

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 editing 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 Software 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 Software to be loaded and run in a SoL System with additional memory. SoL System Software includes BASIC, FOCAL, a Scientific Calculator and numerous "game" packages including an 8K assembly language version of STARTREK called TREK80.

CONSOL 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

SOFTWARE SECTION

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 via the keyboard. The SoL is now operating as a computer and is ready to accept one of the following commands:

D Ump	Dump memory locations to screen
E Nter	Enter data to memory
E Xecute	Execute a program in external memory
B Asic	Execute a program located at address zero
T ErminAl	Return to terminal mode
T Load	Load program or data from cassette tape
M ODE	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

MICROCOMPUTER DEVELOPMENT SOFTWARE

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 specified;

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

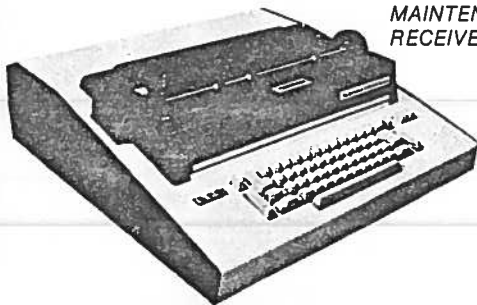
The enter command places sequential bytes into memory beginning at the specified address. Data, represented as hexadecimal values, are entered from the keyboard for storage in memory. Values are entered one line at a time with each line terminated by a carriage return or linefeed. The ENTER command function 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

RONDURE COMPANY

Where We Ship from Inventory the Same Day Your Order Arrives*

A SELECTRIC TERMINAL COMPLETE WITH RS-232/C INTERFACE AND CERTIFIED FOR MAINTENANCE BY A NATIONAL SERVICE COMPANY. SHIPPED THE SAME DAY WE RECEIVE YOUR CHECK*



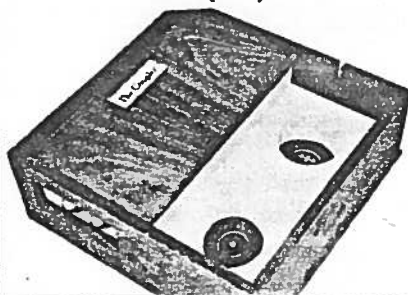
\$895⁰⁰

TERMS: Check or Money Order. For Modems, Base Keyboard, Switch Blk., add \$2.00 shipping and handling. All others shipping packaging and shipping collect.

*Maintenance limited to cities in which service now offered. Shipped the same day as certified check or money order arrives. When regular checks accompany order, equipment is shipped when regular check clears.

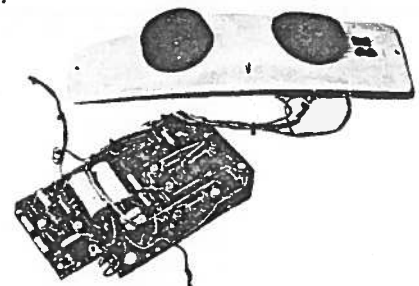
NEW ADDRESS
2522 BUTLER, DALLAS, TEXAS 75235
Phone: (214) 630-4621

ALSO NOTE: NO EQUIPMENT INCLUDES PRINTS OR DOCUMENTATION (unless stated). NO CONNECTING CORDS OR CONNECTORS. EQUIPMENT IS SHIPPED ON AN AS IS WHERE IS — BASIS. EXCEPT WHERE EXPRESSLY STATED IN WRITING, NO REPRESENTATION OR WARRANTY IS MADE AS TO THE QUALITY, CONDITION OR WORKING ORDER OF ANY EQUIPMENT OR PART.



LEFT: ACOUSTICAL MODEMS
* ORIGINATE ONLY * USED *
UNTESTED * IN WOOD ENCLOSURE
\$20.00ea. 2 for \$35

RIGHT: ACOUSTICAL MODEMS
* ORIGINATE ONLY * USED *
UNTESTED * Physically fit into Model 33
Teletype. Manufactured by Paragon Par-
tial Documentation 2 for \$25



the left to correct the error.

TLoad <speed>

CONSOL includes routines to read standardized Software 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 Software 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.

Header: ** ALS-3 PROGRAM DEVELOPMENT SYSTEM **
PROCESSOR TECHNOLOGY CORP.
6289 HOLLIS STREET
EMERYVILLE, CALIF. 94608
CONSOL
COPYRIGHT 1976

SYSTEM SOFTWARE
<<>> CONSOL <<>>
VERSION: 1.015
RELEASE: 10.11.76
THIS PROGRAM IS THE MINIMUM SoL OPERATING SYSTEM.
IT PROVIDES ALL NECESSARY DISPLAY ROUTINES ALONG WITH
STANDARDIZED CALLING POINTS FOR INPUT/OUTPUT OPERATIONS.
COMMANDS ARE ALSO AVAILABLE TO ENTER DATA TO MEMORY
AND TO DISPLAY MEMORY DATA ON THE SCREEN. AN EXECUTE
COMMAND IS PROVIDED TO ALLOW PROGRAM EXECUTION OUTSIDE
OF THE CONSOL PROM AND PROVISION IS MADE FOR CASSETTE
TAPE LOAD OF PROGRAMS OR DATA.
NOTE: CONSOL,SOLOS AND SOLED ARE REGISTERED TRADEMARKS
OF:
PROCESSOR TECHNOLOGY CORP.
EMERYVILLE,CALIF
THE FOLLOWING CODE IS THE PROPERTY OF PROCESSOR
TECHNOLOGY CORP. IT IS DISTRIBUTED ON A "PERSONAL USE"
BASIS FOR THE BENEFIT OF SoL SYSTEM OWNERS. ALL FORMS
OF THE CODE ARE COPYRIGHTED 1976 BY PROCESSOR TECHNOLOGY
AND ALL RIGHTS THEREIN ARE RESERVED.
AUTO-STARTUP CODE
DB S FOUR PHASE WONDER
JMP S STRTA
----- SoL SYSTEM I/O ROUTINES -----
THE FOLLOWING CODE IS STANDARDIZED FOR ALL SoL SYSTEM
SOFTWARE IT PROVIDES COMMON ENTRY POINTS FOR INPUT AND
OUTPUT OPERATIONS. CONSOL DOES NOT MAKE PROVISION FOR
PARALLEL I/O OPERATIONS BECAUSE OF SPACE LIMITATIONS.
JUMP TABLE INPUT/OUTPUT ROUTINES
THIS ROUTINE OUTPUTS THE CHARACTER IN REGISTER 'B' TO
THE OUTPUT DEVICE POINTED TO BY THE CURRENT OUTPUT SELECT
REGISTER. THE DEVICES ARE DEFINED AS FOLLOWS:
0 - DIS SCREEN
1 - SERIAL OUTPUT PORT
2 - PARALLEL OUTPUT PORT (NOT AVAILABLE ON CONSOL)
3 - ERROR HANDLER
ENTRY AT: SOUT SELECTS CURRENT OUTPUT DEVICE
AOUT SELECTS DEVICE IN REGISTER 'A'
SOUT SOUT LDA SOUT GET PORT NUMBER FROM MEMORY LOCATION
AOUT AOUT AND J KEEP IT IN CONTROL
PUSH H H L.L. ADDRESS IF LATER
LXI H,UTAB POINT TO TABLE
MVC L COMPUTE ADDRESS
ADJ L
MOV L,A WE HAVE IT
JMP DISPT GO TO BL....
THIS ROUTINE INPUTS A CHARACTER TO REGISTER 'A' FROM
THE CURRENT INPUT DEVICE POINTED TO BY THE CURRENT INPUT
SELECT REGISTER.
ENTRY POINTS ARE DEFINED:
0 - KEYBOARD INPUT
1 - SERIAL INPUT
2 - PARALLEL INPUT (NOT AVAILABLE ON CONSOL)
3 - ERROR HANDLER
SIMP LDA SIMP GET PORT NUMBER FROM MEMORY LOCATION
SIMP SIMP AND J WE MUST BE REASONABLE
PUSH H SAVE HL
MVAR H POINT TO TABLE
MVC L THE RATE
ADD L
MOV L,A DONE
JMP DISPT WE HAVE THE ADDRESS...GO TO BLAVEN
KEYBOARD INPUT STATUS CHECK
THIS ROUTINE TESTS THE KEYBOARD STATUS AND RETURNS
WITH THE TEST BITS SET.
KSTAT IN STAYT GET STATUS WORD
AND KDR TEST KEYBOARD BIT
MVC FLGZ ARE SET
KEYBOARD DATA INPUT
THIS ROUTINE, ENTRY AT SMDAT, GETS THE DATA FROM THE
KEYBOARD. ON RETURN THE CHARACTER IS IN REGISTER 'A'
SMDAT POP H JUMP TABLE ENTRY POINT
CALL SSTAT CHECK STATUS
JMP SREAD WAIT FOR INPUT
IN EDATA GET DATA
RET GO BACK WITH IT
SERIAL INPUT STATUS CHECK
SSTAT IN SSTAT GET SERIAL STATUS WORD
AND SDR TEST FOR SERIAL DATA READY
MVC FLGZ ARE SET
SERIAL DATA INPUT
SSTAT POP H RESTORE HL FROM JUMP TABLE ENTRY
CALL SSTAT NORMAL ENTRY POINT
JMP SREAD WAIT FOR INPUT
IN SDATA GET DATA BYTE
WE HAVE IT
SERIAL DATA OUTPUT
SSTAT POP H JUMP TABLE ENTRY POINT
IN SSTAT GET PORT STATUS
BAL L PUT HIGH BIT IN CARRY
LOOP UNTIL TRANSMITTER BUFFER IS EMPTY
MOV A,B GET THE CHARACTER BALE
OUT SDATA SEND IT OUT
RET AND WE'RE DONE

