Consol 1K Resident Operating System

By Processor Technology
Emeryville, CA

INTRODUCTION

CONSOL is a resident software operating system for Processor Technology's new single PCB intelligent microcomputer terminal product called SoL Terminal Computer™. The basic SoL Terminal Computer includes the following functional sections;

- 8080A CPU
- 1K bytes of static, low power RAM
- UART controlled RS-232 and 20 ma TTY serial I/O port with multiple baud rates of 75 → 9600 - Switch selectable.
- Video display circuit identical to the VDM-1 sold by Processor Technology. The video display section includes its own 1K bytes of RAM refresh memory. This RAM is in addition to the 1K RW RAM program memory.
- Parallel input-output port for data communications with fully implemented handshaking logic.
- ASCII Keyboard input port.
- PROM/ROM plug-in personality module for up to 2048 bytes of stored program.
- 300 or 1200 baud rate Kansas City Standard Audio cassette tape interface.
- MIB bus compatibility and expansion capability with all Altair/IMSAI/PTC bus plug-in products.

SOL TERMINAL COMPUTER SOFTWARE CONFIGURATIONS

The SoL Terminal Computer™ can be configured by plug-in resident software modules as a stand alone microcomputer or as an intelligent remote editoring terminal.

Basic system operating modes are stored in ROM or PROM on plug-in personality modules with a capacity of up to 2048 words. These modules may be changed in a few seconds to totally reconfigure the system for different applications. Other operating programs, such as BASIC AND FOCAL High level languages, can be loaded automatically into read/write memory (RAM) from cassette tape or floppy disc.

Software control programs for the operation of SoL are designed for three different levels of use. The first level program, CONSOL,™ is contained in 1K of PROM and is designed to allow simple terminal operations. In addition, CONSOL allows direct control of the basic computer functions for entering data to, or examining data in, any memory location, or executing a program stored at a known location in memory.

The second level, designed for advanced terminal operations, is the SOLED™ editing terminal system. SOLED uses the full 2048 word capacity of a personality module and contains code to allow screen, file and cassette tape editing/transmission operations.

The stand-alone operating system, SOLOS,™ turns the SoL into a versatile computer that is easy to use, but every bit as powerful as any 8080-based system available today. Using SOLOS and the built-in cassette interface, BASIC can be loaded in less than a minute following power-on. BASIC programs can be both saved and executed from cassette. The SoL operating under SOLOS, brings true 8080 computer power away from hardware tinkering to direct application and problem solving.

CONSOL SOFTWARE

CONSOL is configured to allow the SoL TERMINAL/COMPUTER to operate as a standard CRT terminal and to provide access to the essential computer capabilities inherent within SoL. The CONSOL software allows self test and small diagnostic programs to be entered to the system memory and executed thus providing verification of correct system operation. In addition, CONSOL contains standardized entry points for all normal I/O operations. These entry point routines are common with each of the SoL System Solftware allowing each personality module in the SoL product line to interface with external programs in an almost identical manner.

A cassette read routine is included in the CONSOL module software allowing SoL System Solftware to be loaded and run in a SoL System with additional memory. SoL System Solftware includes BASIC, FOCAL, a Scientific Calculator and numerous "game" packages including an 8K assembly language version of STARTREK called TREK80.

CONSUL OPERATION

When power is applied to the SoL unit, CONSOL initializes the system ram area, clears the screen, and enters the terminal mode.

In this mode the Sol System acts as a standard CRT terminal sending keyboard data to an output port and displaying received data on the screen. The COMMAND KEYS of the keyboard are not transmitted to the output port but are interpreted as direct internal operation keys. CURSOR MOVEMENT, HOME and CLEAR SCREEN all operate in this manner, while

MODE causes an immediate change in the operation of the SoL Terminal Computer.

When the MODE key is depressed CONSOL issues a prompt (>) and waits for a command line to be entered vie the keyboard. The SoL is now operating as a computer and is ready to accept one of the following commands:

DUmp	Dump memory locations to screen
ENter	Enter data to memory
EXecute	Execute a program in external memory
BAsic	Execute a program located at address zero
TErminal	Return to terminal mode
TLoad	Load program or data from cassette tape
MODE	Press key to start new command line

CONSOL COMMANDS

The seven SoL CONSOL commands are defined in the following:

DUMP <addr> <addr>

The DUMP command displays memory data on the screen in ASCII Hexidecimal representation. As with all SoL commands the command is recognized by the first two characters and up to ten additional characters can be input without an error being forced. Thus, DU; DUST; DUMP; DUMPTHESE would all be recognized as being a DUMP command.

At least one address must follow the command or an error will result. Entering the command DU followed by addr will result in the data at 'addr' being displayed on the screen. If two addresses are defined then all values from the first address to the last address will be displayed. The following example shows the DUMP command with the start and terminating addresses sepcified;

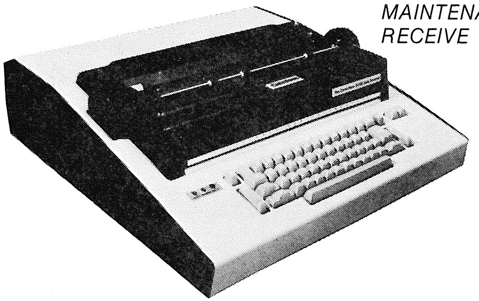
```
DUMP 0      EF
```

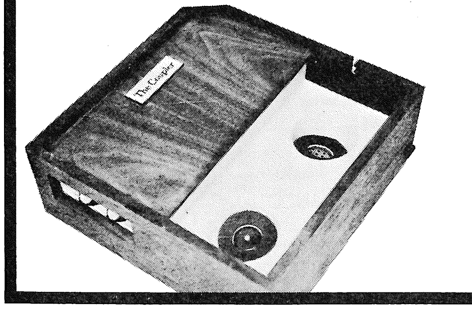
Up to ten blanks may be inserted between each parameter without forcing an error condition. Errors are flagged by a question mark (?) replacing the character where the error occurred. For example if the DU command were given without an address, the question mark would appear ten spaces to the right of the 'U' character of the dump command.

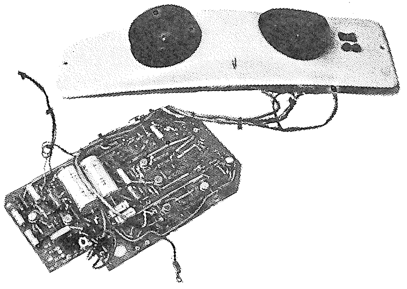
ENTER addr

The enter command places sequential bytes into memory beginning at the specified address. Data, represented as hexidecimal values, are entered from the keyboard for storage in memory. Values are entered one line at a time with each line terminated by a carriage return or linefeed. The ENTER command function itself is terminated with a slash (/) and the CONSOL operating system returns to the command mode when the slash is encountered.

