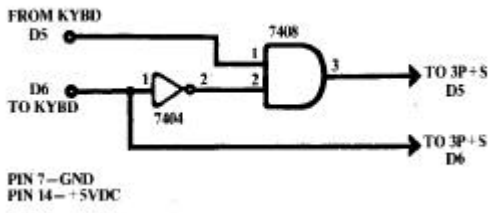
(continued on page 11)

Ups and Downs
or How to Type in Upper Case Only without Shifting

If you have a keyboard with both upper and lower case operation, the frequent shifts are a pain when you're entering alpha-numeric data. Here's a simple cure that sets data bit D5 low when a lower case alpha-betic character is output from the keyboard, thus presenting it to the computer as upper case.

You'll need two chips, either a 7404 and a 7408 or a 74LS04 and a 74LS08. (Either pair works fine.) Install them on a small piece of perf board as shown in the diagram, and tie all unused inputs to +5V.

The pins you'll be using on the 04 chip are 3, 5, 9, 11, and 13; on the 08 chip use 4, 5, 9, 10, 12, and 13.



FLASH ~~ VDM Access Flicker Eliminated

That blasted flicker you get whenever the VDM memory is being frequently accessed occurs because the screen is blanked for a very short time whenever the processor reads or writes to memory. You can eliminate it by modifying the timing system so than VDM memory is accessed only when the beam of the picture tube is off the visible portion of the screen. The VDM has a timing signal that indicates this condition, and you can use it to synchronize access to display memory.

Connect a jumper wire from pin 13 of IC 39 to pin 5 of IC 39. This hooks up an unused section of IC 39 to DI bit 1. Connect a wire from IC 39 pin 14 to IC 15 pin 9. (This allows programs which access the VDM memory to use the timing signal.) Now when C8 is input, bit 1 will be low whenever the display memory can be accessed.

We'll explore the implications further in the next issue, and provide an example program or two.

Murphy's Law of Thermodynamic
Things get worse under pressure.
Lowery's Law
If it jams—force it. If it breaks, it needed replacing anyway.

Application News
Ham Computer Based on Sol Terminal System

SYSTEM 4000 ham computer was developed by Curtis Electro Devices, Inc., Box 4090, Mountain View, Ca. 94040. The company makes ham keyers and an industrial line of PROM programmers. We asked the president, John G. Curtis, to comment on working with the Sol system; he did all of the hardware and software work himself, developing programs for the Sol Terminal on ALTAIR with ALS-8, VDM 1, 40K of RAM, 3P+ 1, Tarbell cassette system, Bytesaver, ASR-33 TTY, and Motorola video monitor. Jack's report was glowing:

"I had no previous experience or trainin software development. It was all learned on the fly from books and practical experience. (Try, try and try again!!!) . . . Not too much assistance was required. The ALS-8 is easy to use and the Simulator program is absolutely essential. The Editor is also absolutely necessary. (Everything is necessary!!)...

"The Sol went together with a minimum of effort and trouble even though it was one of the first units actually put on the line. There were things which didn't work but with the help of PTC (especially Bob Marsh and Aram) things were put in order in a hurry.

"In my opinion, the Sol terminal is the most ideal small computer system available today (certainly at that price) and every needed feature is there. For our purposes it was absolutely ideal from every standpoint. We are now able to get the computer into the hamshack on a commercial basis. This has been predicted for sometime, now it has happened."

System 4000 Ham Computer

The SYSTEM 4000 is a full scale standard desk top minicomputer specially equipped with firmware programs and interfaces for the amateur radio operator. Being a stand-alone computer, it can also run business, educational, scientific or games programs.

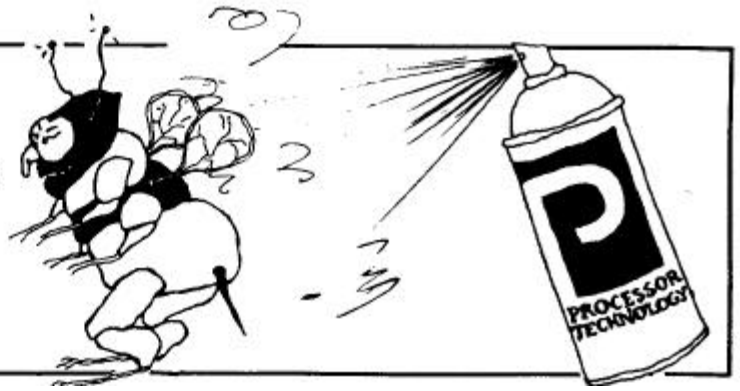
The SYSTEM 40000 is derived from the Sol terminal Computer and can take advantage of Processor Technology's programs and accessories. System 4000 uses the standard S-100 bus for plug-in accessory boards, and the owner can add or exchange PROM integrated circuits to update the system. Features:

Morse reader, capable of receiving code at speeds of 6-250 WPM (or higher). CRT or TTY output, selectable for upper and lower case.
Paddle keyer, with dot= and dash memory, iambic and full self completion. CRT or TTY printout.
Keyboard keyer, sends Morse from keyboard. CRT or TTY output.
ASCII terminal, half or full duplex.

Complete details can be obtained by writing to: Curtis Electro Devices, Inc., P O. Box 4090, Mountain View, Ca. 94040.

Murphy's Third Law
In any field of scientific endeavor, anything that can go wrong will go wrong.
Sevarenid's Law
The chief cause of problems is solutions.
Brooks's Law
Adding manpower to a late software project makes it later.
O'Tooles's Commentary on Murphy's Laws
Murphy was an optimist.

Bug Squad

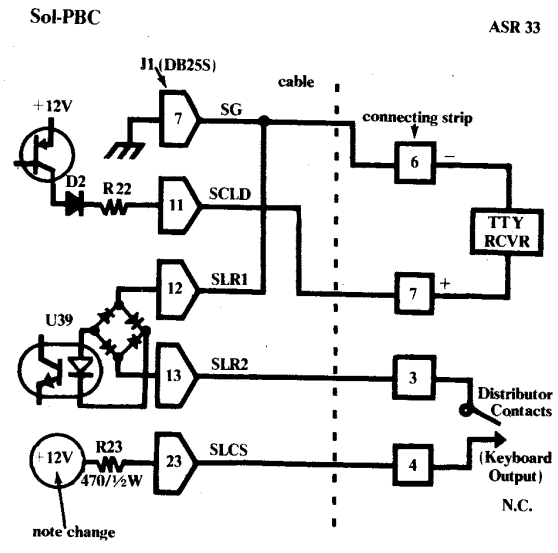


A major purpose of this newsletter is helping you stomp out the bugs that inev-itably occur even in the best of systems. Bug Squad will be a regular feature. We'll tackle the problems we've encountered most frequently, and you're also invited to send in any problems that have been bugging you. Send solutions, too, if you've found them. Share the wealth.

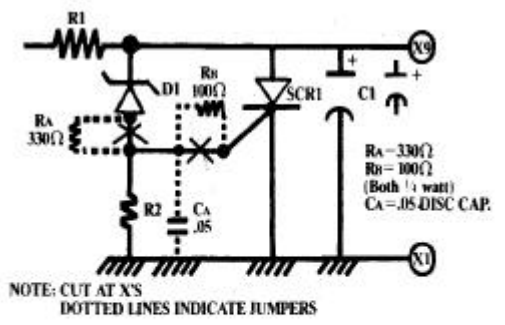
For starters, here's how to fix a few of the bugs that crept into early Sol systems:

Memory Protect/Unprotect Lines
***The bug*: Erratic behavior when early Sol-PC, Sol-10, and Sol-20 units are used with S-100 bus compatible memory modules.**
The squasher: On early Sol circuit boards, the protect (pin 70) and unprotect (pin 20) lines are floating. Simply ground bus line 70 on the Sol PCB itself to disable the memory protect signal.

Current Loop Source
***The bug*: R23, a 470 ohm 1/2w resistor, is incorrectly tied to +5 on early Sol PC boards, producing less than a full 20mA current to teletype connections.**
The squasher: Reconnect R23 to + 12V which is only 1/4" away. Check to make sure that R29 remains connected to +5V. We suggest that the 20mA current loop connections (such as for ASR33) be made as follows:



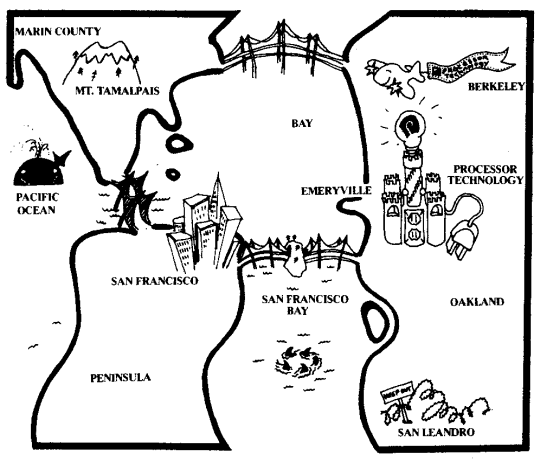
Accidental Avalanche
***The bug*: On some Sol Power Supply REG Boards, the SCR1 (MCR 106-2) in the +5 volt regulated supply will intermittently go into the avalanche mode. This draws sufficient current to drop the +5 volt to about +1V This modification will prevent the accidental triggering, but will retain the overvoltage protection of the crowbar circuit.**
The squasher: Make the modification here by adding the components shown and cutting appropriate traces (as shown).



Law of Selective Gravity (The Buttered Side Down Law)
An object will fall so as to do the most damage.

Where the %×&*# is Emeryville?

Emeryville, Ca., is located at the foot of the Oakland Bay Bridge, squeezed between Oakland and Berkeley. Neither wishes to claim it. Emeryville consists largely of mudflats and factories, the principle landmarks being a highrise apartment complex called Watergate (no relation), a shoreline assemblage of driftwood sculptures, and Processor Technology. Before the rise of Processor Technology, the main industry was legal gambling clubs. We, however, plan to put Emeryville on the map.