-ORG C0 04

SOFTWARE SECTION

MICROCOMPUTER DEVELOPMENT SOFTWARE

C960 0160 * VIDEO DISPLAY DRIVER ROUTINES
 C961 0161 *
 C962 0162 *
 C963 0163 * THESE ROUTINES ALLOW FOR STANDARD VIDEO TERMINAL
 C964 0164 * OPERATIONS. ON ENTRY, THE CHARACTER FOR OUTPUT IS IN
 C965 0165 * REGISTER B AND ALL REGISTERS ARE UNALTERED ON RETURN.
 C966 0166 * THE 'CONSOLE' VERSION OF THIS ROUTINE IS A MINOR
 C967 0167 * IMPLEMENTATION OF ROUTINES ORIGINATED BY:
 C968 0168 * IAN KETTLERBROUGH
 C969 0169 * OF
 C970 0170 * COLLEGE STATION, TEXAS
 C971 0171 *
 C972 0172 * SOLOS AND SOLES CONTAIN THE ESC SEQUENCES AND OTHER
 C973 0173 * FULL IMPLEMENTATION FEATURES.
 C974 0174 *
 C975 0175 *
 C976 0176 *
 C977 0177 *
 C978 0178 *
 C979 0179 *
 C980 0180 *
 C981 0181 *
 C982 0182 *
 C983 0183 *
 C984 0184 *
 C985 0185 *
 C986 0186 *
 C987 0187 *
 C988 0188 *
 C989 0189 *
 C990 0190 *
 C991 0191 *
 C992 0192 *
 C993 0193 *
 C994 0194 *
 C995 0195 *
 C996 0196 *
 C997 0197 *
 C998 0198 *
 C999 0199 *
 C1000 0200 *
 C1001 0201 *
 C1002 0202 *
 C1003 0203 *
 C1004 0204 *
 C1005 0205 *
 C1006 0206 *
 C1007 0207 *
 C1008 0208 *
 C1009 0209 *
 C1010 0210 *
 C1011 0211 *
 C1012 0212 *
 C1013 0213 *
 C1014 0214 *
 C1015 0215 *
 C1016 0216 *
 C1017 0217 *
 C1018 0218 *
 C1019 0219 *
 C1020 0220 *
 C1021 0221 *
 C1022 0222 *
 C1023 0223 *
 C1024 0224 *
 C1025 0225 *
 C1026 0226 *
 C1027 0227 *
 C1028 0228 *
 C1029 0229 *
 C1030 0230 *
 C1031 0231 *
 C1032 0232 *
 C1033 0233 *
 C1034 0234 *
 C1035 0235 *
 C1036 0236 *
 C1037 0237 *
 C1038 0238 *
 C1039 0239 *
 C1040 0240 *
 C1041 0241 *
 C1042 0242 *
 C1043 0243 *
 C1044 0244 *
 C1045 0245 *
 C1046 0246 *
 C1047 0247 *
 C1048 0248 *
 C1049 0249 *
 C1050 0250 *
 C1051 0251 *
 C1052 0252 *
 C1053 0253 *
 C1054 0254 *
 C1055 0255 *
 C1056 0256 *
 C1057 0257 *
 C1058 0258 *
 C1059 0259 *
 C1060 0260 *
 C1061 0261 *
 C1062 0262 *
 C1063 0263 *
 C1064 0264 *
 C1065 0265 *
 C1066 0266 *
 C1067 0267 *
 C1068 0268 *
 C1069 0269 *
 C1070 0270 *
 C1071 0271 *
 C1072 0272 *
 C1073 0273 *
 C1074 0274 *
 C1075 0275 *
 C1076 0276 *
 C1077 0277 *
 C1078 0278 *
 C1079 0279 *
 C1080 0280 *
 C1081 0281 *
 C1082 0282 *
 C1083 0283 *
 C1084 0284 *
 C1085 0285 *
 C1086 0286 *
 C1087 0287 *
 C1088 0288 *
 C1089 0289 *
 C1090 0290 *
 C1091 0291 *
 C1092 0292 *
 C1093 0293 *
 C1094 0294 *
 C1095 0295 *
 C1096 0296 *
 C1097 0297 *
 C1098 0298 *
 C1099 0299 *
 C1100 0300 *
 C1101 0301 *
 C1102 0302 *
 C1103 0303 *
 C1104 0304 *
 C1105 0305 *
 C1106 0306 *
 C1107 0307 *
 C1108 0308 *
 C1109 0309 *
 C1110 0310 *
 C1111 0311 *
 C1112 0312 *
 C1113 0313 *
 C1114 0314 *
 C1115 0315 *
 C1116 0316 *
 C1117 0317 *
 C1118 0318 *
 C1119 0319 *
 C1120 0320 *
 C1121 0321 *
 C1122 0322 *
 C1123 0323 *
 C1124 0324 *
 C1125 0325 *
 C1126 0326 *
 C1127 0327 *
 C1128 0328 *
 C1129 0329 *
 C1130 0330 *
 C1131 0331 *
 C1132 0332 *
 C1133 0333 *
 C1134 0334 *