Data input lines are terminated with a carriage return or line feed. If the terminator is a C/R, CONSOL will erase all characters from the current cursor location to the end of the screen line. In this case, all valid input should be to the left of the cursor. If an error occurred during input the cursor may be moved back to







the left to correct the error.

TLoad <speed>

CONSOL includes routines to read standardized Solftware from cassette tape. This standardized software is recorded on tape with a sixteen byte header that includes NAME, LOAD INFORMATION, FILE TYPE and execute address. CONSOL, because of space limitations, is unable to search for a program or file by name. After receiving the TLOAD command, CONSOL locates the next available data, uses the header information and loads the file to memory.

After loading the data, CONSOL returns to the command mode where the EXEC command can be used to execute the just loaded program. In addition, a return can normally be made to the command mode by pressing the MODE key. Program Space limitations again limited the mechanization of the escape function during the header search, so if the operation software system locks up in this routine the standard SoL restart must be used to escape.

The Audio Cassette Interface electronics within the SoL will record or receive data at either of two standard speeds. TLOAD will accept a parameter to select this speed, 0 being high speed and 1 being low (300 and 1200 bits per second). If no parameter is given, CONSOL will default to high speed operation as all standard SoL-System Solftware is recorded at this speed.

EXecute addr

The execute command is used to run programs located in external memory. CONSOL branches to the external routine in a manner similar to a CALL so that the program can return to the command mode using a standard RET instruction if normal stack operations are used.

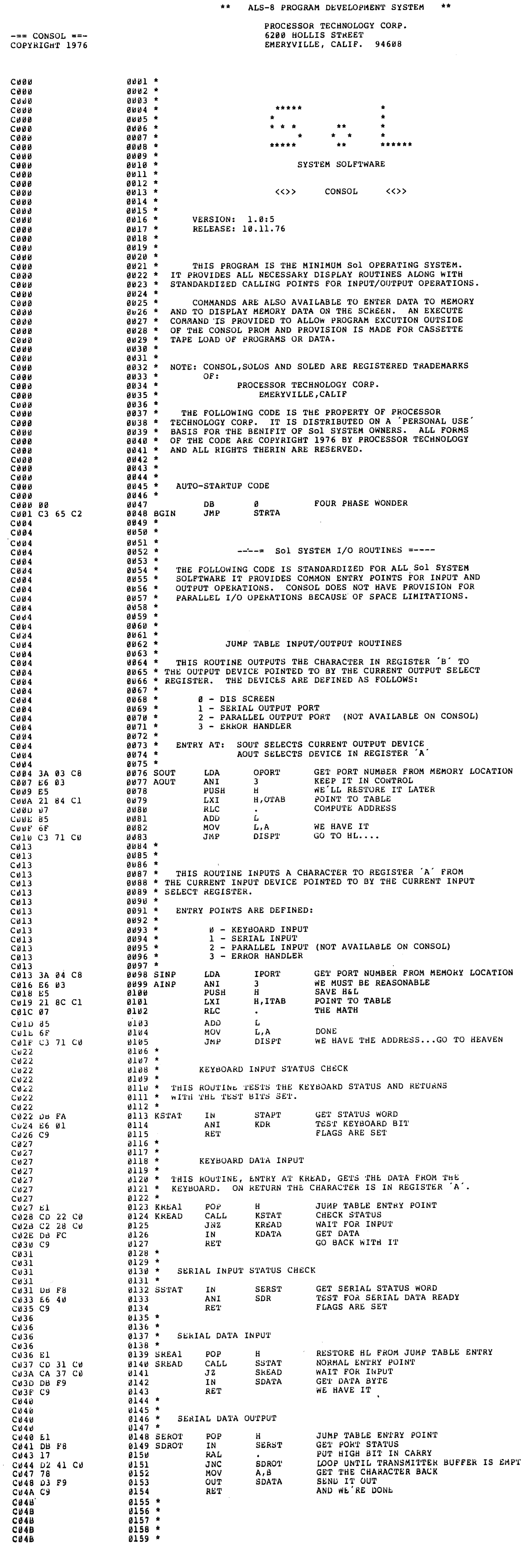
BAsic

The BASIC command is provided for executing programs whose starting address is 0. (Such as SoL-BASIC5).

STANDARD I/O ROUTINES

All SoL System personality modules contain similar I/O code for input/output operations. CONSOL, using 1K of memory, has routines for KEYBOARD and SERIAL PORT input as well as SERIAL and VIDEO DISPLAY OUTPUT. Although the same code for SOLOS and SOLED contains expanded functions, the I/O operations appear almost identical when used with external software.