Consol Source Listing

For those of you who haven't seen it yet, here's the source list for the minimum Sol operating System, CONSOL. It provides all necessary display routines, along with standardized calling points for input/output operations.

```
-== CONSOL ==-
COPYRIGHT 1976



C000                0001 *
C000                0002 *
C000                0003 *
C000                0004 *                    *****              *
C000                0005 *                   *                   *
C000                0006 *                   * * *       **      *
C000                0007 *                       *      *  *     *
C000                0008 *                   *****       **      ******
C000                0009 *
C000                0010 *                        SYSTEM SOLFTWARE
C000                0011 *
C000                0012 *
C000                0013 *                    <<>>     CONSOL     <<>>
C000                0014 *
C000                0015 *
C000                0016 *      VERSION: 1.0:5
C000                0017 *      RELEASE: 10.11.76
C000                0018 *
C000                0019 *
C000                0020 *
C000                0021 *
C000                0022 *
C000                0023 *
C000                0024 *
C000                0025 *      COMMANDS ARE ALSO AVAILABLE TO ENTER DATA TO MEMORY
C000                0026 *  AND TO DISPLAY MEMORY DATA ON THE SCREEN.  AN EXECUTE
C000                0027 *  COMMAND IS PROVIDED TO ALLOW PROGRAM EXCUTION OUTSIDE
C000                0028 *  OF THE CONSOL PROM AND PROVISION IS MADE FOR CASSETTE
C000                0029 *  TAPE LOAD OF PROGRAMS OR DATA.
C000                0030 *
C000                0031 *
C000                0032 *  NOTE: CONSOL,SOLOS AND SOLED ARE REGISTERED TRADEMARKS
C000                0033 *        OF:
C000                0034 *              PROCESSOR TECHNOLOGY CORP.
C000                0035 *                  EMERYVILLE,CALIF
C000                0036 *
C000                0037 *    THE FOLLOWING CODE IS THE PROPERTY OF PROCESSOR
C000                0038 *  TECHNOLOGY CORP.  IT IS DISTRIBUTED ON A 'PERSONAL USE
C000                0039 *  BASIS FOR THE BENIFIT OF SOL SYSTEM OWNERS.  ALL FORMS
C000                0040 *  OF THE CODE ARE COPYRIGHT 1976 BY PROCESSOR TECHNOLOGY
C000                0041 *  AND ALL RIGHTS THERIN ARE RESERVED.
C000                0042 *
C000                0043 *
C000                0044 *
C000                0045 *   AUTO-STARTUP CODE
C000                0046 *
C000 00             0047            DB      0           FOUR PHASE WONDER
C001 C3 65 C2       0048 BGIN       JMP     STRTA
C004                0049 *
C004                0050 *
C004                0051 *
```

```
C004                    0052 *              ----=  SoL SYSTEM I/O ROUTINES =-----
C004                    0053 *
C004                    0054 *   THE FOLLOWING CODE IS STANDARDIZED FOR ALL SoL SYSTEM
C004                    0055 *   SOLFTWARE IT PROVIDES COMMON ENTRY POINTS FOR INPUT AND
C004                    0056 *   OUTPUT OPERATIONS.  CONSOL DOES NOT HAVE PROVISION FOR
C004                    0057 *   PARALLEL I/O OPERATIONS BECAUSE OF SPACE LIMITATIONS.
C004                    0058 *
C004                    0059 *
C004                    0060 *
C004                    0061 *
C004                    0062 *            JUMP TABLE INPUT/OUTPUT ROUTINES
C004                    0063 *
C004                    0064 *   THIS ROUTINE OUTPUTS THE CHARACTER IN REGISTER 'B' TO
C004                    0065 * THE OUTPUT DEVICE POINTED TO BY THE CURRENT OUTPUT SELECT
C004                    0066 * REGISTER.  THE DEVICES ARE DEFINED AS FOLLOWS:
C004                    0067 *
C004                    0068 *       0 - VDM SCREEN
C004                    0069 *       1 - SERIAL OUTPUT PORT
C004                    0070 *       2 - PARALLEL OUTPUT PORT (NOT AVAILABLE ON CONSOL)
C004                    0071 *       3 - ERROR HANDLER
C004                    0072 *
C004                    0073 *   ENTRY AT:  SOUT SELECTS CURRENT OUTPUT DEVICE
C004                    0074 *              AOUT SELECTS DEVICE IN REGISTER 'A'
C004                    0075 *
C004 3A 03 C8           0076 SOUT     LDA      OPORT       GET PORT NUMBER FROM MEMORY LOCATION
C007 E6 03              0077 AOUT     ANI      3           KEEP IT IN CONTROL
C009 E5                 0078          PUSH     H           WE'LL RESTORE IT LATER
C00A 21 84 C1           0079          LXI      H,OTAB      POINT TO TABLE
C00D 07                 0080          RLC      .           COMPUTE ADDRESS
C00E 85                 0081          ADD      L
C00F 6F                 0082          MOV      L,A         WE HAVE IT
C010 C3 71 C0           0083          JMP      DISPT       GO TO HL....
C013                    0084 *
C013                    0085 *
C013                    0086 *
C013                    0087 *   THIS ROUTINE INPUTS A CHARACTER TO REGISTER 'A' FROM
C013                    0088 * THE CURRENT INPUT DEVICE POINTED TO BY THE CURRENT INPUT
C013                    0089 * SELECT REGISTER.
C013                    0090 *
C013                    0091 *   ENTRY POINTS ARE DEFINED:
C013                    0092 *
C013                    0093 *        0 - KEYBOARD INPUT
C013                    0094 *        1 - SERIAL INPUT
C013                    0095 *        2 - PARALLEL INPUT (NOT AVAILABLE ON CONSOL)
C013                    0096 *        3 - ERROR HANDLER
C013                    0097 *
C013 3A 04 C8           0098 SINP     LDA      IPORT       GET PORT NUMBER FROM MEMORY LOCATION
C016 E6 03              0099 AINP     ANI      3           WE MUST BE REASONABLE
C018 E5                 0100          PUSH     H           SAVE H&L
C019 21 8C C1           0101          LXI      H,ITAB      POINT TO TABLE
C01C 07                 0102          RLC      .           THE MATH
C01D 85                 0103          ADD      L
C01E 6F                 0104          MOV      L,A         DONE
C01F C3 71 C0           0105          JMP      DISPT       WE HAVE THE ADDRESS ... GO TO HEAVEN
C022                    0106 *
C022                    0107 *
C022                    0108 *       KEYBOARD INPUT STATUS CHECK
C022                    0109 *
C022                    0110 *  THIS ROUTINE TESTS THE KEYBOARD STATUS AND RETURNS
C022                    0111 *  WITH THE TEST BITS SET.
C022                    0112 *
C022 DB FA              0113 KSTAT    IN       STAPT       GET STATUS WORD
C024 E6 01              0114          ANI      KDR         TEST KEYBOARD BIT
C026 C9                 0115          RET      .           FLAGS ARE SET
C027                    0116 *
C027                    0117 *
C027                    0118 *        KEYBOARD DATA INPUT
C027                    0119 *
C027                    0120 *  THIS ROUTINE, ENTRY AT KREAD, GETS THE DATA FROM THE
C027                    0121 *  KEYBOARD.  ON RETURN THE CHARACTER IS IN REGISTER 'A'.
C027                    0122 *
C027 E1                 0123 KREA1    POP      H           JUMP TABLE ENTRY POINT
C028 CD 22 C0           0124 KREAD    CALL     KSTAT       CHECK STATUS
C02B C2 28 C0           0125          JNZ      KREAD       WAIT FOR INPUT
C02E DB FC              0126          IN       KDATA       GET DATA
C030 C9                 0127          RET      .           GO BACK WITH IT
C031                    0128 *
C031                    0129 *
C031                    0130 *   SERIAL INPUT STATUS CHECK
C031                    0131 *
C031 DB F8              0132 SSTAT    IN       SERST       GET SERIAL STATUS WORD
C033 E6 40              0133          ANI      SDR         TEST FOR SERIAL DATA READY
C035 C9                 0134          RET      .           FLAGS ARE SET
C036                    0135 *
C036                    0136 *
C036                    0137 *    SERIAL DATA INPUT
C036                    0138 *
C036 E1                 0139 SREA1    POP      H           RESTORE HL FROM JUMP TABLE ENTRY
C037 CD 31 C0           0140 SREAD    CALL     SSTAT       NORMAL ENTRY POINT
C03A CA 37 C0           0141          JZ       SREAD       WAIT FOR INPUT
C03D DB F9              0142          IN       SDATA       GET DATA BYTE
C03F C9                 0143          RET      .           WE HAVE IT
C040                    0144 *
C040                    0145 *
C040                    0146 *    SERIAL DATA OUTPUT
C040                    0147 *
C040 E1                 0148 SEROT    POP      H           JUMP TABLE ENTRY POINT
C041 DB F8              0149 SDROT    IN       SERST       GET PORT STATUS
C043 17                 0150          RAL      .           PUT HIGH BIT IN CARRY
C044 D2 41 C0           0151          JNC      SDROT       LOOP UNTIL TRANSMITTER BUFFER IS
EMPTY
C047 78                 0152          MOV      A,B         GET THE CHARACTER BACK
C048 D3 F9              0153          OUT      SDATA       SEND IT OUT
C04A C9                 0154          RET      .           AND WE'RE DONE
C048                    0155 *
C04B                    0156 *
C048                    0157 *
C048                    0158 *
C04B                    0159 *
C04B                    0160 *                VIDEO DISPLAY DRIVER ROUTINES
C04B                    0161 *
C048                    0162 *
C04B                    0163 *  THESE ROUTINES ALLOW FOR STANDARD VIDEO TERMINAL
C04B                    0164 *  OPERATIONS.  ON ENTRY, THE CHARACTER FOR OUTPUT IS IN
C04B                    0165 *  REGISTER B AND ALL REGISTERS ARE UNALTERED ON RETURN.
C04B                    0166 *
C04B                    0167 *     THE 'CONSOL' VERSION OF THIS ROUTINE IS A MINIMUM
C04B                    0168 *  IMPLEMENTATION OF ROUTINES ORIGINATED BY:
C04B                    0169 *
C04B                    0170 *               IAN KETTLEBOROUGH
C04B                    0171 *                     OF
C04B                    0172 *             COLLEGE STATION, TEXAS
C04B                    0173 *
C04B                    0174 *   SOLOS AND SOLED CONTAIN THE ESC SEQUENCES AND OTHER