C1135 0335 *
 C1136 0336 *
 C1137 0337 *
 C1138 0338 *
 C1139 0339 *
 C1140 0340 *
 C1141 0341 *
 C1142 0342 *
 C1143 0343 *
 C1144 0344 *
 C1145 0345 *
 C1146 0346 *
 C1147 0347 *
 C1148 0348 *
 C1149 0349 *
 C1150 0350 *
 C1151 0351 *
 C1152 0352 *
 C1153 0353 *
 C1154 0354 *
 C1155 0355 *
 C1156 0356 *
 C1157 0357 *
 C1158 0358 *
 C1159 0359 *
 C1160 0360 *
 C1161 0361 *
 C1162 0362 *
 C1163 0363 *
 C1164 0364 *
 C1165 0365 *
 C1166 0366 *
 C1167 0367 *
 C1168 0368 *
 C1169 0369 *
 C1170 0370 *
 C1171 0371 *
 C1172 0372 *
 C1173 0373 *
 C1174 0374 *
 C1175 0375 *
 C1176 0376 *
 C1177 0377 *
 C1178 0378 *
 C1179 0379 *
 C1180 0380 *
 C1181 0381 *
 C1182 0382 *
 C1183 0383 *
 C1184 0384 *
 C1185 0385 *
 C1186 0386 *
 C1187 0387 *
 C1188 0388 *
 C1189 0389 *
 C1190 0390 *
 C1191 0391 *
 C1192 0392 *
 C1193 0393 *
 C1194 0394 *
 C1195 0395 *
 C1196 0396 *
 C1197 0397 *
 C1198 0398 *
 C1199 0399 *
 C1200 0400 *
 C1201 0401 *
 C1202 0402 *
 C1203 0403 *
 C1204 0404 *
 C1205 0405 *
 C1206 0406 *
 C1207 0407 *
 C1208 0408 *
 C1209 0409 *
 C1210 0410 *
 C1211 0411 *
 C1212 0412 *
 C1213 0413 *
 C1214 0414 *
 C1215 0415 *
 C1216 0416 *
 C1217 0417 *
 C1218 0418 *
 C1219 0419 *
 C1220 0420 *
 C1221 0421 *
 C1222 0422 *
 C1223 0423 *
 C1224 0424 *
 C1225 0425 *
 C1226 0426 *
 C1227 0427 *
 C1228 0428 *
 C1229 0429 *
 C1230 0430 *
 C1231 0431 *
 C1232 0432 *
 C1233 0433 *
 C1234 0434 *
 C1235 0435 *
 C1236 0436 *
 C1237 0437 *
 C1238 0438 *
 C1239 0439 *
 C1240 0440 *
 C1241 0441 *
 C1242 0442 *
 C1243 0443 *
 C1244 0444 *
 C1245 0445 *
 C1246 0446 *
 C1247 0447 *
 C1248 0448 *
 C1249 0449 *
 C1250 0450 *
 C1251 0451 *
 C1252 0452 *
 C1253 0453 *
 C1254 0454 *
 C1255 0455 *
 C1256 0456 *
 C1257 0457 *
 C1258 0458 *
 C1259 0459 *
 C1260 0460 *
 C1261 0461 *
 C1262 0462 *
 C1263 0463 *
 C1264 0464 *
 C1265 0465 *
 C1266 0466 *
 C1267 0467 *
 C1268 0468 *
 C1269 0469 *
 C1270 0470 *
 C1271 0471 *
 C1272 0472 *
 C1273 0473 *
 C1274 0474 *
 C1275 0475 *
 C1276 0476 *
 C1277 0477 *
 C1278 0478 *
 C1279 0479 *
 C1280 0480 *
 C1281 0481 *
 C1282 0482 *
 C1283 0483 *
 C1284 0484 *
 C1285 0485 *
 C1286 0486 *
 C1287 0487 *
 C1288 0488 *
 C1289 0489 *
 C1290 0490 *
 C1291 0491 *
 C1292 0492 *
 C1293 0493 *
 C1294 0494 *
 C1295 0495 *
 C1296 0496 *
 C1297 0497 *
 C1298 0498 *
 C1299 0499 *
 C1300 0500 *
 C1301 0501 *
 C1302 0502 *
 C1303 0503 *
 C1304 0504 *
 C1305 0505 *
 C1306 0506 *
 C1307 0507 *
 C1308 0508 *
 C1309 0509 *
 C1310 0510 *
 C1311 0511 *
 C1312 0512 *
 C1313 0513 *
 C1314 0514 *
 C1315 0515 *
 C1316 0516 *
 C1317 0517 *
 C1318 0518 *
 C1319 0519 *
 C1320 0520 *
 C1321 0521 *
 C1322 0522 *
 C1323 0523 *
 C1324 0524 *
 C1325 0525 *
 C1326 0526 *
 C1327 0527 *
 C1328 0528 *
 C1329 0529 *
 C1330 0530 *
 C1331 0531 *
 C1332 0532 *
 C1333 0533 *
 C1334 0534 *
 C1335 0535 *
 C1336 0536 *
 C1337 0537 *
 C1338 0538 *
 C1339 0539 *
 C1340 0540 *
 C1341 0541 *
 C1342 0542 *
 C1343 0543 *
 C1344 0544 *
 C1345 0545 *
 C1346 0546 *
 C1347 0547 *
 C1348 0548 *
 C1349 0549 *
 C1350 0550 *
 C1351 0551 *
 C1352 0552 *
 C1353 0553 *
 C1354 0554 *
 C1355 0555 *
 C1356 0556 *
 C1357 0557 *
 C1358 0558 *
 C1359 0559 *
 C1360 0560 *
 C1361 0561 *
 C1362 0562 *
 C1363 0563 *
 C1364 0564 *
 C1365 0565 *
 C1366 0566 *
 C1367 0567 *
 C1368 0568 *
 C1369 0569 *
 C1370 0570 *
 C1371 0571 *
 C1372 0572 *
 C1373 0573 *
 C1374 0574 *
 C1375 0575 *
 C1376 0576 *
 C1377 0577 *
 C1378 0578 *
 C1379 0579 *
 C1380 0580 *
 C1381 0581 *
 C1382 0582 *
 C1383 0583 *
 C1384 0584 *
 C1385 0585 *
 C1386 0586 *
 C1387 0587 *
 C1388 0588 *
 C1389 0589 *
 C1390 0590 *
 C1391 0591 *
 C1392 0592 *
 C1393 0593 *
 C1394 0594 *
 C1395 0595 *
 C1396 0596 *
 C1397 0597 *
 C1398 0598 *
 C1399 0599 *
 C1400 0600 *