SoL BASIC5 for example performs all I/O using the jump table of the personality modules. Thus, without altering BASIC the user may output to either the serial port or to the display screen. Provision is also made within BASIC to programatically change to any of the four available Input or Output options. CONSOL is of course limited to the two provided.



```
                                      **   ALS-8 PROGRAM DEVELOPMENT SYSTEM   **

                                              PROCESSOR TECHNOLOGY CORP.
-== CONSOL ==-                                6200 HOLLIS STREET
COPYRIGHT 1976                                EMERYVILLE, CALIF.  94608

C000                 *
C000                 *
C000                 *
C000                 *                     *****               *
C000                 *                    *                    *
C000                 *                    * * *       **       *
C000                 *                         *     *  *      *
C000                 *                    *****       **       ******
C000                 *
C000                 *                         SYSTEM SOLFTWARE
C000                 *
C000                 *
C000                 *                     <<>>     CONSOL     <<>>
C000                 *
C000                 *
C000                 *      VERSION:  1.0:5
C000                 *      RELEASE: 10.11.76
C000                 *
C000                 *
C000                 *
C000                 *      THIS PROGRAM IS THE MINIMUM Sol OPERATING SYSTEM.
C000                 *  IT PROVIDES ALL NECESSARY DISPLAY ROUTINES ALONG WITH
C000                 *  STANDARDIZED CALLING POINTS FOR INPUT/OUTPUT OPERATIONS.
C000                 *
C000                 *      COMMANDS ARE ALSO AVAILABLE TO ENTER DATA TO MEMORY
C000                 *  AND TO DISPLAY MEMORY DATA ON THE SCREEN.  AN EXECUTE
C000                 *  COMMAND IS PROVIDED TO ALLOW PROGRAM EXCUTION OUTSIDE
C000                 *  OF THE CONSOL PROM AND PROVISION IS MADE FOR CASSETTE
C000                 *  TAPE LOAD OF PROGRAMS OR DATA.
C000                 *
C000                 *
C000                 *  NOTE: CONSOL,SOLOS AND SOLED ARE REGISTERED TRADEMARKS
C000                 *        OF:
C000                 *              PROCESSOR TECHNOLOGY CORP.
C000                 *                  EMERYVILLE,CALIF
C000                 *
C000                 *    THE FOLLOWING CODE IS THE PROPERTY OF PROCESSOR
C000                 *  TECHNOLOGY CORP.  IT IS DISTRIBUTED ON A 'PERSONAL USE'
C000                 *  BASIS FOR THE BENIFIT OF Sol SYSTEM OWNERS.  ALL FORMS
C000                 *  OF THE CODE ARE COPYRIGHT 1976 BY PROCESSOR TECHNOLOGY
C000                 *  AND ALL RIGHTS THERIN ARE RESERVED.
C000                 *
C000                 *
C000                 *
C000                 *   AUTO-STARTUP CODE
C000                 *
C000 00                       DB       0          FOUR PHASE WONDER
C001 C3 65 C2        BGIN     JMP      STRTA
C004                 *
C004                 *
C004                 *
C004                 *              -----=  Sol SYSTEM I/O ROUTINES =----
C004                 *
C004                 *   THE FOLLOWING CODE IS STANDARDIZED FOR ALL Sol SYSTEM
C004                 *   SOLFTWARE IT PROVIDES COMMON ENTRY POINTS FOR INPUT AND
C004                 *   OUTPUT OPERATIONS.  CONSOL DOES NOT HAVE PROVISION FOR
C004                 *   PARALLEL I/O OPERATIONS BECAUSE OF SPACE LIMITATIONS.
C004                 *
C004                 *
C004                 *
C004                 *
C004                 *            JUMP TABLE INPUT/OUTPUT ROUTINES
C004                 *
C004                 *   THIS ROUTINE OUTPUTS THE CHARACTER IN REGISTER 'B' TO
C004                 * THE OUTPUT DEVICE POINTED TO BY THE CURRENT OUTPUT SELECT
C004                 * REGISTER.  THE DEVICES ARE DEFINED AS FOLLOWS:
C004                 *
C004                 *       0 - DIS SCREEN
C004                 *       1 - SERIAL OUTPUT PORT
C004                 *       2 - PARALLEL OUTPUT PORT  (NOT AVAILABLE ON CONSOL)
C004                 *       3 - ERROR HANDLER
C004                 *
C004                 *   ENTRY AT:  SOUT SELECTS CURRENT OUTPUT DEVICE
C004                 *              AOUT SELECTS DEVICE IN REGISTER 'A'
C004                 *
C004 3A 03 C8        SOUT     LDA      OPORT      GET PORT NUMBER FROM MEMORY LOCATION
C007 E6 03           AOUT     ANI      3          KEEP IT IN CONTROL
C009 E5                       PUSH     H          WE'LL RESTORE IT LATER
C00A 21 84 C1                 LXI      H,OTAB     POINT TO TABLE
C00D 07                       RLC      .          COMPUTE ADDRESS
C00E 85                       ADD      L
C00F 6F                       MOV      L,A        WE HAVE IT
C010 C3 71 C0                 JMP      DISPT      GO TO HL....
C013                 *
C013                 *
C013                 *
C013                 *   THIS ROUTINE INPUTS A CHARACTER TO REGISTER 'A' FROM
C013                 * THE CURRENT INPUT DEVICE POINTED TO BY THE CURRENT INPUT
C013                 * SELECT REGISTER.
C013                 *
C013                 *   ENTRY POINTS ARE DEFINED:
C013                 *
C013                 *         0 - KEYBOARD INPUT
C013                 *         1 - SERIAL INPUT
C013                 *         2 - PARALLEL INPUT (NOT AVAILABLE ON CONSOL)
C013                 *         3 - ERROR HANDLER
C013                 *
C013 3A 04 C8        SINP     LDA      IPORT      GET PORT NUMBER FROM MEMORY LOCATION
C016 E6 03           AINP     ANI      3          WE MUST BE REASONABLE
C018 E5                       PUSH     H          SAVE H&L
C019 21 8C C1                 LXI      H,ITAB     POINT TO TABLE