C04B                    0175 *  FULL IMPLEMENTATION FEATURES.
C04B                    0176 *
C04B E5                 0177 VDMOT    PUSH     H           SAVE EVERYBODY
C04C D5                 0178 VDMO1    PUSH     D           ENTRY FROM DEVICE SELECT
C04D C5                 0179          PUSH     B
C04E F5                 0180          PUSH     PSW
C04F 78                 0181          MOV      A,B         SAVE IN B...STRIP PARITY BEFORE
SCREEN!
C050 21 65 C1           0182          LXI      H,TBL
C053 CD 62 C0           0183          CALL     TSRCH       GO PROCESS
C056                    0184 *
C056 CD 21 C1           0185 GOBACK   CALL     VDADD       GET SCREEN ADDRESS
C059 7E                 0186          MOV      A,M
C05A F6 80              0187          ORI      80H
C05C 77                 0188          MOV      M,A         CURSOR IS BACK ON
C05D F1                 0189 GOBK     POP      PSW
C05E C1                 0190          POP      B
C05F D1                 0191          POP      D           RESTORE ALL REGISTERS
C060 E1                 0192          POP      H
C061 C9                 0193          RET      .           EXIT FROM VDMOT
C062                    0194 *
C062                    0195 *
C062 7E                 0196 TSRCH    MOV      A,M         GET CHR FROM TABLE
C063 B7                 0197          ORA      A
C064 CA 7B C0           0198          JZ       CHAR        ZERO IS THE LAST
C067 B8                 0199          CMP      B           TEST THE CHR
C068 23                 0200          INX      H           POINT FORWARD
C069 C2 76 C0           0201          JNZ      NEXT
C06C E5                 0202          PUSH     H           FOUND ONE ... SAVE ADDRESS
C06D CD 3D C1           0203          CALL     CREM        REMOVE CURSOR
C070 E1                 0204          POP      H
C071                    0205 *
C071                    0206 *
C071                    0207 *    THIS ROUTINE DISPATCHES TO THE ADDRESS POINTED TO
C071                    0208 *  BY THE HL REGISTER PAIR.  THE RETURN ADDRESS IS THE
C071                    0209 *  LAST ENTRY ON THE STACK.
C071                    0210 *
C071 7E                 0211 DISPT    MOV      A,M         GET LOW BYTE
C072 23                 0212          INX      H
C073 66                 0213          MOV      H,M         AND THE HIGH
C074 6F                 0214          MOV      L,A         WE HAVE PLACED THEM BOTH
C075 E9                 0215          PCHL     .           GO TO IT
C076                    0216 *
C076                    0217 *
C076 23                 0218 NEXT     INX      H           GO TO NEXT
C077 23                 0219          INX      H
C078 C3 62 C0           0220          JMP      TSRCH
C07B                    0221 *
C07B                    0222 *
C07B 78                 0223 CHAR     MOV      A,B         GET CHARACTER
C07C B7                 0224          ORA      A
C07D C8                 0225          RZ       .           RETURN IF A NULL
C07E FE 7F              0226          CPI      7FH         IS IT A DEL?
C080 C8                 0227          RZ       .           GO BACK IF SO
C081                    0228 *
C081                    0229 *
C081                    0230 *
C081 CD 21 C1           0231 OCHAR    CALL     VDADD       GET SCREEN ADDRESS
C084 78                 0232          MOV      A,B         GET CHARACTER FOR OUTPUT
C085 E6 7F              0233          ANI      7FH         NO HIGH BITS GO PAST HERE
C087 77                 0234          MOV      M,A         PUT CHR ON SCREEN
C088 3A 00 C8           0235          LDA      NCHAR       GET CHARACTER POSITION
C08B FE 3F              0236          CPI      63          END OF LINE?
C08D DA AD C0           0237          JC       OK
C090 3A 01 C8           0238          LDA      LINE
C093 FE 0F              0239          CPI      15          END OF SCREEN?
C095 C2 AD C0           0240          JNZ      OK
C098                    0241 *
C098                    0242 *   END OF SCREEN...ROLL UP ONE LINE
C098                    0243 *
C098 AF                 0244 SCROLL   XRA      A
C099 32 00 C8           0245          STA      NCHAR       BACK TO FIRST CHAR POSITION
C09C 4F                 0246 SROL     MOV      C,A
C09D CD 28 C1           0247          CALL     VDAD        CALCULATE LINE TO BE BLANKED
C0A0 AF                 0248          XRA      A
C0A1 CD E9 C0           0249          CALL     CLIN1       CLEAR IT
C0A4 3A 02 C8           0250          LDA      BOT
C0A7 3C                 0251          INR      A
C0A8 E6 0F              0252          ANI      0FH
C0AA C3 DD C0           0253          JMP      ERAS3
C0AD                    0254 *
C0AD                    0255 *   INCREMENT LINE COUNTER IF NECESSARY
C0AD                    0256 *
C0AD 3A 00 C8           0257 OK       LDA      NCHAR       GET CHR POSITION
C0B0 3C                 0258          INR      A
C0B1 32 00 C8           0259          STA      NCHAR       STORE THE NEW
C0B4 FE 40              0260          CPI      64
C0B6 D8                 0261          RC
C0B7 AF                 0262          XRA      A           WE'RE PAST THE END... REWIND THE COUNT
C0B8 32 00 C8           0263          STA      NCHAR
C0BB 3A 01 C8           0264          LDA      LINE        GET THE LINE COUNT
C0BE 3C                 0265          INR      A
C0BF E6 0F              0266          ANI      0FH         MOD 15 INCREMENT
C0C1 32 01 C8           0267 CUR      STA      LINE        STORE THE NEW
C0C4 C9                 0268          RET
C0C5                    0269 *
C0C5                    0270 *    ERASE SCREEN
C0C5                    0271 *
C0C5 21 00 CC           0272 PERSE    LXI      H,VDMEM     POINT TO SCREEN
C0C8 36 A0              0273          MVI      M,80H+' '   THIS IS THE CURSOR
C0CA                    0274 *
C0CA 23                 0275 ERAS1    INX      H           BUMP THE COUNT
C0CB 7C                 0276          MOV      A,H         GET HIGH ORDER
C0CC FE D0              0277          CPI      0D0H        THE TOP
C0CE D2 D6 C0           0278          JNC      ERAS2
C0D1 36 20              0279          MVI      M,' '       PUT IN A BLANK
C0D3 C3 CA C0           0280          JMP      ERAS1
C0D6                    0281 *
C0D6 AF                 0282 ERAS2    XRA      A
C0D7 32 01 C8           0283          STA      LINE        ZERO LINE
C0DA 32 00 C8           0284          STA      NCHAR       LEFT SIDE OF SCREEN
C0DD                    0285 *
C0DD D3 FE              0286 ERAS3    OUT      DSTAT       RESET SCROLL PARAMETERS
C0DF 32 02 C8           0287          STA      BOT         BEGINNING OF TEXT OFFSET
C0E2 C9                 0288          RET
C0E3                    0289 *
C0E3                    0290 *
C0E3 CD 21 C1           0291 CLINE    CALL     VDADD       GET CURRENT SCREEN ADDRESS
C0E6 3A 00 C8           0292          LDA      NCHAR       CURRENT CURSOR POSITION
C0E9 FE 40              0293 CLIN1    CPI      64          NO MORE THAN 63
C0EB D0                 0294          RNC      .           ALL DONE
C0EC 36 20              0295          MVI      M,' '       ALL SPACED OUT
C0EE 23                 0296          INX      H
C0EF 3C                 0297          INR      A
C0F0 C3 E9 C0           0298          JMP      CLIN1       LOOP TO END OF LINE
C0F3                    0299 *
C0F3                    0300 *    HOME CURSOR
C0F3                    0301 *
C0F3 AF                 0302 PHOME    XRA      A
C0F4 32 00 C8           0303          STA      NCHAR
C0F7 C3 C1 C0           0304          JMP      CUR
C0FA                    0305 *
C0FA                    0306 * MOVE CURSOR DOWN ONE LINE
C0FA                    0307 *
C0FA 3A 01 C8           0308 PDOWN    LDA      LINE
C0FD FE 0F              0309          CPI      15
C0FF C8                 0310          RZ       .           HOW FAR IS DOWN?
C100 3C                 0311          INR      A
C101 C3 C1 C0           0312          JMP      CUR
C104                    0313 *
C104                    0314 * ROUTINE TO MOVE THE CURSOR UP ONE LINE
C104                    0315 *
C104 3A 01 C8           0316 PUP      LDA      LINE        GET LINE COUNT
C107 B7                 0317          ORA      A
C108 C8                 0318          RZ       .           DON'T GO MORE UP THAN UP
C109 3D                 0319          DCR      A
C10A C3 C1 C0           0320          JMP      CUR
C10D                    0321 *
C10D                    0322 *  MOVE CURSOR LEFT ONE POSITION
C10D                    0323 *
C10D 3A 00 C8           0324 PLEFT    LDA      NCHAR
C110 B7                 0325          ORA      A
C111 C8                 0326          RZ       .           DON'T GO MORE BACK THAN BACK
C112 3D                 0327          DCR      A
C113 32 00 C8           0328 PCUR     STA      NCHAR
C116 C9                 0329          RET
C117                    0330 *
C117                    0331 *     CURSOR  RIGHT ONE POSITION
C117                    0332 *
C117 3A 00 C8           0333 PRIT     LDA      NCHAR
C11A FE 3F              0334          CPI      63
C11C C8                 0335          RZ       .           HOW RIGHT CAN WE BE?
C11D 3C                 0336          INR      A
```

```
C11E C3 13 C1          JMP     PCUR
C121                *
C121                *   ROUTINE TO CALCULATE SCREEN ADDRESS
C121                *
C121                *   ENTRY AT:    RETURNS:
C121                *
C121                *         VDADD  CURRENT SCREEN ADDRESS
C121                *         VDAD2  ADDRESS OF CURRENT LINE, CHAR 'C'
C121                *         VDAD   LINE 'A', CHARACTER POSITION 'C'
C121                *
C121 3A 00 C8       VDADD   LDA     NCHAR       GET CHARACTER POSITION
C124 4F                     MOV     C,A         IC' KEEPS IT
C125 3A 01 C8       VDAD2   LDA     LINE        LINE POSITION
C128 6F             VDAD    MOV     L,A         INTO 'L'
C129 3A 02 C8               LDA     BOT         GET TEXT OFFSET
C12C 85                     ADD     L           ADD IT TO THE LINE POSITION
C12D 0F                     RRC     .           TIMES TWO