A0VDMEM
SHR 8

CH TD

02 TB

02 TB

02 TB

02 TB

02 TB

02 TB

02 TB

02 TB

02 TB

02 TB

02 TB

02 TB

02 TB

02 TB

02 TB

02 TB

02 TB

C370 C2 73 C3	0034	JNE	NULL	STILL NULL IF NOT READ	C068	1020 *		
C368	0035 *				C069	1020 *		
C366 CD C3 C3	0036	CALL	CACOK	CHECK CAC AND FALL THROUGH TO ERROR IF NO GOOD	C069	1020 *		
C363 CA 61 L3	0037	JS	LDLOUP	TEST US	C069	1031 *		
C364	0038 *				C069	1032 MCHAR	DS	1
C365 06 47	0039	LD	R, G, 4-40	CALL CHARACTER	C069	1033 LINE	DS	1
C368 06 48 C6	0040	CALL	VMOVE	PUT IT ON THE SCREEN	C069	1034 M21	DS	1
C366 C1 84 C1	0041	JMP	CONWD		C069	1035 UPMS1	DS	1
C364	0042 *				C069	1036 IPMT	DS	1
C364	0043 *				C069	1037 *		
C364	0044 *				C069	1038 *		
C364	0045 *				C069	1039 *		
C364 04	0046 ENH1	XLMS		GET SCAN ADDRESS	C069	1040 HEAD	DS	5
L367 36 3F	0047 LRA2	JNE	M, 7	PUT A QUESTION MARK THERE	C069	1041	DS	1
L361 C3 04 L1	0048	JNE	CONWD	AND GO TO COMMAND RUE	C069	1042 WTYPE	DS	1
L364	0049 *				C069	1043 BLCK	DS	2
C364	0050 *				C069	1044 LMDM	DS	2
C364 04 04	0051	LD	R, 10	FIND 10 NULLS	C069	1045 -LEAD	DS	2
C364 04 04	0052	LD	R, 10	FIND 10 NULLS	C069	1046 WSP	DS	3
C364 04 04	0053	LD	R, 10	FIND 10 NULLS	C069	1047 *		
C364 04 04	0054	LD	R, 10	FIND 10 NULLS	C069	1048 MLEN	DS	1
C364 04 04	0055	LD	R, 10	FIND 10 NULLS	C069	1049 *		
C364 04 04	0056	LD	R, 10	FIND 10 NULLS	C069	1050 *		
C364 04 04	0057	LD	R, 10	FIND 10 NULLS	C069	1051 *		
C364 04 04	0058	LD	R, 10	FIND 10 NULLS	C069	1052 *		
C364 04 04	0059	LD	R, 10	FIND 10 NULLS	C069	1053 *		
C364 04 04	0060	LD	R, 10	FIND 10 NULLS	C069	1054 *		
C364 04 04	0061	LD	R, 10	FIND 10 NULLS	C069	1055 *		
C364 04 04	0062	LD	R, 10	FIND 10 NULLS	C069	1056 *		
C364 04 04	0063	LD	R, 10	FIND 10 NULLS	C069	1057 *		
C364 04 04	0064	LD	R, 10	FIND 10 NULLS	C069	1058 *		
C364 04 04	0065	LD	R, 10	FIND 10 NULLS	C069	1059 *		
C364 04 04	0066	LD	R, 10	FIND 10 NULLS	C069	1060 *		
C364 04 04	0067	LD	R, 10	FIND 10 NULLS	C069	1061 *		
C364 04 04	0068	LD	R, 10	FIND 10 NULLS	C069	1062 *		
C364 04 04	0069	LD	R, 10	FIND 10 NULLS	C069	1063 *		
C364 04 04	0070	LD	R, 10	FIND 10 NULLS	C069	1064 *		
C364 04 04	0071	LD	R, 10	FIND 10 NULLS	C069	1065 *		
C364 04 04	0072	LD	R, 10	FIND 10 NULLS	C069	1066 *		
C364 04 04	0073	LD	R, 10	FIND 10 NULLS	C069	1067 *		
C364 04 04	0074	LD	R, 10	FIND 10 NULLS	C069	1068 *		
C364 04 04	0075	LD	R, 10	FIND 10 NULLS	C069	1069 *		
C364 04 04	0076	LD	R, 10	FIND 10 NULLS	C069	1070 *		
C364 04 04	0077	LD	R, 10	FIND 10 NULLS	C069	1071 *		
C364 04 04	0078	LD	R, 10	FIND 10 NULLS	C069	1072 *		
C364 04 04	0079	LD	R, 10	FIND 10 NULLS	C069	1073 *		
C364 04 04	0080	LD	R, 10	FIND 10 NULLS	C069	1074 *		
C364 04 04	0081	LD	R, 10	FIND 10 NULLS	C069	1075 *		
C364 04 04	0082	LD	R, 10	FIND 10 NULLS	C069	1076 *		
C364 04 04	0083	LD	R, 10	FIND 10 NULLS	C069	1077 *		
C364 04 04	0084	LD	R, 10	FIND 10 NULLS	C069	1078 *		
C364 04 04	0085	LD	R, 10	FIND 10 NULLS	C069	1079 *		
C364 04 04	0086	LD	R, 10	FIND 10 NULLS	C069	1080 *		
C364 04 04	0087	LD	R, 10	FIND 10 NULLS	C069	1081 *		
C364 04 04	0088	LD	R, 10	FIND 10 NULLS	C069	1082 *		
C364 04 04	0089	LD	R, 10	FIND 10 NULLS	C069	1083 *		
C364 04 04	0090	LD	R, 10	FIND 10 NULLS	C069	1084 *		
C364 04 04	0091	LD	R, 10	FIND 10 NULLS	C069	1085 *		
C364 04 04	0092	LD	R, 10	FIND 10 NULLS	C069	1086 *		
C364 04 04	0093	LD	R, 10	FIND 10 NULLS	C069	1087 *		
C364 04 04	0094	LD	R, 10	FIND 10 NULLS	C069	1088 *		
C364 04 04	0095	LD	R, 10	FIND 10 NULLS	C069	1089 *		
C364 04 04	0096	LD	R, 10	FIND 10 NULLS	C069	1090 *		
C364 04 04	0097	LD	R, 10	FIND 10 NULLS	C069	1091 *		
C364 04 04	0098	LD	R, 10	FIND 10 NULLS	C069	1092 *		
C364 04 04	0099	LD	R, 10	FIND 10 NULLS	C069	1093 *		
C364 04 04	0100	LD	R, 10	FIND 10 NULLS	C069	1094 *		
C364 04 04	0101	LD	R, 10	FIND 10 NULLS	C069	1095 *		
C364 04 04	0102	LD	R, 10	FIND 10 NULLS	C069	1096 *		
C364 04 04	0103	LD	R, 10	FIND 10 NULLS	C069	1097 *		
C364 04 04	0104	LD	R, 10	FIND 10 NULLS	C069	1098 *		
C364 04 04	0105	LD	R, 10	FIND 10 NULLS	C069	1099 *		
C364 04 04	0106	LD	R, 10	FIND 10 NULLS	C069	1100 *		
C364 04 04	0107	LD	R, 10	FIND 10 NULLS	C069	1101 *		
C364 04 04	0108	LD	R, 10	FIND 10 NULLS	C069	1102 *		
C364 04 04	0109	LD	R, 10	FIND 10 NULLS	C069	1103 *		
C364 04 04	0110	LD	R, 10	FIND 10 NULLS	C069	1104 *		
C364 04 04	0111	LD	R, 10	FIND 10 NULLS	C069	1105 *		
C364 04 04	0112	LD	R, 10	FIND 10 NULLS	C069	1106 *		
C364 04 04	0113	LD	R, 10	FIND 10 NULLS	C069	1107 *		
C364 04 04	0114	LD	R, 10	FIND 10 NULLS	C069	1108 *		
C364 04 04	0115	LD	R, 10	FIND 10 NULLS	C069	1109 *		
C364 04 04	0116	LD	R, 10	FIND 10 NULLS	C069	1110 *		
C364 04 04	0117	LD	R, 10	FIND 10 NULLS	C069	1111 *		
C364 04 04	0118	LD	R, 10	FIND 10 NULLS	C069	1112 *		
C364 04 04	0119	LD	R, 10	FIND 10 NULLS	C069	1113 *		
C364 04 04	0120	LD	R, 10	FIND 10 NULLS	C069	1114 *		
C364 04 04	0121	LD	R, 10	FIND 10 NULLS	C069	1115 *		
C364 04 04	0122	LD	R, 10	FIND 10 NULLS	C069	1116 *		
C364 04 04	0123	LD	R, 10	FIND 10 NULLS	C069	1117 *		
C364 04 04	0124	LD	R, 10	FIND 10 NULLS	C069	1118 *		
C364 04 04	0125	LD	R, 10	FIND 10 NULLS	C069	1119 *		
C364 04 04	0126	LD	R, 10	FIND 10 NULLS	C069	1120 *		
C364 04 04	0127	LD	R, 10	FIND 10 NULLS	C069	1121 *		
C364 04 04	0128	LD	R, 10	FIND 10 NULLS	C069	1122 *		
C364 04 04	0129	LD	R, 10	FIND 10 NULLS	C069	1123 *		
C364 04 04	0130	LD	R, 10	FIND 10 NULLS	C069	1124 *		
C364 04 04	0131	LD	R, 10	FIND 10 NULLS	C069	1125 *		
C364 04 04	0132	LD	R, 10	FIND 10 NULLS	C069	1126 *		
C364 04 04	0133	LD	R, 10	FIND 10 NULLS	C069	1127 *		
C364 04 04	0134	LD	R, 10	FIND 10 NULLS	C069	1128 *		
C364 04 04	0135	LD	R, 10	FIND 10 NULLS	C069	1129 *		
C364 04 04	0136	LD	R, 10	FIND 10 NULLS	C069	1130 *		
C364 04 04	0137	LD	R, 10	FIND 10 NULLS	C069	1131 *		
C364 04 04	0138	LD	R, 10	FIND 10 NULLS	C069	1132 *		
C364 04 04	0139	LD	R, 10	FIND 10 NULLS	C069	1133 *		
C364 04 04	0140	LD	R, 10	FIND 10 NULLS	C069	1134 *		
C364 04 04	0141	LD	R, 10	FIND 10 NULLS	C069	1135 *		
C364 04 04	0142	LD	R, 10	FIND 10 NULLS	C069	1136 *		
C364 04 04	0143	LD	R, 10	FIND 10 NULLS	C069	1137 *		
C364 04 04	0144	LD	R, 10	FIND 10 NULLS	C069	1138 *		
C364 04 04	0145	LD	R, 10	FIND 10 NULLS	C069	1139 *		
C364 04 04	0146	LD	R, 10	FIND 10 NULLS	C069	1140 *		
C364 04 04	0147	LD	R, 10	FIND 10 NULLS	C069	1141 *		
C364 04 04	0148	LD	R, 10	FIND 10 NULLS	C069	1142 *		
C364 04 04	0149	LD	R, 10	FIND 10 NULLS	C069	1143 *		
C364 04 04	0150	LD	R, 10	FIND 10 NULLS	C069	1144 *		
C364 04 04	0151	LD	R, 10	FIND 10 NULLS	C069	1145 *		
C364 04 04	0152	LD	R, 10	FIND 10 NULLS	C069	1146 *		
C364 04 04	0153	LD	R, 10	FIND 10 NULLS	C069	1147 *		
C364 04 04	0154	LD	R, 10	FIND 10 NULLS	C069	1148 *		
C364 04 04	0155	LD	R, 10	FIND 10 NULLS	C069	1149 *		
C364 04 04	0156	LD	R, 10	FIND 10 NULLS	C069	1150 *		
C364 04 04	0157	LD	R, 10	FIND 10 NULLS	C069	1151 *		
C364 04 04	0158	LD	R, 10	FIND 10 NULLS	C069	1152 *		
C364 04 04	0159	LD	R, 10	FIND 10 NULLS	C069	1153 *		
C364 04 04	0160	LD	R, 10	FIND 10 NULLS	C069	1154 *		
C364 04 04	0161	LD	R, 10	FIND 10 NULLS	C069	1155 *		
C364 04 04	0162	LD	R, 10	FIND 10 NULLS	C069	1156 *		
C364 04 04	0163	LD	R, 10	FIND 10 NULLS	C069	1157 *		
C364 04 04	0164	LD	R, 10	FIND 10 NULLS	C069	1158 *		
C364 04 04	0165	LD	R, 10	FIND 10 NULLS	C069	1159 *		
C364 04 04	0166	LD	R, 10	FIND 10 NULLS	C069	1160 *		
C364 04 04	0167	LD	R, 10	FIND 10 NULLS	C069	1161 *		
C364 04 04	0168	LD	R, 10	FIND 10 NULLS	C069	1162 *		
C364 04 04	0169	LD	R, 10	FIND 10 NULLS	C069	1163 *		
C364 04 04	0170	LD	R, 10	FIND 10 NULLS	C069	1164 *		
C364 04 04	0171	LD	R, 10					