C01C 07                       RLC      .          THE MATH
C01D 85                       ADD      L
C01E 6F                       MOV      L,A        DONE
C01F C3 71 C0                 JMP      DISPT      WE HAVE THE ADDRESS...GO TO HEAVEN
C022                 *
C022                 *
C022                 *       KEYBOARD INPUT STATUS CHECK
C022                 *
C022                 *  THIS ROUTINE TESTS THE KEYBOARD STATUS AND RETURNS
C022                 *  WITH THE TEST BITS SET.
C022                 *
C022 DB FA           KSTAT    IN       STAPT      GET STATUS WORD
C024 E6 01                    ANI      KDR        TEST KEYBOARD BIT
C026 C9                       RET                 FLAGS ARE SET
C027                 *
C027                 *
C027                 *       KEYBOARD DATA INPUT
C027                 *
C027                 *  THIS ROUTINE, ENTRY AT KREAD, GETS THE DATA FROM THE
C027                 *  KEYBOARD.  ON RETURN THE CHARACTER IS IN REGISTER 'A'.
C027                 *
C027 E1              KREA1    POP      H          JUMP TABLE ENTRY POINT
C028 CD 22 C0        KREAD    CALL     KSTAT      CHECK STATUS
C02B C2 28 C0                 JNZ      KREAD      WAIT FOR INPUT
C02E DB FC                    IN       KDATA      GET DATA
C030 C9                       RET                 GO BACK WITH IT
C031                 *
C031                 *
C031                 *   SERIAL INPUT STATUS CHECK
C031                 *
C031 DB F8           SSTAT    IN       SERST      GET SERIAL STATUS WORD
C033 E6 40                    ANI      SDR        TEST FOR SERIAL DATA READY
C035 C9                       RET                 FLAGS ARE SET
C036                 *
C036                 *
C036                 *   SERIAL DATA INPUT
C036                 *
C036 E1              SREA1    POP      H          RESTORE HL FROM JUMP TABLE ENTRY
C037 CD 31 C0        SREAD    CALL     SSTAT      NORMAL ENTRY POINT
C03A CA 37 C0                 JZ       SREAD      WAIT FOR INPUT
C03D DB F9                    IN       SDATA      GET DATA BYTE
C03F C9                       RET                 WE HAVE IT
C040                 *
C040                 *
C040                 *   SERIAL DATA OUTPUT
C040                 *
C040 E1              SEROT    POP      H          JUMP TABLE ENTRY POINT
C041 DB F8           SDROT    IN       SERST      GET PORT STATUS
C043 17                       RAL      .          PUT HIGH BIT IN CARRY
C044 D2 41 C0                 JNC      SDROT      LOOP UNTIL TRANSMITTER BUFFER IS EMPTY
C047 78                       MOV      A,B        GET THE CHARACTER BACK
C048 D3 F9                    OUT      SDATA      SEND IT OUT
C04A C9                       RET                 AND WE'RE DONE
C04B                 *
C04B                 *
C04B                 *
C04B                 *
C04B                 *
```

```
C04B            0160 *            VIDEO DISPLAY DRIVER ROUTINES
C04B            0161 *
C04B            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   0337          JMP      PCUR
C121            0338 *
C121            0339 *   ROUTINE TO CALCULATE SCREEN ADDRESS
C121            0340 *
C121            0341 *   ENTRY AT:    RETURNS:
C121            0342 *
C121            0343 *          VDADD  CURRENT SCREEN ADDRESS
C121            0344 *          VDAD2  ADDRESS OF CURRENT LINE, CHAR 'C'
C121            0345 *          VDAD   LINE 'A', CHARACTER POSITION 'C'
C121            0346 *
C121 3A 00 C8   0347 VDADD    LDA      NCHAR       GET CHARACTER POSITION
C124 4F         0348          MOV      C,A         'C' KEEPS IT
C125 3A 01 C8   0349 VDAD2    LDA      LINE        LINE POSITION
C128 6F         0350 VDAD     MOV      L,A         INTO 'L'
C129 3A 02 C8   0351          LDA      BOT         GET TEXT OFFSET
C12C 85         0352          ADD      L           ADD IT TO THE LINE POSITION
C12D 0F         0353          RRC      .           TIMES TWO
C12E 0F         0354          RRC      .           MAKES FOUR
C12F 6F         0355          MOV      L,A         L HAS IT
C130 E6 03      0356          ANI      3           MOD THREE FOR LATER
C132 57         0357          MOV      D,A
C133 3E CC      0358          MVI      A,<VDMEM    LOW SCREEN OFFSET
C135 82         0359          ADD      D
C136 67         0360          MOV      H,A         NOW H IS DONE
C137 7D         0361          MOV      A,L         TWIST L'S ARM
C138 E6 C0      0362          ANI      0C0H
C13A 81         0363          ADD      C
C13B 6F         0364          MOV      L,A
C13C C9         0365          RET                  H & L ARE NOW PERVERTED
C13D            0366 *
C13D            0367 *    ROUTINE TO REMOVE CURSOR
C13D            0368 *
C13D CD 21 C1   0369 CREM     CALL     VDADD       GET CURRENT SCREEN ADDRESS
C140 7E         0370          MOV      A,M
C141 E6 7F      0371          ANI      7FH         STRIP OFF THE CURSOR
C143 77         0372          MOV      M,A
C144 C9         0373          RET
C145            0374 *
C145            0375 *     ROUTINE TO BACKSPACE
C145            0376 *
C145 CD 0D C1   0377 PBACK    CALL     PLEFT
C148 CD 21 C1   0378          CALL     VDADD       GET SCREEN ADDRESS
C14B 36 20      0379          MVI      M,' '       PUT A BLANK THERE
C14D C9         0380          RET
C14E            0381 *
C14E            0382 *     ROUTINE TO PROCESS A CARRIAGE RETURN
C14E            0383 *
C14E CD E3 C0   0384 PCR      CALL     CLINE       CLEAR FROM CURRENT CURSOR TO END OF LINE
C151 AF         0385          XRA      A           REWIND IT
C152 C3 13 C1   0386          JMP      PCUR        AND STORE THE NEW VALUE
C155            0387 *
C155            0388 *   ROUTINE TO PROCESS LINEFEED
C155            0389 *
C155 3A 01 C8   0390 PLF      LDA      LINE        GET LINE COUNT