C12E 0F                     RRC     .           MAKES FOUR
C12F 6F                     MOV     L,A         L HAS IT
C130 E6 03                  ANI     3           MOD THREE FOR LATER
C132 57                     MOV     D,A
C133 3E CC                  MVI     A,<VDMEM    LOW SCREEN OFFSET
C135 82                     ADD     D
C136 67                     MOV     H,A         NOW H IS DONE
C137 7D                     MOV     A,L         TWIST L'S ARM
C138 E6 C0                  ANI     0C0H
C13A 81                     ADD     C
C13B 6F                     MOV     L,A
C13C C9                     RET                 H & L ARE NOW PERVERTED
C13D                *
C13D                *    ROUTINE TO REMOVE CURSOR
C13D                *
C13D CD 21 C1       CREM    CALL    VDADD       GET CURRENT SCREEN ADDRESS
C140 7E                     MOV     A,M
C141 E6 7F                  ANI     7FH         STRIP OFF THE CURSOR
C143 77                     MOV     M,A
C144 C9                     RET
C145                *
C145                *     ROUTINE TO BACKSPACE
C145                *
C145 CD 0D C1       PBACK   CALL    PLEFT
C148 CD 21 C1               CALL    VDADD       GET SCREEN ADDRESS
C14B 36 20                  MVI     M,' '       PUT A BLANK THERE
C14D C9                     RET
C14E                *
C14E                *     ROUTINE TO PROCESS A CARRIAGE RETURN
C14E                *
C14E CD E3 C0       PCR     CALL    CLINE       CLEAR FROM CURRENT CURSOR TO END OF
LINE
C151 AF                     XRA     A           REWIND IT
C152 C3 13 C1               JMP     PCUR        AND STORE THE NEW VALUE
C155                *
C155                *   ROUTINE TO PROCESS LINEFEED
C155                *
C155 3A 01 C8       PLF     LDA     LINE        GET LINE COUNT
C158 FE 0F                  CPI     15          ARE WE AT THE BOTTOM?
C15A D2 61 C1               JNC     SC
C15D 3C                     INR     A
C15E C3 C1 C0               JMP     CUR         ONE MORE LINE UP
C161                *
C161 AF             SC      XRA     A
C162 C3 9C C0               JMP     SROL
C165                *
C165                *
C165                *
C165                *     THIS TABLE DEFINES THE CHARACTERS FOR SPECIAL
C165                * PROCESSING.  IF THE CHARACTER IS NOT IN THE TABLE IT
C165                * GOES TO THE SCREEN.
C165                *
C165 8B             TBL     DB      CLEAR       SCREEN
C166 C5 C0                  DW      PERSE
C168 97                     DB      UP          CURSOR
C169 04 C1                  DW      PUP
C16B 9A                     DB      DOWN
C16C FA C0                  DW      PDOWN
C16E 81                     DB      LEFT
C16F 0D C1                  DW      PLEFT
C171 93                     DB      RIGHT
C172 17 C1                  DW      PRIT
C174 8E                     DB      HOME
C175 F3 C0                  DW      PHOME
C177 0D                     DB      CR          CARRIAGE RETURN
C178 4E C1                  DW      PCR
C17A 0A                     DB      LF          LINE FEED
C17B 55 C1                  DW      PLF
C17D 5F                     DB      BACKS       BACK SPACE
C17E 45 C1                  DW      PBACK
C180 80                     DB      MODE        MODE KEY
C181 B4 C1                  DW      COMND
C183 00                     DB      0           END OF TABLE
C184                *
C184                *   OUTPUT DEVICE TABLE
C184                *
C184 4C C0          OTAB    DW      VDMO1       VDM DRIVER
C186 40 C0                  DW      SEROT       SERIAL OUTPUT
C188 AD C1                  DW      ERROT       ERROR HANDLER (FOR CONSOL)
C18A AD C1                  DW      ERROT       ERROR HANDLER
C18C                *
C18C                *   INPUT DEVICE TABLE
C18C                *
C18C 27 C0          ITAB    DW      KREA1       KEYBOARD INPUT
C18E 36 C0                  DW      SREA1       SERIAL INPUT
C190 AD C1                  DW      ERROT       ERROR HANDLER (FOR CONSOL)
C192 AD C1                  DW      ERROT       ERROR HANDLER
C194                *
C194                *
C194                *            COMMAND TABLE
C194                *
C194                *  THIS TABLE DESCRIBES THE VALID COMMANDS FOR CONSOL
C194                *
C194 54 45          COMTAB  ASC     'TE'
C196 70 C2                  DW      TERM
C198 44 55                  ASC     'DU'
C19A A1 C2                  DW      DUMP
C19C 45 4E                  ASC     'EN'
C19E 07 C3                  DW      ENTER
C1A0 45 58                  ASC     'EX'
C1A2 35 C3                  DW      EXEC
C1A4 54 4C                  ASC     'TL'
C1A6 39 C3                  DW      TLOAD
C1A8 42 41                  ASC     'BA'        SPECIAL COMMAND TO EXECUTE 0
C1AA 00 00                  DW      0
C1AC 00                     DB      0           END OF TABLE MARK
C1AD                *
C1AD                *
C1AD                *              CONSOL PORT ERROR HANDLER
C1AD                *
C1AD AF             ERROT   XRA     A
C1AE 32 04 C8               STA     IPORT       DEFAULT TO SCREEN
C1B1 32 03 C8               STA     OPORT       DEFAULT TO SCREEN
C1B4                *
C1B4                *
C1B4                *
C1B4                *
C1B4                *            =--  COMMAND MODE  --=
C1B4                *
C1B4                *
C1B4                *   THIS ROUTINE GETS AND PROCESSES COMMANDS
C1B4                *
C1B4 31 00 CC       COMND   LXI     SP,SYSTP    SET STACK POINTER
C1B7 CD 12 C2               CALL    PROMPT      PUT PROMPT ON SCREEN
C1BA CD C3 C1               CALL    GCLIN       GET COMMAND LINE
C1BD CD E1 C1               CALL    COPRC       PROCESS THE LINE
C1C0 C3 B4 C1               JMP     COMND       OVER AND OVER
C1C3                *
C1C3                *
C1C3                *
C1C3                *   THIS ROUTINE READS A COMMAND LINE FROM THE SYSTEM
C1C3                *  KEYBOARD AND PROCESSES IT TO THE SCREEN.
C1C3                *
C1C3                *  C/R   TERMINATES THE SEQUENCE ERASING ALL CHARS TO THE
C1C3                *        RIGHT OF THE CURSOR
C1C3                *  L/F   TERMINATES THE SEQUENCE
C1C3                *  MODE  RESTARTS THE COMMAND LINE.
C1C3                *
C1C3 CD 28 C0       GCLIN   CALL    KREAD       READ KEYBOARD
C1C6 FE 20                  CPI     20H
C1C8 47                     MOV     B,A
C1C9 DA D2 C1               JC      PROCS       PROCESS CONTROL CHARACTER
C1CC CD 4B C0       CONT    CALL    VDMOT
C1CF C3 C3 C1               JMP     GCLIN
C1D2                *
C1D2                *   PROCESS CONTROL KEYS
C1D2                *
C1D2 FE 0D          PROCS   CPI     CR
C1D4 CA DD C1               JZ      CRPRC       ERASE THE REMAINING LINE PRIOR TO RETURN
C1D7 FE 0A                  CPI     LF
C1D9 C8                     RZ      .           IF SO GO PROCESS
C1DA C3 C3 C1               JMP     GCLIN       NO CONTROL CHARS TO SCREEN
C1DD                *
C1DD                *
C1DD CD E3 C0       CRPRC   CALL    CLINE       CLEAR REMAINING LINE
C1E0 C9                     RET     .           NOW PROCESS
C1E1                *
C1E1                *
C1E1                *      FIND AND PROCESS    COMMAND
C1E1                *
C1E1 CD 3D C1       COPRC   CALL    CREM        REMOVE THE CURSOR
C1E4 0E 01                  MVI     C,1         SET FOR CHARACTER POSITION
C1E6 CD 25 C1               CALL    VDAD2       GET SCREEN ADDRESS
C1E9 EB                     XCHG
C1EA CD 32 C2               CALL    SCHR        SCAN PAST BLANKS
C1ED CA 8E C3               JZ      ERR1        NO COMMAND?
C1F0 EB                     XCHG    .           HL HAS FIRST CHR
C1F1 11 94 C1               LXI     D,COMTAB    POINT TO COMMAND TABLE
C1F4                *
C1F4                *   THIS ROUTINE SEARCHES THROUGH A TABLE, POINTED TO
C1F4                *  BY 'DE', FOR A DOUBLE CHARACTER MATCH OF THE 'HL'
C1F4                *  MEMORY CONTENT.  IF NO MATCH IS FOUND THE SCAN ENDS
C1F4                *  BY PLACING A QUESTION MARK WITHIN THE SEARCH STRING.
C1F4                *
C1F4 1A             FDCOM   LDAX    D
C1F5 B7                     ORA     A           TEST FOR TABLE END
C1F6 CA 8F C3               JZ      ERR2        NOT FOUND..COMMAND ERROR
C1F9 E5                     PUSH    H           SAVE START OF SCAN ADDRESS
C1FA BE                     CMP     M           TEST FIRST CHR
C1FB 13                     INX     D
C1FC C2 0B C2               JNZ     NCOM
C1FF                *
C1FF 23                     INX     H
C200 1A                     LDAX    D
C201 BE                     CMP     M           NOW SECOND CHARACTER
C202 C2 0B C2               JNZ     NCOM        GOODNESS
C205                *
C205 C1                     POP     B           CLEAR THE STACK
C206 EB                     XCHG    .           DE HAS SCAN ADDRESS
C207 23                     INX     H           HL HAS COMMAND ADDRESS
C208 C3 71 C0               JMP     DISPT       DISPATCH TO IT
C20B                *
C20B                *
C20B 13             NCOM    INX     D           GO TO NEXT ENTRY
C20C 13                     INX     D
C20D 13                     INX     D
C20E E1                     POP     H           GET BACK ORIGINAL ADDRESS
C20F C3 F4 C1               JMP     FDCOM       CONTINUE SEARCH
C212                *
C212                *
C212                *      OUTPUT A CRLF FOLLOWED BY A PROMPT
C212                *  ( WITH CONSOL ALL OPERATIONS ARE ON THE SCREEN)
C212                *
C212 CD 1A C2       PROMPT  CALL    CRLF
C215 06 3E                  MVI     B,'>'       THE PROMPT
C217 C3 4B C0               JMP     VDMOT       PUT IT ON THE SCREEN
C21A                *
C21A 06 0A          CRLF    MVI     B,LF        LINE FEED
C21C CD 4B C0               CALL    VDMOT
C21F 06 0D                  MVI     B,CR        CARRIAGE RETURN
C221 C3 4B C0               JMP     VDMOT       PUT IT OUT AND RETURN
C224                *
C224                *
C224                *  SCAN OVER UP TO 12 CHARACTERS LOOKING FOR A BLANK
C224                *