ASCII CHARACTER	OUTPUT 87654321	ASCII CHARACTER	OUTPUT 87654321	ASCII CHARACTER	OUTPUT 87654321
NULL	00000000	7 <i>w</i>	10110111	o	01101111
SOH	00000001	8 <i>k</i>	10111000	p	11110000
STX	00000010 <i>↑B none</i>	9 <i>z</i>	00111001	q	01110001
ETX	00000011	: <i>z</i>	00111010	r	01110010
EOT	00001000 <i>↑D CLEAR SCREEN</i>	; {	10111011	s	11110011
ENQ	00000101	LESS THAN ;	00111100	t	01110100
ACK	00000110 <i>↑F MODE</i>	EQUALS }	10111101	u	11110101
BELL <i>Sol on board</i>	00001111	MORE THAN <i>delete</i>	10111110	v	11110110
BS <i>Backspace</i>	00010000	?	00111111	w	01110111
HT <i>TAB</i>	00010001	@	11000000	x	01111000
LF <i>Like Feed</i>	00010100 <i>φA_H 10_D</i>	A	01000001	y	11111001
VT	00010101	B	01000010	z	11111010
FF	00010100	C	11000011	{	01111011
CR <i>Return</i>	00011001 <i>φD_H 13_D</i>	D	01000100		11111100
SO	00011100	E	11000101	}	01111101
SI	00011111	F	11000110	~	01111110
DLE	00100000	G	01000111	DEL <i>Delete</i>	11111111
DC1	00010001	H	01001000		
DC2	00010010	I	11001001		
DC3	00010011	J	11001010		
DC4	00010100	K	01001011		
NAK	00010101	L	11001100		
SYN	00010110	M	01001101		
ETB	00010111	N	01001110		
CAN	00011000	O	11001111		
EM	00011001	P	01010000		
SUB	00011010	Q	11010001		
ESC	00011011	R	11010010		
FS	00011100	S	01010011		
GS	00011101	T	11010100		
RS	00011110	U	01010101		
US	00011111	V	01010110		
		W	11010111		
SPACE	01000000	X	11011000		
! <i>a</i>	00100001 <i>a</i>	Y	01011001		
" <i>b</i>	00100010 <i>b</i>	Z	01011010		
# <i>c</i>	00100011 <i>c</i>	[11011011		
\$ <i>d</i>	00100100	\	01011100		
% <i>e</i>	00100101]	11011101		
& <i>f</i>	00100110	^	11011110		
' <i>g</i>	00100111	_	01011111		
(<i>h</i>	00101000	~	01100000 <i>a</i>		
) <i>i</i>	00101001	a	11100001		
* <i>j</i>	00101010	b	11100010		
PLUS <i>k</i>	00101011	c	01100011		
, <i>l</i>	00101100	d	11100100		
MINUS <i>m</i>	00101101	e	01100101		
. <i>n</i>	00101110	f	01100110		
/ <i>o</i>	00101111	g	11100111		
0 <i>p</i>	00110000	h	11101000		
1 <i>q</i>	00110001	i	01101001		
2 <i>r</i>	00110010	j	01101010		
3 <i>s</i>	00110011	k	11101011		
4 <i>t</i>	00110100	l	01101100		
5 <i>u</i>	00110101	m	11101101		
6 <i>v</i>	00110110	n	11101110		