C158 FE 0F      0391          CPI      15          ARE WE AT THE BOTTOM?
C15A D2 61 C1   0392          JNC      SC
C15D 3C         0393          INR      A
C15E C3 C1 C0   0394          JMP      CUR         ONE MORE LINE UP
C161            0395 *
C161 AF         0396 SC       XRA      A
C162 C3 9C C0   0397          JMP      SROL
C165            0398 *
C165            0399 *
C165            0400 *
C165            0401 *     THIS TABLE DEFINES THE CHARACTERS FOR SPECIAL
C165            0402 * PROCESSING.  IF THE CHARACTER IS NOT IN THE TABLE IT
C165            0403 * GOES TO THE SCREEN.
C165            0404 *
C165 8B         0405 TBL      DB       CLEAR       SCREEN
C166 C5 C0      0406          DW       PERSE
C168 97         0407          DB       UP          CURSOR
C169 04 C1      0408          DW       PUP
C16B 9A         0409          DB       DOWN        "
C16C FA C0      0410          DW       PDOWN
C16E 81         0411          DB       LEFT
C16F 0D C1      0412          DW       PLEFT
C171 93         0413          DB       RIGHT       "
C172 17 C1      0414          DW       PRIT
C174 8E         0415          DB       HOME        "
C175 F3 C0      0416          DW       PHOME
C177 0D         0417          DB       CR          CARRIAGE RETURN
C178 4E C1      0418          DW       PCR
C17A 0A         0419          DB       LF          LINE FEED
C17B 55 C1      0420          DW       PLF
C17D 5F         0421          DB       BACKS       BACK SPACE
C17E 45 C1      0422          DW       PBACK
C180 80         0423          DB       MODE        MODE KEY
C181 B4 C1      0424          DW       COMND
C183 00         0425          DB       0           END OF TABLE
C184            0426 *
C184            0427 *   OUTPUT DEVICE TABLE
C184            0428 *
C184 4C C0      0429 OTAB     DW       VDMO1       DISPLAY DRIVER
C186 40 C0      0430          DW       SEROT       SERIAL OUTPUT
C188 AD C1      0431          DW       ERROT       ERROR HANDLER (FOR CONSOL)
C18A AD C1      0432          DW       ERROT       ERROR HANDLER
C18C            0433 *
C18C            0434 *   INPUT DEVICE TABLE
C18C            0435 *
C18C 27 C0      0436 ITAB     DW       KREA1       KEYBOARD INPUT
C18E 36 C0      0437          DW       SREA1       SERIAL INPUT
C190 AD C1      0438          DW       ERROT       ERROR HANDLER (FOR CONSOL)
C192 AD C1      0439          DW       ERROT       ERROR HANDLER
C194            0440 *
C194            0441 *
C194            0442 *          COMMAND TABLE
C194            0443 *
C194            0444 *  THIS TABLE DESCRIBES THE VALID COMMANDS FOR CONSOL
C194            0445 *
C194 54 45      0446 COMTAB   ASC      'TE'
C196 70 C2      0447          DW       TERM
C198 44 55      0448          ASC      'DU'
C19A A1 C2      0449          DW       DUMP
C19C 45 4E      0450          ASC      'EN'
C19E 07 C3      0451          DW       ENTER
C1A0 45 58      0452          ASC      'EX'
C1A2 35 C3      0453          DW       EXEC
C1A4 54 4C      0454          ASC      'TL'
C1A6 39 C3      0455          DW       TLOAD
C1A8 42 41      0456          ASC      'BA'        SPECIAL COMMAND TO EXECUTE 0
C1AA 00 00      0457          DW       0
C1AC 00         0458          DB       0           END OF TABLE MARK
C1AD            0459 *
C1AD            0460 *
C1AD            0461 *           CONSOL PORT ERROR HANDLER
C1AD            0462 *
C1AD AF         0463 ERROT    XRA      A
C1AE 32 04 C8   0464          STA      IPORT       DEFAULT TO SCREEN
C1B1 32 03 C8   0465          STA      OPORT       DEFAULT TO SCREEN
C1B4            0466 *
C1B4            0467 *
C1B4            0468 *
C1B4            0469 *
C1B4            0470 *           =--  COMMAND MODE  --=
C1B4            0471 *
C1B4            0472 *
C1B4            0473 *   THIS ROUTINE GETS AND PROCESSES COMMANDS
C1B4            0474 *
C1B4 31 00 CC   0475 COMND    LXI      SP,SYSTP    SET STACK POINTER
C1B7 CD 12 C2   0476          CALL     PROMPT      PUT PROMPT ON SCREEN
C1BA CD C3 C1   0477          CALL     GCLIN       GET COMMAND LINE
C1BD CD E1 C1   0478          CALL     COPRC       PROCESS THE LINE
C1C0 C3 B4 C1   0479          JMP      COMND       OVER AND OVER
C1C3            0480 *
C1C3            0481 *
C1C3            0482 *
C1C3            0483 *   THIS ROUTINE READS A COMMAND LINE FROM THE SYSTEM
C1C3            0484 *  KEYBOARD AND PROCESSES IT TO THE SCREEN.
C1C3            0485 *
C1C3            0486 *  C/R   TERMINATES THE SEQUENCE ERASING ALL CHARS TO THE
C1C3            0487 *        RIGHT OF THE CURSOR
C1C3            0488 *  L/F   TERMINATES THE SEQUENCE
C1C3            0489 *  MODE  RESTARTS THE COMMAND LINE.
C1C3            0490 *
C1C3 CD 28 C0   0491 GCLIN    CALL     KREAD       READ KEYBOARD
C1C6 FE 20      0492          CPI      20H
C1C8 47         0493          MOV      B,A
C1C9 DA D2 C1   0494          JC       PROCS       PROCESS CONTROL CHARACTER
C1CC CD 4B C0   0495 CONT     CALL     VDMOT
C1CF C3 C3 C1   0496          JMP      GCLIN
C1D2            0497 *
C1D2            0498 *   PROCESS CONTROL KEYS
C1D2            0499 *
C1D2 FE 0D      0500 PROCS    CPI      CR
C1D4 CA DD C1   0501          JZ       CRPRC       ERASE THE REMAINING LINE PRIOR TO RETURN
C1D7 FE 0A      0502          CPI      LF
C1D9 C8         0503          RZ       .           IF SO GO PROCESS
C1DA C3 C3 C1   0504          JMP      GCLIN       NO CONTROL CHARS TO SCREEN
C1DD            0505 *
C1DD            0506 *
C1DD CD E3 C0   0507 CRPRC    CALL     CLINE       CLEAR REMAINING LINE
C1E0 C9         0508          RET                  NOW PROCESS
C1E1            0509 *
C1E1            0510 *
C1E1            0511 *      FIND AND PROCESS COMMAND
```

```
C1E1                0512 *
C1E1 CD 3D C1       0513 COPRC    CALL    CREM       REMOVE THE CURSOR