C224 0E 0C          SBLK    MVI     C,12        MAXIMUM COMMAND STRING
C226 1A             SBLK1   LDAX    D
C227 FE 20                  CPI     BLANK
C229 CA 32 C2               JZ      SCHR        GOT A BLANK NOW SCAN PAST IT
C22C 13                     INX     D
C22D 0D                     DCR     C           NO MORE THAN TWELVE
C22E C2 26 C2               JNZ     SBLK1
C231 C9                     RET     .           GO BACK WITH ZERO FLAG SET
C232                *
C232                *
C232                *  SCAN PAST UP TO 10 BLANK POSITIONS LOOKING FOR
C232                * A NON BLANK CHARACTER.
C232                *
C232 0E 0A          SCHR    MVI     C,10        SCAN TO FIRST NON BLANK CHR WITHIN 10
C234 1A             SCHR1   LDAX    D           GET NEXT CHARACTER
C235 FE 20                  CPI     SPACE
C237 C0                     RNZ     .           WE'RE PAST THEM
C238 13                     INX     D           NEXT SCAN ADDRESS
C239 0D                     DCR     C
C23A C8                     RZ      .           COMMAND ERROR
C23B C3 34 C2               JMP     SCHR1       KEEP LOOPING
C23E                *
C23E                *  THIS ROUTINE SCANS OVER CHARACTERS, PAST BLANKS AND
C23E                * CONVERTS THE FOLLOWING ADDRESS TO HEX.  ERRORS RETURN TO
C23E                * THE ERROR HANDLER.
C23E                *
C23E CD 24 C2       SCONV   CALL    SBLK
C241 CA 8E C3               JZ      ERR1
C244                *
C244                *  THIS ROUTINE CONVERTS ASCII DIGITS INTO BINARY FOLLOWING
C244                * A STANDARD HEX CONVERSION.  THE SCAN STOPS WHEN AN ASCII
C244                * SPACE IS ENCOUNTERED.  PARAMETER ERRORS REPLACE THE ERROR
C244                * CHARACTER ON THE SCREEN WITH A QUESTION MARK.
C244                *
C244 21 00 00       SHEX    LXI     H,0         CLEAR H & L
C247 1A             SHE1    LDAX    D           GET CHARACTER
C248 FE 20                  CPI     20H         IS IT A SPACE?
C24A C8                     RZ      .           IF SO
C24B                *
C24B 29             HCONV   DAD     H           MAKE ROOM FOR THE NEW ONE
C24C 29                     DAD     H
C24D 29                     DAD     H
C24E 29                     DAD     H
C24F CD 5B C2               CALL    HCOV1       DO THE CONVERSION
C252 D2 8E C3               JNC     ERR1        NOT VALID HEXIDECIMAL VALUE
C255 85                     ADD     L
C256 6F                     MOV     L,A         MOVE IT IN
C257 13                     INX     D           BUMP THE POINTER
C258 C3 47 C2               JMP     SHE1
C25B                *
C25B D6 30          HCOV1   SUI     48          REMOVE ASCII BIAS
C25D FE 0A                  CPI     10
C25F D8                     RC      .           IF LESS THAN 9
C260 D6 07                  SUI     7           IT'S A LETTER??
C262 FE 10                  CPI     10H
```

```
C264 C9              0623          RET       .          WITH TEST IN HAND
C265                 0624 *
C265                 0625 *
C265                 0626 *    SYSTEM START UP, CLEAR PART OF RAM AND SET STACK
C265                 0627 * POINTER, FALLING THROUGH TO TERMINAL MODE.
C265                 0628 *
C265 AF              0629 STRTA    XRA       A
C266 4F              0630          MOV       C,A        WE CLEAR THE FIRST 256 BYTES
C267 21 00 C8        0631          LXI       H,SYSRAM   POINT TO SYSTEM RAM
C26A                 0632 *
C26A 77              0633 CLERA    MOV       M,A
C26B 23              0634          INX       H
C26C 0C              0635          INR       C
C26D C2 6A C2        0636          JNZ       CLERA      CLEAR FIRST 256 BYTES
C270                 0637 *
C270                 0638 *
C270                 0639 *
C270                 0640 *                TERM COMMAND
C270                 0641 *
C270                 0642 *   THIS ROUTINE GETS CHARACTERS FROM THE SYSTEM KEYBOARD
C270                 0643 *  AND OUTPUTS THEM TO THE SERIAL OUTPUT PORT.  IT IS
C270                 0644 *  INTENDED TO CONFIGURE THE SoL AS A STANDARD VIDEO
C270                 0645 *  TERMINAL.  COMMAND KEYS ARE NOT OUTPUT TO THE OUTPUT
C270                 0646 *  PORT BUT ARE INTERPRETED AS DIRECT SoL COMMANDS.
C270                 0647 *  THE MODE COMMAND, RECEIVED BY THE KEYBOARD, PUTS THE
C270                 0648 *  SoL IN THE COMMAND MODE.
C270                 0649 *
C270                 0650 *
C270 31 00 CC        0651 TERM     LXI       SP,SYSTP   SET STACK POINTER
C273 CD ED C3        0652          CALL      TOFF       SLOW DOWN THE TAPES
C276 CD C5 C0        0653          CALL      PERSE      CLEAR THE SCREEN
C279                 0654 *
C279 CD 22 C0        0655 KIN      CALL      KSTAT      IS THERE ONE WAITINGI
C27C C2 90 C2        0656          JNZ       TIN
C27F DB FC           0657          IN        KDATA      GET THE CHARACTER
C281 47              0658          MOV       B,A
C282 E6 80           0659          ANI       80H        COMMAND KEY?
C284 CA 8D C2        0660          JZ        TOUT
C287 CD 4B C0        0661          CALL      VDMOT      PROCESS IT
C28A C3 90 C2        0662          JMP       TIN
C28D                 0663 *
C28D CD 41 C0        0664 TOUT     CALL      SDROT      OUTPUT IT TO THE SERIAL PORT
C290 CD 31 C0        0665 TIN      CALL      SSTAT      GET SERIAL STATUS
C293 CA 79 C2        0666          JZ        KIN        LOOP IF NOT
C296 DB F9           0667          IN        SDATA      GET DATA
C298 E6 7F           0668          ANI       7FH        NO HIGH BITS FROM HERE
C29A 47              0669          MOV       B,A        IT'S OUTPUT FROM 'B'
C29B CD 4B C0        0670          CALL      VDMOT      PUT IT ON THE SCREEN
C29E C3 79 C2        0671          JMP       KIN        LOOP OVER AND OVER
C2A1                 0672 *
C2A1                 0673 *
C2A1                 0674 *
C2A1                 0675 *              DUMP COMMAND
C2A1                 0676 *
C2A1                 0677 *     THIS ROUTINE DUMPS CHARACTERS FROM MEMORY TO THE
C2A1                 0678 *  CURRENT OUTPUT DEVICE.  (WITH CONSOL ALL OUTPUT GOES TO
C2A1                 0679 *  THE SCREEN).  ALL VALUES ARE DESPLAYED AS ASCII HEX.
C2A1                 0680 *
C2A1                 0681 *  THE COMMAND FORM IS AS FOLLOWS:
C2A1                 0682 *
C2A1                 0683 *        DUmp addr1 addr2
C2A1                 0684 *
C2A1                 0685 *    THE VALUES FROM ADDR1 TO ADDR2 ARE THEN OUTPUT TO THE
C2A1                 0686 *  OUTPUT DEVICE.  IF ONLY ADDR1 IS SPECIFIED THEN THE
C2A1                 0687 *  VALUE AT THAT ADDRESS IS OUTPUT.
C2A1                 0688 *
C2A1 CD 3E C2        0689 DUMP     CALL      SCONV      SCAN TO FIRST ADDRESS AND CONVERT IT
C2A4 E5              0690          PUSH      H          SAVE THE VALUE
C2A5 CD 32 C2        0691          CALL      SCHR       GET THE NEXT
C2A8 E1              0692          POP       H
C2A9 CA B4 C2        0693          JZ        POVER      NO SECOND VALUE
C2AC E5              0694          PUSH      H
C2AD CD 44 C2        0695          CALL      SHEX       GET SECOND
C2B0 D1              0696          POP       D          THIS IS THE FIRST
C2B1 C3 B6 C2        0697          JMP       NPASS      MIND BENDERS
C2B4                 0698 *
C2B4 54              0699 POVER    MOV       D,H        NO SECOND PARAMETER COPY FIRST TO DE
C2B5 5D              0700          MOV       E,L
C2B6 EB              0701 NPASS    XCHG      .          HL HAS START, DE HAS END
C2B7                 0702 *
C2B7 CD 1A C2        0703 DLOOP    CALL      CRLF
C2BA DB FC           0704          IN        KDATA
C2BC FE 80           0705          CPI       MODE        MODE KEY' WILL ESCAPE THE DUMP
C2BE CA B4 C1        0706          JZ        COMND
C2C1 CD DD C2        0707          CALL      ADOUT      OUTPUT ADDRESS
C2C4 CD E5 C2        0708          CALL      BOUT       ANOTHER SPACE TO KEEP IT PRETTY
C2C7 0E 10           0709          MVI       C,16       VALUES PER LINE
C2C9                 0710 *
C2C9 7E              0711 DLP1     MOV       A,M        GET THE CHR
C2CA C5              0712          PUSH      B          SAVE VALUE COUNT
C2CB CD E2 C2        0713          CALL      HBOUT      SEND IT OUT WITH A BLANK
C2CE CD 02 C3        0714          CALL      ACOMP      COMPARE ADDRESSES
C2D1 D2 B4 C1        0715          JNC       COMND      ALL DONE
C2D4 C1              0716          POP       B          VALUES PER LINE
C2D5 23              0717          INX       H
C2D6 0D              0718          DCR       C          BUMP THE LINE COUNT
C2D7 C2 C9 C2        0719          JNZ       DLP1       NOT ZERO IF MORE FOR THIS LINE
C2DA C3 B7 C2        0720          JMP       DLOOP      DO A LFCR BEFORE THE NEXT
C2DD                 0721 *
C2DD                 0722 *    OUTPUT HL AS HEX 16 BIT VALUE
C2DD                 0723 *
C2DD 7C              0724 ADOUT    MOV       A,H        H FIRST
C2DE CD EA C2        0725          CALL      HEOUT
C2E1 7D              0726          MOV       A,L        THEN L FOLLOWED BY A SPACE
C2E2                 0727 *
C2E2 CD EA C2        0728 HBOUT    CALL      HEOUT
C2E5 06 20           0729 BOUT     MVI       B,' '
C2E7 C3 4B C0        0730          JMP       VDMOT      CONSOL PUTS IT ON THE SCREEN
C2EA                 0731 *
C2EA 4F              0732 HEOUT    MOV       C,A        GET THE CHARACTER
C2EB 0F              0733          RRC
C2EC 0F              0734          RRC       .          MOVE THE HIGH FOUR DOWN
C2ED 0F              0735          RRC
C2EE 0F              0736          RRC
C2EF CD F3 C2        0737          CALL      HEOU1      PUT THEM OUT
C2F2 79              0738          MOV       A,C        THIS TIME THE-LOW FOUR
C2F3                 0739 *
C2F3 E6 0F           0740 HEOU1    ANI       0FH        FOUR ON THE FLOOR
C2F5 C6 30           0741          ADI       48         WE WORK WITH ASCII HERE
C2F7 FE 3A           0742          CPI       58         0-9?
C2F9 DA FE C2        0743          JC        OUTH       YUPI
C2FC C6 07           0744          ADI       7          MAKE IT A LETTER