ASCII CHARACTER	OUTPUT 87654321	ASCII CHARACTER	OUTPUT 87654321	ASCII CHARACTER	OUTPUT 87654321
NULL	00000000	7	10110111	o	01101111
SOH	10000001	8	10111000	p	11110000
STX	10000010 <i>FB Home</i>	9	00111001	q	01110001
ETX	00000011	:	00111010	r	01110010
EOT	10000100 <i>FD CLEAR SCREEN</i>	;	10111011	s	11110011
ENQ	00000101	LESS THAN	00111100	t	01110100
ACK	00000110 <i>AF MODE</i>	EQUALS	10111101	u	11110101
BELL <i>Sol on board Comp</i>	10000111	MORE THAN	10111110	v	11110110
BS <i>Backspace</i>	10001000	?	00111111	w	01110111
HT <i>TAB</i>	00001001	@	11000000	x	01111000
LF <i>LineFeed</i>	00001010 <i>DA</i>	A	01000001	y	11111001
VT	10001011	B	01000010	z	11111010
FF	00001100	C	11000011	{	01111011
CR <i>Return Carriage</i>	10001101 <i>DA</i>	D	01000100		11111100
SO	10001110	E	11000101	}	01111101
SI	00001111	F	11000110	~	01111110
DLE	10010000	G	01000111	DEL <i>Delete</i>	11111111
DC1	00010001	H	01001000		
DC2	00010010	I	11001001		
DC3	10010011	J	11001010		
DC4	00010100	K	01001011		
NAK	10010101	L	11001100		
SYN	10010110	M	01001101		
ETB	00010111	N	01001110		
CAN	00011000	O	11001111		
EM	10011001	P	01010000		
SUB	10011010	Q	11010001		
ESC	00011011	R	11010010		
FS	10011100	S	01010011		
GS	00011101	T	11010100		
RS	00011110	U	01010101		
US	10011111	V	01010110		
SPACE	10100000	W	11010111		
!	00100001	X	11011000		
"	00100010	Y	01011001		
#	10100011	Z	01011010		
\$	00100100	[11011011		
%	10100101	\	01011100		
&	10100110]	11011101		
'	00100111	^	11011110		
(00101000	_	01011111		
)	10101001	~	01100000		
*	10101010	a	11100001		
PLUS	00101011	b	11100010		
,	10101100	c	01100011		
MINUS	00101101	d	11100100		
.	00101110	e	01100101		
/	10101111	f	01100110		
0	00110000	g	11100111		
1	10110001	h	11101000		
2	10110010	i	01101001		
3	00110011	j	01101010		
4	10110100	k	11101011		
5	00110101	l	01101100		
6	00110110	m	11101101		
		n	11101110		

lower case = '001000001'B OR appearance


```

C000                org 0c000h
C000 00             DB 0
C001 C365C2       BGIN: JMP STRTA
*
* JUMP TABLE I/O ROUTINES
* 0 - DISPLAY SCREEN
* 1 - SERIAL OUTPUT
* 2 - PARALLEL OUTPUT
* 3 - ERROT HANDLER
* ENTRY AT: SOUT SELECTS CURRENT OUTPUT DEVICE
* AOUT SELECTS DEVICE IN REGISTER "A"
C004 3A03C8       SOUT: LDA OPORT
C007 E603         AOUT: ANI 3
C009 E5           PUSH H
C00A 2184C1       LXI H,OTAB
C00D 07           RLC
C00E 85           ADD L
C00F 6F           MOV L,A
C010 C371C0       JMP DISPT
*
* ENTRY POINTS ARE DEFINED:
* 0 - KEYBOARD IN
* 1 - SERIAL IN
* 2 - PARALLEL IN
* 3 - ERROR HANDLER
C013 3A04C8       SINP: LDA IPORT
C016 E603         AINP: ANI 3
C018 E5           PUSH H
C019 218CC1       LXI H,ITAB
C01C 07           RLC
C01D 85           ADD L
C01E 6F           MOV L,A
C01F C371C0       JMP DISPT
*
* KEYBOARD INPUT STATUS CHECK
C022 DBFA        KSTAT: IN STAPT
C024 E601         ANI KDR
C026 C9           RET
*
* KEYBOARD DATA INPUT
C027 E1           KREA1: POP H
C028 CD22C0       KREAD: CALL KSTAT
C02B C228C0       JNZ KREAD
C02E DBFC         IN KDATA
C030 C9           RET
*
* SERIAL INPUT STATUS CHECK
C031 DBF8        SSTAT: IN SERST
C033 E640         ANI SDR
C035 C9           RET
*