C1E4 0E 01          0514          MVI     C,1        SET FOR CHARACTER POSITION
C1E6 CD 25 C1       0515          CALL    VDAD2      GET SCREEN ADDRESS
C1E9 EB             0516          XCHG    .
C1EA CD 32 C2       0517          CALL    SCHR       SCAN PAST BLANKS
C1ED CA 8E C3       0518          JZ      ERR1       NO COMMAND?
C1F0 EB             0519          XCHG    .          HL HAS FIRST CHR
C1F1 11 94 C1       0520          LXI     D,COMTAB   POINT TO COMMAND TABLE
C1F4                0521 *
C1F4                0522 *  THIS ROUTINE SEARCHES THROUGH A TABLE, POINTED TO
C1F4                0523 * BY 'DE', FOR A DOUBLE CHARACTER MATCH OF THE 'HL'
C1F4                0524 * MEMORY CONTENT.  IF NO MATCH IS FOUND THE SCAN ENDS
C1F4                0525 * BY PLACING A QUESTION MARK WITHIN THE SEARCH STRING.
C1F4                0526 *
C1F4 1A             0527 FDCOM    LDAX    D
C1F5 B7             0528          ORA     A          TEST FOR TABLE END
C1F6 CA 8F C3       0529          JZ      ERR2       NOT FOUND..COMMAND ERROR
C1F9 E5             0530          PUSH    H          SAVE START OF SCAN ADDRESS
C1FA BE             0531          CMP     M          TEST FIRST CHR
C1FB 13             0532          INX     D
C1FC C2 0B C2       0533          JNZ     NCOM
C1FF                0534 *
C1FF 23             0535          INX     H
C200 1A             0536          LDAX    D
C201 BE             0537          CMP     M          NOW SECOND CHARACTER
C202 C2 0B C2       0538          JNZ     NCOM       GOODNESS
C205                0539 *
C205 C1             0540          POP     B          CLEAR THE STACK
C206 EB             0541          XCHG    .          DE HAS SCAN ADDRESS
C207 23             0542          INX     H          HL HAS COMMAND ADDRESS
C208 C3 71 C0       0543          JMP     DISPT      DISPATCH TO IT
C20B                0544 *
C20B                0545 *
C20B 13             0546 NCOM     INX     D          GO TO NEXT ENTRY
C20C 13             0547          INX     D
C20D 13             0548          INX     D
C20E E1             0549          POP     H          GET BACK ORIGINAL ADDRESS
C20F C3 F4 C1       0550          JMP     FDCOM      CONTINUE SEARCH
C212                0551 *
C212                0552 *
C212                0553 *      OUTPUT A CRLF FOLLOWED BY A PROMPT
C212                0554 *  ( WITH CONSOL ALL OPERATIONS ARE ON THE SCREEN)
C212                0555 *
C212 CD 1A C2       0556 PROMPT   CALL    CRLF
C215 06 3E          0557          MVI     B,'>'      THE PROMPT
C217 C3 4B C0       0558          JMP     VDMOT      PUT IT ON THE SCREEN
C21A                0559 *
C21A 06 0A          0560 CRLF     MVI     B,LF       LINE FEED
C21C CD 4B C0       0561          CALL    VDMOT
C21F 06 0D          0562          MVI     B,CR       CARRIAGE RETURN
C221 C3 4B C0       0563          JMP     VDMOT      PUT IT OUT AND RETURN
C224                0564 *
C224                0565 *
C224                0566 *  SCAN OVER UP TO 12 CHARACTERS LOOKING FOR A BLANK
C224                0567 *
C224 0E 0C          0568 SBLK     MVI     C,12       MAXIMUM COMMAND STRING
C226 1A             0569 SBLK1    LDAX    D
C227 FE 20          0570          CPI     BLANK
C229 CA 32 C2       0571          JZ      SCHR       GOT A BLANK NOW SCAN PAST IT
C22C 13             0572          INX     D
C22D 0D             0573          DCR     C          NO MORE THAN TWELVE
C22E C2 26 C2       0574          JNZ     SBLK1
C231 C9             0575          RET                GO BACK WITH ZERO FLAG SET
C232                0576 *
C232                0577 *
C232                0578 *  SCAN PAST UP TO 10 BLANK POSITIONS LOOKING FOR
C232                0579 * A NON BLANK CHARACTER.
C232                0580 *
C232 0E 0A          0581 SCHR     MVI     C,10       SCAN TO FIRST NON BLANK CHR WITHIN 10
C234 1A             0582 SCHR1    LDAX    D          GET NEXT CHARACTER
C235 FE 20          0583          CPI     SPACE
C237 C0             0584          RNZ     .          WE'RE PAST THEM
C238 13             0585          INX     D          NEXT SCAN ADDRESS
C239 0D             0586          DCR     C
C23A C8             0587          RZ      .          COMMAND ERROR
C23B C3 34 C2       0588          JMP     SCHR1      KEEP LOOPING
C23E                0589 *
C23E                0590 *  THIS ROUTINE SCANS OVER CHARACTERS, PAST BLANKS AND
C23E                0591 * CONVERTS THE FOLLOWING ADDRESS TO HEX.  ERRORS RETURN TO
C23E                0592 * THE ERROR HANDLER.
C23E                0593 *
C23E CD 24 C2       0594 SCONV    CALL    SBLK
C241 CA 8E C3       0595          JZ      ERR1
C244                0596 *
C244                0597 *  THIS ROUTINE CONVERTS ASCII DIGITS INTO BINARY FOLLOWING
C244                0598 * A STANDARD HEX CONVERSION.  THE SCAN STOPS WHEN AN ASCII
C244                0599 * SPACE IS ENCOUNTERED.  PARAMETER ERRORS REPLACE THE ERROR
C244                0600 * CHARACTER ON THE SCREEN WITH A QUESTION MARK.
C244                0601 *
C244 21 00 00       0602 SHEX     LXI     H,0        CLEAR H & L
C247 1A             0603 SHE1     LDAX    D          GET CHARACTER
C248 FE 20          0604          CPI     20H        IS IT A SPACE?
C24A C8             0605          RZ      .          IF SO
C24B                0606 *
C24B 29             0607 HCONV    DAD     H          MAKE ROOM FOR THE NEW ONE
C24C 29             0608          DAD     H
C24D 29             0609          DAD     H
C24E 29             0610          DAD     H
C24F CD 5B C2       0611          CALL    HCOV1      DO THE CONVERSION
C252 D2 8E C3       0612          JNC     ERR1       NOT VALID HEXIDECIMAL VALUE
C255 85             0613          ADD     L