C2FE 47              0745 OUTH     MOV       B,A        OUTPUT IT FROM REGISTER 'B'
C2FF C3 4B C0        0746          JMP       VDMOT
C302                 0747 *
C302                 0748 *   COMPARE DE AND HL
C302                 0749 *
C302 7D              0750 ACOMP    MOV       A,L
C303 93              0751          SUB       E
C304 7C              0752          MOV       A,H
C305 9A              0753          SBB       D
C306 C9              0754          RET       .          FLAGS ARE SET
C307                 0755 *
C307                 0756 *
C307                 0757 *             ENTER COMMAND
C307                 0758 *
C307                 0759 *   THIS ROUTINE GETS VALUES FROM THE KEYBOARD AND ENTERS
C307                 0760 * THEM INTO MEMORY.  THE INPUT VALUES ARE SCANNED FOLLOWING
C307                 0761 * A STANDARD 'GCLIN' INPUT SO ON SCREEN EDITING MAY TAKE
C307                 0762 * PLACE PRIOR TO THE LINE TERMINATOR.  A BACK SLASH '/'
C307                 0763 * ENDS THE ROUTINE AND-RETURNS CONTROL TO THE COMMAND MODE.
C307                 0764 *
C307 CD 3E C2        0765 ENTER    CALL      SCONV      SCAN OVER CHARS AND GET ADDRESS
C30A E5              0766          PUSH      H          SAVE ADDRESS
C30B                 0767 *
C30B CD 1A C2        0768 ENLOP    CALL      CRLF
C30E 06 3A           0769          MVI       B,':'
C310 CD CC C1        0770          CALL      CONT       GET LINE OF INPUT
C313 CD 3D C1        0771          CALL      CREM       REMOVE THE CURSOR
C316 0E 01           0772          MVI       C,1        START SCAN
C318 CD 25 C1        0773          CALL      VDAD2      GET ADDRESS
C31B EB              0775          XCHG      .          ....TO DE
C31C                 0776 *
C31C 0E 03           0777 ENLO1    MVI       C,3        NO MORE THAN THREE SPACES BETWEEN VALUES
C31E CD 34 C2        0000          CALL      SCHR1 SCAN TO NEXT VALUE
C321 CA 0B C3        0778          JZ        ENLOP      LAST ENTRY FOUND START NEW LINE
C324 1A              0779 ENLO2    LDAX      D          GET THE CHR
C325 FE 2F           0780          CPI       '/'        COMMAND TERMINATOR?
C327 CA B4 C1        0781          JZ        COMND      IF SO...
C32A CD 44 C2        0782          CALL      SHEX       CONVERT VALUE
C32D 7D              0783          MOV       A,L        GET LOW PART AS CONVERTED
C32E E1              0784          POP       H          GET MEMORY ADDRESS
C32F 77              0785          MOV       M,A        PUT IN THE VALUE
C330 23              0786          INX       H
C331 E5              0787          PUSH      H          BACK GOES THE ADDRESS
C332 C3 1C C3        0788          JMP       ENLO1      CONTINUE THE SCAN
C335                 0789 *
C335                 0790 *
C335                 0791 *
C335                 0792 *              EXECUTE COMMAND
C335                 0793 *
C335                 0794 *   THIS ROUTINE GETS THE FOLLOWING PARAMETER AND DOES A
C335                 0795 * PROGRAM JUMP TO THE LOCATION GIVEN BY IT.  IF PROPER
C335                 0796 * STACK OPERATIONS ARE USED WITHIN THE EXTERNAL PROGRAM
C335                 0797 * IT CAN DO A STANDARD 'RET'URN TO THE CONSOL COMMAND MODE.
C335                 0798 *
C335                 0799 *
C335 CD 3E C2        0800 EXEC     CALL      SCONV      SCAN PAST BLANKS AND GET PARAMETER
C338 E9              0801          PCHL                 GO ........ (AMD TAKE NOTE)
C339                 0802 *
C339                 0803 *
C339                 0804 *
C339                 0805 *           TAPE LOAD COMMAND
C339                 0806 *
C339                 0807 *   THIS ROUTINE READS FROM EITHER TAPE UNIT PLACING
C339                 0808 * THE READ DATA INTO MEMORY.  WHILE SPACE WITHIN CONSOL
C339                 0809 * DOES NOT ALLOW FOR 'STANDARD' TAPE ROUTINES THIS
C339                 0810 * COMMAND WILL LOAD SoL- BASIC5 AND OTHER STANDARD SoL
C339                 0811 * SYSTEM SOLFTWARE FOR DIRECT EXECUTION.
C339                 0812 *
C339                 0813 *
C339 CD 24 C2        0814 TLOAD    CALL      SBLK       SCAN TO SPEED PARAMETER
C33C CA 4A C3        0815          JZ        DFLT       DEFAULT TO HIGH SPEED IF NONE
C33F CD 44 C2        0816          CALL      SHEX       CONVERT IT
C342 7D              0817          MOV       A,L        GET VALUE
C343 E6 01           0818          ANI       1          ONLY BIT ZERO COUNTS
C345 3E 20           0819          MVI       A,32       PRETEND ITS SLOW
C347 C2 4B C3        0820          JNZ       SETSP
C34A                 0821 *
C34A AF              0822 DFLT     XRA       A          MAKE IT FAST
C34B F6 C0           0823 SETSP    ORI       TAPE1+TAPE2  CONSOL STARTS BOTH TAPES
C34D D3 FA           0824          OUT       STAPT      START TAPES AND SELECT SPEED
C34F CD F1 C3        0825          CALL      DELAY      WAIT WHILE THE TAPE UNIT WINDS UP
C352 DB FB           0826          IN        TDATA      CLEAR THE UART FLAGS
C354                 0827 *
C354 CD 94 C3        0828 TLOD1    CALL      RHEAD      READ PAST HEADER
C357 C2 54 C3        0829          JNZ       TLOD1      IF ERROR START OVER
C35A                 0830 *
C35A 2A 0C C8        0831          LHLD      BLOCK      GET BLOCK SIZE
C35D EB              0832          XCHG      .          ...TO DE
C35E 2A 0E C8        0833          LHLD      LOADR      GET LOAD ADDRESS
C361                 0834 *
C361 7A              0835 LOLOOP   MOV       A,D        GET COUNT
C362 B3              0836          ORA       E
C363 CA ED C3        0837          JZ        TOFF       COUNT IS ZERO-TURN OFF TAPE AND RETURN
C366 01 00 FF        0838          LXI       B,-256     THIS MANY PRIOR TO CRC TEST
C369 EB              0839          XCHG      .          COUNT TO HL
C36A 09              0840          DAD       B          A LITTLE MATH
C36B D2 E1 C3        0841          JNC       LBLK       NO CARRY, IT'S THE LAST BLOCK
C36E 06 00           0842          MVI       B,0        256 TO READ
C370                 0843 *
C370 0E 00           0844 RDBLK    MVI       C,0        ZERO THE CRC
C372 EB              0845          XCHG      .          ROUND ROBIN
C373                 0000 *
C373 CD C6 C3        0846 RTBYT    CALL      TAPIN      GET CHARACTER
C376 77              0847          MOV       M,A        STORE IT
C377 23              0848          INX       H          BUMP MEMORY LOCATION
C378 A9              0849          XRA       C          UPDATE THE CRC
C379 2F              0850          CMA
C37A 91              0851          SUB       C
C37B 4F              0852          MOV       C,A        STORE THE NEW
C37C 05              0853          DCR       B          COUNT DOWN
C37D C2 73 C3        0854          JNZ       RTBYT      STILL MORE IF NOT ZERO
C380                 0855 *
C380 CD C1 C3        0856          CALL      CRCCK      CHECK CRC AND FALL THROUGH TO ERROR IF NO GOOD
C383 CA 61 C3        0857          JZ        LOLOOP     TEST OK
C386                 0858 *
C386 06 07           0859 TERR     MVI       B,'G'-40H  BELL CHARACTER
C388 CD 4B C0        0860          CALL      VDMOT      PUT IT ON THE SCREEN
C38B C3 B4 C1        0861          JMP       COMND
C38E                 0862 *
C38E                 0863 *
C38E                 0864 *    CONSOL ERROR HANDLER
C38E                 0865 *
C38E EB              0866 ERR1     XCHG      .          GET SCAN ADDRESS
C38F 36 3F           0867 ERR2     MVI       M,'?'      PUT A QUESTION MARK THERE
C391 C3 B4 C1        0868          JMP       COMND      AND GO TO COMMAND MODE
C394                 0869 *
C394                 0870 *   READ  THE HEADER
C394                 0871 *
C394 06 0A           0872 RHEAD    MVI       B,10       FIND 10 NULLS
C396 DB FA           0873 RHEA1    IN        STAPT      GET A BYTE
C398 E6 40           0874          ANI       TDR
C39A CA 96 C3        0875          JZ        RHEA1
C39D DB FB           0876          IN        TDATA      IGNORE ERROR CONDITIONS
C39F B7              0877          ORA       A          ZERO?
C3A0 C2 94 C3        0878          JNZ       RHEAD
C3A3 05              0879          DCR       B
C3A4 C2 96 C3        0880          JNZ       RHEA1      LOOP UNTIL 10 IN A ROW
C3A7                 0881 *
C3A7                 0882 *    WAIT FOR THE START CHARACTER
C3A7                 0883 *
C3A7 CD C6 C3        0884 SOHL     CALL      TAPIN
C3AA 3D              0885          DCR       A
C3AB C2 A7 C3        0886          JNZ       SOHL       WAIT FOR A '1'
C3AE                 0887 *
C3AE                 0888 *    NOW  GET THE HEADER
C3AE                 0889 *
C3AE 21 05 C8        0890          LXI       H,THEAD    POINT TO BUFFER
C3B1 01 00 10        0891          LXI       B,HLEN*256  LENGTH OF HEADER IN 'B',C<O
C3B4                 0892 *
C3B4 CD C6 C3        0893 RHED1    CALL      TAPIN      GET BYTE
C3B7 77              0894          MOV       M,A        STORE IT
C3B8 23              0895          INX       H          INCREMENT ADDRESS
C3B9 A9              0896          XRA       C          NOW CALCULATE THE CRC
C3BA 2F              0897          CMA       .          INSIDE OUT AND UPSIDE DOWN
C3BB 91              0898          SUB       C          SQUEEZE IT
C3BC 4F              0899          MOV       C,A        AND SAVE AGAIN
C3BD 05              0900          DCR       B          WHOLE HEADER YET?
C3BE C2 B4 C3        0901          JNZ       RHED1      LOOP UNTIL DONE
C3C1                 0902 *
C3C1                 0903 *   THIS  ROUTINE GETS THE NEXT BYTE AND COMPARES IT
C3C1                 0904 * TO THE  VALUE IN REGISTER C. THE FLAGS ARE SET ON
C3C1                 0905 * RETURN.
C3C1                 0906 *
C3C1 CD C6 C3        0907 CRCCK    CALL      TAPIN      GET CRC BYTE
```

```
C3C5 C9              0909          RET
C3C6                 0910 *
C3C6                 0911 *
C3C6                 0912 *    THIS ROUTINE GETS THE NEXT AVAILABLE BYTE FROM THE
C3C6                 0913 *  TAPE.  WHILE WAITING FOR THE BYTE THE KEYBOARD IS TESTED