```

```

* SERIAL DATA INPUT
C036 E1      SREA1: POP  H
C037 CD31C0 SREAD: CALL SSTAT
C03A CA37C0      JZ   SREAD
C03D DBF9        IN   SDATA
C03F C9         RET

*
* SERIAL DATA OUTPUT
C040 E1      SEROT: POP  H
C041 DBF8    SDROT: IN   SERST
C043 17      RAL
C044 D241C0  JNC   SDROT
C047 78      MOV  A,B
C048 D3F9    OUT  SDATA
C04A C9      RET

*
* VIDEO DISPLAY DRIVER
C04B E5      VDMOT: PUSH H
C04C D5      VDMO1: PUSH D
C04D C5      PUSH B
C04E F5      PUSH PSW
C04F 78      MOV  A,B
C050 2165C1  LXI  H,TBL
C053 CD62C0  CALL TSRCH

*
C056 CD21C1  GOBACK: CALL VDADD
C059 7E      MOV  A,M
C05A F680    ORI  80H
C05C 77      MOV  M,A
C05D F1      GOBK:  POP  PSW
C05E C1      POP  B
C05F D1      POP  D
C060 E1      POP  H
C061 C9      RET

*
C062 7E      TSRCH: MOV  A,M
C063 B7      ORA  A
C064 CA7BC0  JZ   CHAR
C067 B8      CMP  B
C068 23      INX  H
C069 C276C0  JNZ  NEXT
C06C E5      PUSH H
C06D CD3DC1  CALL CREM
C070 E1      POP  H

*
* DISPATCH
C071 7E      DISPT: MOV  A,M
C072 23      INX  H
C073 66      MOV  H,M
C074 6F      MOV  L,A
C075 E9      PCHL

*
C076 23      NEXT:  INX  H

```

```

C077 23          INX  H
C078 C362C0     JMP  TSRCH
*
C07B 78          CHAR:  MOV  A,B
C07C B7          ORA  A
C07D C8          RZ
C07E FE7F       CPI  7FH
C080 C8          RZ
*
C081 CD21C1     OCHAR: CALL VDADD
C084 78          MOV  A,B
C085 E67F       ANI  7FH
C087 77          MOV  M,A
C088 3A00C8     LDA  NCHAR
C08B FE3F       CPI  63
C08D DAADCO     JC   OK
C090 3A01C8     LDA  LINE
C093 FEOF       CPI  15
C095 C2ADCO     JNZ  OK
*
* END OF SCREEN. ROLL UP ONE LINE
C098 AF          SCROLL: XRA  A
C099 3200C8     STA  NCHAR
C09C 4F          SROL:  MOV  C,A
C09D CD28C1     CALL VDAD
COA0 AF          XRA  A
COA1 CDE9C0     CALL CLIN1
COA4 3A02C8     LDA  BOT
COA7 3C          INR  A
COA8 E60F       ANI  OFH
COAA C3DDCO     JMP  ERAS3
*
* INCREMENT LINE COUNTER, IF NECESSARY
COAD 3A00C8     OK:   LDA  NCHAR
COB0 3C          INR  A
COB1 3200C8     STA  NCHAR
COB4 FE40       CPI  64
COB6 D8          RC
COB7 AF          XRA  A
COB8 3200C8     STA  NCHAR
COBB 3A01C8     LDA  LINE
COBE 3C          INR  A
COBF E60F       ANI  OFH
COC1 3201C8     CUR:  STA  LINE
COC4 C9          RET
*
* ERASE SCREEN
COC5 2100CC     PERSE: LXI  H,VDMEM
COC8 36A0       MVI  M,80H+' ' ;CURSOR
*
COCA 23          ERAS1: INX  H
COCB 7C          MOV  A,H
COCC FEDO       CPI  ODOH

```

```

COCE D2D6C0          jnc  eras2
COD1 3620            MVI  M, ' '
COD3 C3CAC0          JMP  ERAS1
*
COD6 AF              ERAS2: XRA  A
COD7 3201C8          STA  LINE
CODA 3200C8          STA  NCHAR
*
CDDD D3FE            ERAS3: OUT  DSTAT
CODF 3202C8          STA  BOT
COE2 C9              RET
*
COE3 CD21C1          CLINE: CALL VDADD
COE6 3A00C8          LDA  NCHAR
COE9 FE40            CLIN1: CPI  64
COEB D0              RNC
COEC 3620            MVI  M, ' '
COEE 23              INX  H
COEF 3C              INR  A
COF0 C3E9C0          JMP  CLIN1
*
*   HOME  CURSOR
COF3 AF              PHOME: XRA  A
COF4 3200C8          STA  NCHAR
COF7 C3C1C0          JMP  CUR
*
*   MOVE  CURSOR DOWN ONE LINE
COFA 3A01C8          PDOWN: LDA  LINE
COFD FEOF            CPI  15
COFF C8              RZ
C100 3C              INR  A
C101 C3C1C0          JMP  CUR
*
*   MOVE  CURSOR UP ONE LINE
C104 3A01C8          PUP:  LDA  LINE
C107 B7              ORA  A
C108 C8              RZ
C109 3D              DCR  A
C10A C3C1C0          JMP  CUR
*
*   MOVE  CURSOR LEFT ONE COLUMN
C10D 3A00C8          PLEFT: LDA  NCHAR
C110 B7              ORA  A
C111 C8              RZ
C112 3D              DCR  A
C113 3200C8          PCUR: STA  NCHAR
C116 C9              RET
*
*   MOVE  CURSOR RIGHT ONE COLUMN
C117 3A00C8          PRIT: LDA  NCHAR
C11A FE3F            CPI  63
C11C C8              RZ
C11D 3C              INR  A

```

```

C11E C313C1      JMP  PCUR
*
* CALCULATE SCREEN ADDRESSES
* VDADD=CURRENT SCREEN ADDRESS
* VDAD2=ADDRESS OF LINE
* VDAD =LINE IN 'A', CHARACTER IN 'C'
C121 3A00C8      VDADD: LDA  NCHAR
C124 4F          MOV  C,A
C125 3A01C8      VDAD2: LDA  LINE
C128 6F          VDAD:  MOV  L,A
C129 3A02C8      LDA  BOT
C12C 85          ADD  L
C12D 0F          RRC
C12E 0F          RRC
C12F 6F          MOV  L,A
C130 E603        ANI  3
C132 57          MOV  D,A
C133 3ECC        MVI  A,VDMEM SHR 8
*
* Mvi A,<VDMEM
C135 82          ADD  D
C136 67          MOV  H,A
C137 7D          MOV  A,L
C138 E6C0        ANI  OCOH
C13A 81          ADD  C
C13B 6F          MOV  L,A
C13C C9          RET
*
* REMOVE CURSOR
C13D CD21C1      CREM: CALL VDADD
C140 7E          MOV  A,M
C141 E67F        ANI  7FH
C143 77          MOV  M,A
C144 C9          RET
*
* BACKSPACE
C145 CD0DC1      PBACK: CALL PLEFT
C148 CD21C1      CALL VDADD
C14B 3620        MVI  M,' '
C14D C9          RET
*
* CARRIAGE RETURN
C14E CDE3C0      PCR:   CALL CLINE
C151 AF          XRA  A
C152 C313C1      JMP  PCUR
*
* LINEFEED
C155 3A01C8      PLF:   LDA  LINE
C158 FEOF        CPI  15
C15A D261C1      JNC  SC
C15D 3C          INR  A
C15E C3C1C0      JMP  CUR
*
C161 AF          SC:   XRA  A