C256 6F             0614          MOV     L,A        MOVE IT IN
C257 13             0615          INX     D          BUMP THE POINTER
C258 C3 47 C2       0616          JMP     SHE1
C25B                0617 *
C25B D6 30          0618 HCOV1    SUI     48         REMOVE ASCII BIAS
C25D FE 0A          0619          CPI     10
C25F D8             0620          RC      .          IF LESS THAN 9
C260 D6 07          0621          SUI     7          IT'S A LETTER??
C262 FE 10          0622          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 WAITING?
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       YUP!
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      Ø854          JNZ     RTBYT       STILL MORE IF NOT ZERO
C38Ø               Ø855 *
C38Ø CD C1 C3      Ø856          CALL    CRCCK       CHECK CRC AND FALL THROUGH TO ERROR IF NO GOOD
C383 CA 61 C3      Ø857          JZ      LOLOOP      TEST OK
C386               Ø858 *
C386 Ø6 Ø7         Ø859 TERR     MVI     B,'G'-4ØH   BELL CHARACTER
C388 CD 4B CØ      Ø86Ø          CALL    VDMOT       PUT IT ON THE SCREEN
C38B C3 B4 C1      Ø861          JMP     COMND
C38E               Ø862 *
C38E               Ø863 *
C38E               Ø864 *     CONSOL ERROR HANDLER
C38E               Ø865 *
C38E EB            Ø866 ERR1     XCHG    .           GET SCAN ADDRESS
C38F 36 3F         Ø867 ERR2     MVI     M,'?'       PUT A QUESTION MARK THERE
C391 C3 B4 C1      Ø868          JMP     COMND       AND GO TO COMMAND MODE
C394               Ø869 *
C394               Ø87Ø *    READ THE HEADER
C394               Ø871 *
C394 Ø6 ØA         Ø872 RHEAD    MVI     B,1Ø        FIND 1Ø NULLS
C396 DB FA         Ø873 RHEA1    IN      STAPT       GET A BYTE
C398 E6 4Ø         Ø874          ANI     TDR
C39A CA 96 C3      Ø875          JZ      RHEA1
C39D DB FB         Ø876          IN      TDATA       IGNORE ERROR CONDITIONS
C39F B7            Ø877          ORA     A           ZERO?
C3AØ C2 94 C3      Ø878          JNZ     RHEAD
C3A3 Ø5            Ø879          DCR     B
C3A4 C2 96 C3      Ø88Ø          JNZ     RHEA1       LOOP UNTIL 1Ø IN A ROW
C3A7               Ø881 *
C3A7               Ø882 *     WAIT FOR THE START CHARACTER
C3A7               Ø883 *
C3A7 CD C6 C3      Ø884 SOHL     CALL    TAPIN
C3AA 3D            Ø885          DCR     A
C3AB C2 A7 C3      Ø886          JNZ     SOHL        WAIT FOR A '1'
C3AE               Ø887 *
C3AE               Ø888 *     NOW GET THE HEADER
C3AE               Ø889 *
C3AE 21 Ø5 C8      Ø89Ø          LXI     H,THEAD     POINT TO BUFFER
C3B1 Ø1 ØØ 1Ø      Ø891          LXI     B,HLEN*256  LENGTH OF HEADER IN 'B',C<Ø
C3B4               Ø892 *
C3B4 CD C6 C3      Ø893 RHED1    CALL    TAPIN       GET BYTE
C3B7 77            Ø894          MOV     M,A         STORE IT
C3B8 23            Ø895          INX     H           INCREMENT ADDRESS
C3B9 A9            Ø896          XRA     C           NOW CALCULATE THE CRC
C3BA 2F            Ø897          CMA     .           INSIDE OUT AND UPSIDE DOWN
C3BB 91            Ø898          SUB     C           SQUEEZE IT
C3BC 4F            Ø899          MOV     C,A         AND SAVE AGAIN
C3BD Ø5            Ø9ØØ          DCR     B           WHOLE HEADER YET?
C3BE C2 B4 C3      Ø9Ø1          JNZ     RHED1       LOOP UNTIL DONE
C3C1               Ø9Ø2 *
C3C1               Ø9Ø3 *    THIS ROUTINE GETS THE NEXT BYTE AND COMPARES IT
C3C1               Ø9Ø4 * TO THE VALUE IN REGISTER C.  THE FLAGS ARE SET ON
C3C1               Ø9Ø5 * RETURN.
C3C1               Ø9Ø6 *
C3C1 CD C6 C3      Ø9Ø7 CRCCK    CALL    TAPIN       GET CRC BYTE
C3C4 B9            Ø9Ø8          CMP     C           COMPARE IT WITH CALCULATED
C3C5 C9            Ø9Ø9          RET
C3C6               Ø91Ø *
C3C6               Ø911 *
C3C6               Ø912 *     THIS ROUTINE GETS THE NEXT AVAILABLE BYTE FROM THE
C3C6               Ø913 *   TAPE.  WHILE WAITING FOR THE BYTE THE KEYBOARD IS TESTED
C3C6               Ø914 *   FOR A 'MODE' COMMAND.  IF RECEIVED THE TAPE LOAD IS
C3C6               Ø915 *   TERMINATED AND A RETURN TO THE COMMAND MODE IS MADE.
C3C6               Ø916 *
C3C6 DB FA         Ø917 TAPIN    IN      STAPT       CHECK STATUS
C3C8 E6 4Ø         Ø918          ANI     TDR
C3CA C2 D7 C3      Ø919          JNZ     TREDY       ONE IS AVAILABLE
C3CD DB FC         Ø92Ø          IN      KDATA       CHECK FOR MODE WHILE WE'RE WAITING
C3CF FE 8Ø         Ø921          CPI     MODE
C3D1 CA B4 C1      Ø922          JZ      COMND       MODE WAS GIVEN..ABORT OPERATION
C3D4 C3 C6 C3      Ø923          JMP     TAPIN       NOT MODE...STAY IN LOOP
C3D7               Ø924 *
C3D7 DB FA         Ø925 TREDY    IN      STAPT
C3D9 E6 18         Ø926          ANI     TFE+TOE     DATA ERROR?
C3DB C2 86 C3      Ø927          JNZ     TERR        IF FRAMING OR OVERRUN ERROR
C3DE DB FB         Ø928          IN      TDATA       GET THE DATA
C3EØ C9            Ø929          RET
C3E1               Ø93Ø *
C3E1               Ø931 * THIS ROUTINE CALCULATES THE LENGTH OF THE LAST BLOCK
C3E1               Ø932 *
C3E1 Ø1 FF FF      Ø933 LBLK     LXI     B,-1
C3E4 Ø9            Ø934          DAD     B           COMPLEMENT HL
C3E5 23            Ø935          INX     H           .......TWO'S
C3E6 45            Ø936          MOV     B,L         LENGTH TO REGISTER B
C3E7 21 ØØ ØØ      Ø937          LXI     H,Ø         TELL DE WE'RE DONE
C3EA C3 7Ø C3      Ø938          JMP     RDBLK       ONWARD TO THE END
C3ED               Ø939 *
C3ED               Ø94Ø *    THIS ROUTINE TURNS THE TAPE UNITS OFF
C3ED               Ø941 *
C3ED AF            Ø942 TOFF     XRA     A
C3EE D3 FA         Ø943          OUT     STAPT       GIVE COMMAND
C3FØ C9            Ø944          RET                 AND GRIND TO A SLOW STOP
C3F1               Ø945 *
C3F1               Ø946 *