C3C6                 0914 *  FOR A 'MODE' COMMAND.  IF RECEIVED THE TAPE LOAD IS
C3C6                 0915 *  TERMINATED AND A RETURN TO THE COMMAND MODE IS MADE.
C3C6                 0916 *
C3C6 DB FA           0917 TAPIN    IN       STAPT      CHECK STATUS
C3C8 E6 40           0918          ANI      TDR
C3CA C2 D7 C3        0919          JNZ      TREDY      ONE IS AVAILABLE
C3CD DB FC           0920          IN       KDATA      CHECK FOR MODE WHILE WE'RE WAITING
C3CF FE 80           0921          CPI      MODE
C3D1 CA B4 C1        0922          JZ       COMND      MODE WAS GIVEN..ABORT OPERATION
C3D4 C3 C6 C3        0923          JMP      TAPIN      NOT MODE...STAY IN LOOP
C3D7                 0924 *
C3D7 DB FA           0925 TREDY    IN       STAPT
C3D9 E6 18           0926          ANI      TFE+TOE    DATA ERROR?
C3DB C2 86 C3        0927          JNZ      TERR       IF FRAMING OR OVERRUN ERROR
C3DE DB FB           0928          IN       TDATA      GET THE DATA
C3E0 C9              0929          RET
C3E1                 0930 *
C3E1                 0931 * THIS ROUTINE CALCULATES THE LENGTH OF THE LAST BLOCK
C3E1                 0932 *
C3E1 01 FF FF        0933 LBLK     LXI      B,-1
C3E4 09              0934          DAD      B          COMPLEMENT HL
C3E5 23              0935          INX      H          .......TWO'S
C3E6 45              0936          MOV      B,L        LENGTH TO REGISTER 3
C3E7 21 00 00        0937          LXI      H,0        TELL DE WE'RE DONE
C3EA C3 70 C3        0938          JMP      RDBLK      ONWARD TO THE END
C3ED                 0939 *
C3ED                 0940 *   THIS  ROUTINE TURNS THE TAPE UNITS OFF
C3ED                 0941 *
C3ED AF              0942 TOFF     XRA      A
C3EE D3 FA           0943          OUT      STAPT      GIVE COMMAND
C3F0 C9              0944          RET      .          AND GRIND TO A SLOW STOP
C3F1                 0945 *
C3F1                 0946 *
C3F1 11 00 00        0947 DELAY    LXI      D,0        START LOOP
C3F4 1B              0948 DLOP1    DCX      D          DOWN COUNT
C3F5 7A              0949          MOV      A,D
C3F6 B3              0950          ORA      E          TEST FOR ZERO
C3F7 C2 F4 C3        0951          JNZ      DLOP1      IF NOT
C3FA C9              0952          RET
C3FB                 0953 *
C3FB                 0954 *
C3FB                 0955 *
C3FB                 0956 *
C3FB                 0957 *                <<  SoL SYSTEM EQUATES  >>
C3FB                 0958 *
C3FB                 0959 *
C3FB                 0960 *          VDM PARAMETERS
C3FB                 0961 *
C3FB                 0962 VDMEM    EQU      0CC00H     VDM SCREEN MEMORY
C3FB                 0963 *
C3EB                 0964 *
C3FB                 0965 *          KEYBOARD SPECIAL KEY ASSIGNMENTS
C3FB                 0966 *
C3FB                 0967 DOWN     EQU      9AH
C3FB                 0968 UP       EQU      97H
C3FB                 0969 LEFT     EQU      81H
C3FB                 0970 RIGHT    EQU      93H
C3FB                 0971 LOADK    EQU      8CH        LOAD KEY
C3FB                 0972 MODE     EQU      80H
C3FB                 0973 CLEAR    EQU      8BH
C3FB                 0974 HOME     EQU      08EH
C3FB                 0975 BACKS    EQU      5FH        BACKSPACE
C3FB                 0976 LF       EQU      10
C3FB                 0977 CR       EQU      13
C3FB                 0978 BLANK    EQU      ' '
C3FB                 0979 SPACE    EQU      BLANK
C3FB                 0980 CX       EQU      'X'-40H
C3FB                 0981 *
C3FB                 0982 *          PORT ASSIGNMENTS
C3FB                 0983 *
C3FB                 0984 STAPT    EQU      0FAH       STATUS PORT GENERAL
C3FB                 0985 SERST    EQU      0F8H       SERIAL STATUS PORT
C3FB                 0986 SDATA    EQU      0F9H       SERIAL DATA
C3FB                 0987 TDATA    EQU      0FBH       TAPE DATA
C3FB                 0988 KDATA    EQU      0FCH       KEYBOARD DATA
C3FB                 0989 PDATA    EQU      0FDH       PARALLEL DATA
C3FB                 0990 DSTAT    EQU      0FEH       VDM DISPLAY PARAMETER PORT
C3FB                 0991 SENSE    EQU      0FFH       SENSE SWITCHES
C3FB                 0992 *
C3FB                 0993 *
C3FB                 0994 *
C3FB                 0995 *          BIT ASSIGNMENT MASKS
C3FB                 0996 *
C3FB                 0997 SCD      EQU      1          SERIAL CARRIER DETECT
C3FB                 0998 SDSR     EQU      2          SERIAL DATA SET READY
C3FB                 0999 SPE      EQU      4          SERIAL PARITY ERROR
C3FB                 1000 SFE      EQU      8          SERIAL FRAMING ERROR
C3FB                 1001 SOE      EQU      16         SERIAL OVERRUN ERROR
C3FB                 1002 SCTS     EQU      32         SERIAL CLEAR TO SEND
C3FB                 1003 SDR      EQU      64         SERIAL DATA READY
C3FB                 1004 STBE     EQU      128        SERIAL TRANSMITTER BUFFER EMPTY
C3FB                 1005 *
C3FB                 1006 KDR      EQU      1          KEYBOARD DATA READY
C3FB                 1007 PDR      EQU      2          PARALLEL DATA READY
C3FB                 1008 PXDR     EQU      4          PARALLEL DEVICE READY
C3FB                 1009 TFE      EQU      8          TAPE FRAMING ERROR
C3FB                 1010 TOE      EQU      16         TAPE OVERFLOW ERROR
C3FB                 1011 TDR      EQU      64         TAPE DATA READY
C3FB                 1012 TTBE     EQU      128        TAPE TRANSMITTER BUFFER EMPTY
C3FB                 1013 *
C3FB                 1014 SOK      EQU      1          SCROLL OK FLAG
C3FB                 1015 *
C3FB                 1016 TAPE1    EQU      64         TAPE ONE 'ON' BIT
C3FB                 1017 TAPE2    EQU      128        TAPE TWO
C3FB                 1018 *
C3FB                 1019 *
C3FB                 1020 *
C3FB                 1021 *
C3FB                 1022 *               SoL SYSTEM GLOBAL AREA
C3FB                 1023 *
C3FB                 1024          ORG      0C800H     START OF 1K RAM AREA
C800                 1025 *
C800                 1026 SYSRAM   EQU      $          START OF SYSTEM RAM
C800                 1027 SYSTP    EQU      $+1024     STACK IS AT THE TOP
C800                 1028 *
C800                 1029 *
C800                 1030 *           CONSOL PARAMETER AREA
C800                 1031 *
C800                 1032 NCHAR    DS       1          CURRENT CHARACTER POSITION
C801                 1033 LINE     DS       1          CURRENT LINE POSITION
C802                 1034 BOT      DS       1          BEGINNING OF TEXT DISPLACEMENT
C803                 1035 OPORT    DS       1          OUTPUT PORT
C8O4                 1036 IPORT    DS       1          INPUT PORT
C805                 1037 *
C805                 1038 *
C805                 1039 *
C805                 1040 THEAD    DS       5          NAME
C80A                 1041          DS       1          THIS BYTE MUST BE ZERO
C80B                 1042 HTYPE    DS       1          TYPE
C80C                 1043 BLOCK    DS       2          BLOCK SIZE
C80E                 1044 LOADR    DS       2          LOAD ADDRESS
C810                 1045 XEQAD    DS       2          AUTO EXECUTE ADDRESS
C812                 1046 HSPR     DS       3          SPARES
C815                 1047 *
C815                 1048 HLEN     EQU      $-THEAD    LENGTH OF HEADER
C815                 1049 *
C815                 1050 *
```

```
ACOMP   C302    ADOUT   C2DD    AINP    C016    AOUT    C007
BACKS   005F    BGIN    C001    BLANK   0020    BLOCK   C80C
BOT     C802    BOUT    C2E5    CHAR    C07B    CLEAR   008B
CLERA   C26A    CLIN1   C0E9    CLINE   C0E3    COMND   C1B4
COMTA   C194    CONT    C1CC    COPRC   C1E1    CR      000D
CRCCK   C3C1    CREM    C13D    CRLF    C21A    CRPRC   C1DD
CUR     C0C1    CX      0018    DELAY   C3F1    DFLT    C34A
DISPT   C071    DLOOP   C2B7    DLOP1   C3F4    DLP1    C2C9
DOWN    009A    DSTAT   00FE    DUMP    C2A1    ENLO1   C31C
ENLO2   C324    ENLOP   C30B    ENTER   C307    ERAS1   C0CA
ERAS2   C0D6    ERAS3   C0DD    ERR1    C38E    ERR2    C38F
ERROT   C1AD    EXEC    C335    FDCOM   C1F4    GCLIN   C1C3
GOBAC   C056    GOBK    C05D    HBOUT   C2E2    HCONV   C24B
HCOV1   C25B    HEOU1   C2F3    HEOUT   C2EA    HLEN    0010
HOME    008E    HSPR    C812    HTYPE   C80B    IPORT   C804
ITAB    C18C    KDATA   00FC    KDR     0001    KIN     C279
KREA1   CO27    KREAD   C028    KSTAT   C022    LBLK    C3E1
LEFT    0081    LF      000A    LINE    C801    LOADK   008C
LOADR   C80E    LOLOO   C361    MODE    0080    NCHAR   C800
NCOM    C20B    NEXT    C076    NPASS   C2B6    OCHAR   C081
OK      COAD    OPORT   C803    OTAB    C184    OUTH    C2FE
PBACK   C145    PCR     C14E    PCUR    C113    PDATA   00FD
PDOWN   COFA    PDR     0002    PERSE   C0C5    PHOME   C0F3
PLEFT   C1OD    PLF     C155    POVER   C2B4    PRIT    C117
PROCS   C1D2    PROMP   C212    PUP     C104    PXDR    0004
RDBLK   C370    RHEA1   C396    RHEAD   C394    RHED1   C3B4
RIGHT   OO93    RTBYT   C373    SBLK    C224    SBLK1   C226
SC      C161    SCD     0001    SCHR    C232    SCHR1   C234
SCONV   C23E    SCROL   C098    SCTS    0020    SDATA   00F9
SDR     0040    SDROT   C041    SDSR    0002    SENSE   00FF
SEROT   C040    SERST   00F8    SETSP   C34B    SFE     0008
SHE1    C247    SHEX    C244    SINP    C013    SOE     0010
SOHL    C3A7    SOK     0001    SOUT    C004    SPACE   0020
SPE     0004    SREA1   C036    SREAD   C037    SROL    C09C
SSTAT   C031    SIAPT   00FA    STBE    0080    STRTA   C265
SYSRA   C800    SYSTP   CC00    TAPE1   0040    TAPE2   0080
TAPIN   C3C6    TBL     C165    TDATA   00FB    TDR     0040
TERM    C270    TERR    C386    TFE     0008    THEAD   C805
TIN     C290    TLOAD   C339    TLOD1   C354    TOE     0010
TOFF    C3ED    TOUT    C28D    TREDY   C3D7    TSRCH   C062
TTBL    0080    UP      0097    VDAD    C128    VDAD2   C125
VDADD   C121    VDMEM   CC00    VDMO1   C04C    VDMOT   C04B
XEQAD   C810
```

(continued from page 4)

```
009E 22 F1 04    0710         SHLD   SHED+36DH FIREMAN
00A1 2A 78 01    0720         LHLD   FRA1   .. THE LITTLE ENGINE
00A4 22 24 05    0730         SHLD   SHED+3A0H ALSO
00A7 2A 7A 01    0740         LHLD   FRA2   .. HAD A COWCATCHER
00AA 22 26 05    0750         SHLD   SHED+3A2H AND LOTS OF FUNNY
00AD 22 28 05    0760         SHLD   SHED+3A4H WHEELS AND THE
00B0 22 2A 05    0770         SHLD   SHED+3A6H THINGS THAT CONNECTED
00B3 2A 7C 01    0780         LHLD   FRA3   .. AND TWO VERY TINY
00B6 22 2C 05    0780         SHLD   SHED+3A8H WHEELS AT THE VERY
00B9 2A 7E 01    0800         LHLD   FRA4   .. VERY BACK
00BC 22 2E 05    0810         SHLD   SHED+3AAH ALTOGETHER IT LOOKED
00BF 2A 80 01    0820         LHLD   FRA5   .. QUITE LONELY AND YET
00C2 22 30 05    0830         SHLD   SHED+3ACH IT APPEARED VERY
00C5 01 DF FF    0840         LXI    B,-21H  . FUNNY JUST SITTING
00C8 21 62 05    8850         LXI    H,SHED+3DEH THERE ON THE
00CB 36 19       0860 RAILS   MVI    M,19H  .. RAILS
00CD 03          0870         INX    B      .. WITH NOTHING AT ALL
00CE 23          0880         INX    H      .. EVER
00CF AF          0890         XRA    A      .. TO DO
00D0 A8          0900         XRA    B      ..   W E L L !
00D1 C2 CB 00    0910         JNZ    RAILS  .. LET'S RUN IT JUST FOR FUN
00D4 21 00 CC    0920 HOSTL   LXI    H,RRY  GET TRAIN OUT OF SHED
00D7 EB          0930         XCHG   .      MOVE IT FROM SHED
00D8 21 84 01    0940         LXI    H,SHED
00DB 7E          0950 FIRE    MOV    A,M    LIGHT FIRE
00DC 23          0960         INX    H
00DD EB          0970         XCHG
00DE 77          0980         MOV    M,A    ..MOVE IT
00DF 23          0990         INX    H
00E0 7C          1000         MOV    A,H
00E1 EB          1010         XCHG
00E2 FE D0       1020         CPI    0D0H   END OF YARD?
00E4 C2 DB 00    1030         JNZ    FIRE   NO, MAKE MORE STEAM!
00E7 CD ED 00    1040         CALL   TRAVL  GO TAKE TRIP
00EA C3 D4 00    1050         JMP    HOSTLE  NO ROUND TRIPS, JUST DO AGAIN
00ED             1060 .
00ED 01 40 03    1070 TRAVL   LXI    B,64.13 MAKE TRIP 13 MILES
00F0 C5          1080 CHOO    PUSH   B      CHUFF ONCE
00F1 CD FF 00    1090         CALL   STROK  MAKE FORWARD MOTION
00F4 CD 0F 01    1100         CALL   TURN   TURN WHEELS
00F7 C1          1110         POP    B
00F8 0B          1120         DCX    B      CLICK ODOMETER
00F9 AF          1130         XRA    A
00FA A8          1140         XRA    B      13 MILES YET?
00FB C2 F0 00    1150         JNZ    CHOO   NO
00FE C9          1160         RET    .      DO NEXT TRIP
00FF             1170 .
00FF 21 01 CC    1180 STROK   LXI    H,0CC01H MOVE DOWN THE TRACK
0102 0E D0       1190         MVI    C,0D0H
0104 7E          1200 COAL    MOV    A,M    ADD MORE COAL
0105 2B          1210         DCX    H
0106 77          1220         MOV    M,A
0107 23          1230         INX    H
0108 23          1240         INX    H
0109 7C          1250         MOV    A,H
010A B9          1260         CMP    C      ENOUGH COAL?
010B C2 04 01    1270         JNZ    COAL   NO, PUT MORE ON!
010E C9          1280         RET
010F             1290 .
010F 21 00 CC    1300 TURN    LXI    H,0CC00H TURN WHEELS
0112 7E          1310 HISS    MOV    A,M
0113 23          1320         INX    H
0114 FE 06       1330         CPI    6      FIND COWCATCHER
0116 C2 12 01    1340         JNZ    HISS   LOOK AGAIN
0119 23          1350 FOUND   INX    H
011A 7E          1360 AXLE    MOV    A,M
011B FE 11       1370         CPI    11H    WHAT QUARTER TURN?
011D C2 31 01    1380         JNZ    NXT1
0120 11 7E 14    1390         LXI    D,147EH NEXT QUARTER TURN
0123             1400 .
0123 06 07       1410 AXL1    MVI    B,7    DO 4 AXLES
0125 72          1420 AXL2    MOV    M,D    WHEELS MOVED HERE
0126 05          1430         DCR    B      LAST WHEEL?
0127 CA 52 01    1440         JZ     WORK   I'VE BEEN WORKIN'...
012A 23          1450         INX    H      ..ON THE RAILROAD..
012B 73          1460         MOV    M,E    (SIDE RODS MOVED HERE)
012C 23          1470         INX    H      ..ALL THE LIVE..
012D 05          1480         DCR    B      ..LONG DAY.....
012E C2 25 01    1490         JNZ    AXL2
0131             1500 .
0131 FE 14       1510 NXT1    CPI    14H    QUARTER TURN
0133 C2 3C 01    1520         JNZ    NXT2
0136 11 2D 13    1530         LXI    D,132DH
0139 C3 23 01    1540         JMP    AXL1
013C FE 13       1550 NXT2    CPI    13H    QUARTER TURN
```

```
013E C2 47 01    1560         JNZ    NXT3
0141 11 5F 12    1570         LXI    D,125FH
0144 C3 23 01    1580         JMP    AXL1
0147 FE 12       1590 NXT3    CPI    12H    QUARTER TURN
0149 C2 1A 01    1600         JNZ    AXLE
014C 11 2D 11    1610         LXI    D,112DH
014F C3 23 01    1620         JMP    AXL1
0152             1630 .
0152 01 B8 0B    1640 WORK    LXI    B,3000
0155 0B          1650 OIL     DCX    B,
0156 DB 00       1660         IN     0      DID THE CONDUCTOR
0158 E6 40       1670         ANI    RDA    ..FLAG US DOWN?
015A C2 63 01    1680         JNZ    QUIT   QUITTIN' TIME
015D AF          1690         XRA    A      NO, THEN HIGHBALL
015E A8          1700         XRA    B
015F C2 55 01    1710         JNZ    OIL    NO SQUEEKS, PLEASE!
0162 C9          1720         RET
0163             1730 .
0163             1740 . PUT A JUMP TO WHEREVER YOU NEED TO RETURN IN THE NEXT
0163 C3 60 E0    1750 QUIT    JMP    ALS8   ..YOUR RETURN LINK GOES HERE
0166             1760 .
0166             1770 RDA     EQU    40H    PUT YOUR DATA READY FLAG HERE
0166             1780 RRY     EQU    0CC00H
0166             1790 SMOKE   EQU    6FH    SMOKE CHARACTER
0166 01 10       1800 CAB1    DW     1001H  CAB DESCRIPTION
0168 10 5D       1810 CAB2    DW     5D10H  "   "
016A 7E 20       1820 CAB3    DW     207EH  "   "
016C 28 0A       1830 BOI1    DW     0A28H  BOILER DESCRIPTION
016E 0A 19       1840 BOI2    DW     190AH  "      "
0170 0A 0A       1850 BOI3    DW     0A0AH  "      "
0172 0A 5B       1860 BOI4    DW     5B0AH  "      "
0174 39 39       1870 BOI5    DW     3939H  "      "
0176 5D 20       1880 BOI6    DW     205DH  "      "
0178 06 20       1890 FRA1    DW     2006H  FRAME DESCRIPTION
017A 11 2D       1900 FRA2    DW     2D11H  "     "
017C 11 20       1910 FRA3    DW     2011H  "     "
017E 6F 2D       1920 FRA4    DW     2D6FH  "     "
0180 6F 20       1930 FRA5    DW     206FH  "     "
0182 19 19       1940 TIES    DW     1919H TIES DESCRIPTION
0184 00          1950 SHED    NOP    .     THIS IS SHED AREA
0185             1960 CLOUD   EQU    $+0172H  BEGINNING OF CLOUD
0185             1970 .
0185
```

```
:10000000AFD3C82184010100083620230BAFA8C2090021F702110000060F07
:1A001A00CD5000112B00060DCD50001130000609CD50001134000604CD506A
:1A00340000113B000602CD5000113C000601CD5000113E000601CD5000C39A
:1A004E00590019366F2305C25100C921A604361621A904360721AB04366EE7
:1A0068002A660122AE042A680122B0042A6A0122B2042A6C0122E5042A6E09
:1A0082000122E7042A700122E90422EB042A720122ED042A740122EF042A0D
:1A009C00760122F1042A78012224052A7A012226052228052 22A052A7C0195
:1A00B600222C052A7E01222E052A800122300501DFFF21620536190323AF52
:1A00D000A8C2CB002100CCEB2184017E23EB77237CEBFED0C2DB00CDED00B1
:1A00EA00C3D400014003C5CDFF00CD0F01C10BAFA8C2F000C92101CC0ED049
:1A0104007E2B7723237CB9C20401C92100CC7E23FE06C21201237EFE11C2DD
:1A011E003101117E1406077205CA520123732305C22501FE14C23C01112D5C
:1A01380013C32301FE13C24701115F12C32301FE12C21A01112D11C323010C
:1A01520001B80B0BDB00E640C26301AFA8C25501C9C300060110105D7E2046
:1A016C00280A0A190A0A0A5B39395D200620112D11206F2D6F20191900CB
:00
```