C3F1 11 ØØ ØØ      Ø947 DELAY    LXI     D,Ø         START LOOP
C3F4 1B            Ø948 DLOP1    DCX     D           DOWN COUNT
C3F5 7A            Ø949          MOV     A,D
C3F6 B3            Ø95Ø          ORA     E           TEST FOR ZERO
C3F7 C2 F4 C3      Ø951          JNZ     DLOP1       IF NOT
C3FA C9            Ø952          RET
C3FB               Ø953 *
C3FB               Ø954 *
C3FB               Ø955 *
C3FB               Ø956 *
C3FB               Ø957 *                  <<  Sol SYSTEM EQUATES  >>
C3FB               Ø958 *
C3FB               Ø959 *
C3FB               Ø96Ø *          DISPLAY PARAMETERS
C3FB               Ø961 *
C3FB               Ø962 VDMEM    EQU     ØCCØØH      SCREEN MEMORY
C3FB               Ø963 *
C3FB               Ø964 *
C3FB               Ø965 *          KEYBOARD SPECIAL KEY ASSIGNMENTS
C3FB               Ø966 *
C3FB               Ø967 DOWN     EQU     9AH
C3FB               Ø968 UP       EQU     97H
C3FB               Ø969 LEFT     EQU     81H
C3FB               Ø97Ø RIGHT    EQU     93H
C3FB               Ø971 LOADK    EQU     8CH         LOAD KEY
C3FB               Ø972 MODE     EQU     8ØH
C3FB               Ø973 CLEAR    EQU     8BH
C3FB               Ø974 HOME     EQU     Ø8EH
C3FB               Ø975 BACKS    EQU     5FH         BACKSPACE
C3FB               Ø976 LF       EQU     1Ø
C3FB               Ø977 CR       EQU     13
C3FB               Ø978 BLANK    EQU     ' '
C3FB               Ø979 SPACE    EQU     BLANK
C3FB               Ø98Ø CX       EQU     'X'-4ØH
C3FB               Ø981 *
C3FB               Ø982 *          PORT ASSIGNMENTS
C3FB               Ø983 *
C3FB               Ø984 STAPT    EQU     ØFAH        STATUS PORT GENERAL
C3FB               Ø985 SERST    EQU     ØF8H        SERIAL STATUS PORT
C3FB               Ø986 SDATA    EQU     ØF9H        SERIAL DATA
C3FB               Ø987 TDATA    EQU     ØFBH        TAPE DATA
C3FB               Ø988 KDATA    EQU     ØFCH        KEYBOARD DATA
C3FB               Ø989 PDATA    EQU     ØFDH        PARALLEL DATA
C3FB               Ø99Ø DSTAT    EQU     ØFEH        DISPLAY PARAMETER PORT
C3FB               Ø991 SENSE    EQU     ØFFH        SENSE SWITCHES
C3FB               Ø992 *
C3FB               Ø993 *
C3FB               Ø994 *
C3FB               Ø995 *          BIT ASSIGNMENT MASKS
C3FB               Ø996 *
C3FB               Ø997 SCD      EQU     1           SERIAL CARRIER DETECT
C3FB               Ø998 SDSR     EQU     2           SERIAL DATA SET READY
C3FB               Ø999 SPE      EQU     4           SERIAL PARITY ERROR
C3FB               1ØØØ SFE      EQU     8           SERIAL FRAMING ERROR
C3FB               1ØØ1 SOE      EQU     16          SERIAL OVERRUN ERROR
C3FB               1ØØ2 SCTS     EQU     32          SERIAL CLEAR TO SEND
C3FB               1ØØ3 SDR      EQU     64          SERIAL DATA READY
C3FB               1ØØ4 STBE     EQU     128         SERIAL TRANSMITTER BUFFER EMPTY
C3FB               1ØØ5 *
C3FB               1ØØ6 KDR      EQU     1           KEYBOARD DATA READY
C3FB               1ØØ7 PDR      EQU     2           PARALLEL DATA READY
C3FB               1ØØ8 PXDR     EQU     4           PARALLEL DEVICE READY
C3FB               1ØØ9 TFE      EQU     8           TAPE FRAMING ERROR
C3FB               1Ø1Ø TOE      EQU     16          TAPE OVERFLOW ERROR
C3FB               1Ø11 TDR      EQU     64          TAPE DATA READY
C3FB               1Ø12 TTBE     EQU     128         TAPE TRANSMITTER BUFFER EMPTY
C3FB               1Ø13 *
C3FB               1Ø14 SOK      EQU     1           SCROLL OK FLAG
C3FB               1Ø15 *
C3FB               1Ø16 TAPE1    EQU     64          TAPE ONE 'ON' BIT
C3FB               1Ø17 TAPE2    EQU     128         TAPE TWO
C3FB               1Ø18 *
C3FB               1Ø19 *
C3FB               1Ø2Ø *
C3FB               1Ø21 *
C3FB               1Ø22 *               Sol SYSTEM GLOBAL AREA
C3FB               1Ø23 *
C3FB               1Ø24          ORG     ØC8ØØH      START OF 1K RAM AREA
C8ØØ               1Ø25 *
C8ØØ               1Ø26 SYSRAM   EQU     $           START OF SYSTEM RAM
C8ØØ               1Ø27 SYSTP    EQU     $+1Ø24      STACK IS AT THE TOP
```

```
C8ØØ               1Ø28 *
C8ØØ               1Ø29 *
C8ØØ               1Ø3Ø *             CONSOL PARAMETER AREA
C8ØØ               1Ø31 *
C8ØØ               1Ø32 NCHAR    DS      1           CURRENT CHARACTER POSITION
C8Ø1               1Ø33 LINE     DS      1           CURRENT LINE POSITION
C8Ø2               1Ø34 BOT      DS      1           BEGINNING OF TEXT DISPLACEMENT
C8Ø3               1Ø35 OPORT    DS      1           OUTPUT PORT
C8Ø4               1Ø36 IPORT    DS      1           INPUT PORT
C8Ø5               1Ø37 *
C8Ø5               1Ø38 *
C8Ø5               1Ø39 *
C8Ø5               1Ø4Ø THEAD    DS      5           NAME
C8ØA               1Ø41          DS      1           THIS BYTE MUST BE ZERO
C8ØB               1Ø42 HTYPE    DS      1           TYPE
C8ØC               1Ø43 BLOCK    DS      2           BLOCK SIZE
C8ØE               1Ø44 LOADR    DS      2           LOAD ADDRESS
C81Ø               1Ø45 XEQAD    DS      2           AUTO EXECUTE ADDRESS
C812               1Ø46 HSPR     DS      3           SPARES
C815               1Ø47 *
C815               1Ø48 HLEN     EQU     $-THEAD     LENGTH OF HEADER
C815               1Ø49 *
C815               1Ø5Ø *
*
```

LET'S HEAR FROM YOU

Your letters are the lifeblood for us

We really do appreciate your letters and comments about INTERFACE AGE. We need to know what you think about the publication so that we know in what ways we can make it better.

If you like it, tell us, and if there are things you think could be improved, tell us that, too.

We do read all our mail and we appreciate your feedback.

The Editors