```

```

C162 C39CC0          JMP  SROL
*
* TABLE OF SPECIAL CHARACTERS
* AND WHERE TO PROCESS
* IF CHAR IS NOT IN TABLE,
* THEN IT GOES TO THE SCREEN
C165 04          TBL:  DB  CLEAR
C166 C5C0        DW  PERSE
C168 97          DB  UP
C169 04C1        DW  PUP
C16B 9A          DB  DOWN
C16C FAC0        DW  PDOWN
C16E 81          DB  LEFT
C16F 0DC1        DW  PLEFT
C171 93          DB  RIGHT
C172 17C1        DW  PRIT
C174 02          DB  HOME
C175 F3C0        DW  PHOME
C177 0D          DB  CR
C178 4EC1        DW  PCR
C17A 0A          DB  LF
C17B 55C1        DW  PLF
C17D 08          DB  BACKS
C17E 45C1        DW  PBACK
C180 06          DB  MODE
C181 B4C1        DW  COMND
C183 00          DB  0
*
* TABLE OF OUTPUT DEVICES
C184 4CC0        OTAB: DW  VDMO1
C186 40C0        DW  SEROT
C188 ADC1        DW  ERROT
C18A ADC1        DW  ERROT
*
* TABLE OF INPUT DEVICES
C18C 27C0        ITAB: DW  KREA1
C18E 36C0        DW  SREA1
C190 ADC1        DW  ERROT
C192 ADC1        DW  ERROT
*
* TABLE OF COMMANDS
C194 5445        COMTAB: DB  'TE'
C196 70C2        DW  TERM
C198 4455        DB  'DU'
C19A A1C2        DW  DUMP
C19C 454E        DB  'EN'
C19E 07C3        DW  ENTER
C1A0 4558        DB  'EX'
C1A2 35C3        DW  EXEC
C1A4 544C        DB  'TL'
C1A6 39C3        DW  TLOAD
C1A8 4241        DB  'BA'
C1AA 0000        DW  0

```

```

C1AC 00          DB      0
*
*  CONSOL PORT ERROR HANDLER
C1AD AF          ERROT:  XRA  A
C1AE 3204C8      STA  IPORT
C1B1 3203C8      STA  OPORT
*
*  GET AND PROCESS COMMANDS
C1B4 3100CC      COMND:  LXI  SP,SYSTP
C1B7 CD12C2      CALL  PROMPT
C1BA CDC3C1      CALL  GCLIN
C1BD CDE1C1      CALL  COPRC
C1C0 C3B4C1      JMP   COMND
*
*  READ COMMAND FROM KEYBOARD
*  AND SEND IT TO THE SCREEN
*  CR ENDS COMMAND AND ERASES ALL CHARS TO RIGHT
*  OF THE CURSOR
*  LF ENDS COMMAND
*  MODE RESTARTS THE COMMAND LINE
C1C3 CD28C0      GCLIN:  CALL  KREAD
C1C6 FE20        CPI   20H
C1C8 47          MOV   B,A
C1C9 DAD2C1      JC    PROCS
C1CC CD4BC0      CONT:  CALL  VDMOT
C1CF C3C3C1      JMP   GCLIN
*
*  PROCESS CONTROL KEYS
C1D2 FE0D        PROCS:  CPI   CR
C1D4 CADD C1     JZ    CRPRC
C1D7 FE0A        CPI   LF
C1D9 C8          RZ
C1DA C3C3C1     JMP   GCLIN
*
C1DD CDE3C0     CRPRC:  CALL  CLINE
C1E0 C9         RET
*
*  FIND AND PROCESS COMMAND
C1E1 CD3DC1     COPRC:  CALL  CREM
C1E4 OE01       MVI  C,1
C1E6 CD25C1     CALL  VDAD2
C1E9 EB        XCHG
C1EA CD32C2     CALL  SCHR
C1ED CA8EC3     JZ    ERR1
C1F0 EB        XCHG
C1F1 1194C1     LXI  D,COMTAB
*
*  SEARCH TABLE POINTED TO BY "DE"
*  FOR TWO CHAR MATCH OF
*  "HL" MEMORY.  IF NO MATCH
*  THEN INSERT "?"
C1F4 1A        FDCOM:  LDAX D
C1F5 B7        ORA  A

```

```

C1F6 CA8FC3      JZ   ERR2
C1F9 E5          PUSH H
C1FA BE          CMP  M
C1FB 13          INX  D
C1FC C20BC2     JNZ  NCOM
*
C1FF 23          INX  H
C200 1A          LDAX D
C201 BE          CMP  M
C202 C20BC2     JNZ  NCOM
*
C205 C1          POP  B
C206 EB          XCHG
C207 23          INX  H
C208 C371C0     JMP  DISPT
*
C20B 13          NCOM: INX  D
C20C 13          INX  D
C20D 13          INX  D
C20E E1          POP  H
C20F C3F4C1     JMP  FDCOM
*
*   PUT CR, LF, PROMPT;
C212 CD1AC2     PROMPT: CALL CRLF
C215 063E       MVI  B,'>'
C217 C34BC0     JMP  VDMOT
*
C21A 060A       CRLF:  MVI  B,LF
C21C CD4BC0     CALL  VDMOT
C21F 060D       MVI  B,CR
C221 C34BC0     JMP  VDMOT
*
*   scan over up to 12 characters looking for a blank
C224 0E0C       SBLK:  MVI  C,12
C226 1A         SBLK1: LDAX  D
C227 FE20       CPI   BLANK
C229 CA32C2     JZ    SCHR
C22C 13         INX  D
C22D 0D         DCR  C
C22E C226C2     JNZ  SBLK1
C231 C9         RET
*
*   SCAN FOR NON BLANK, OVER UP TO 10 BLANKS
C232 0E0A       SCHR:  MVI  C,10
C234 1A         SCHR1: LDAX  D
C235 FE20       CPI   SPACE
C237 C0         RNZ
C238 13         INX  D
C239 0D         DCR  C
C23A C8         rz
C23B C334C2     JMP  SCHR1
*
*   SCAN PAST CHARS AND BLANKS FOR ADDRESS

```



```

* THEN COVERTS IT TO HEX
C23E CD24C2 SCONV: CALL SBLK
C241 CA8EC3      JZ  ERR1
*
* CONVERT ASCII DIGITS TO BINARY FOLLOWING
* STANDARD HEX CONVERSION. IT STOPS WHEN A SPACE
* IS ENCOUNTERED.
* PARAMETER ERRORS REPLACE THE ERROR CHAR ON THE SCREEN
* WITH A ?.
C244 210000 SHEX: LXI H,0
C247 1A SHE1: LDAX D
C248 FE20      CPI  ' '
C24A C8        RZ
*
C24B 29 HCONV: DAD H
C24C 29      DAD H
C24D 29      DAD H
C24E 29      DAD H
C24F CD5BC2  CALL HCOV1
C252 D28EC3  JNC ERR1
C255 85      ADD L
C256 6F      MOV L,A
C257 13      INX D
C258 C347C2  JMP SHE1
*
C25B D630 HCOV1: SUI '0'
C25D FE0A   CPI 10
C25F D8     RC
C260 D607   SUI 7
C262 FE10   CPI 10H
C264 C9     RET
*
* SYSTEM START
* CLEAR PART OF RAM, SET STACK POINTER,
* AND FALL THROUGH TO TERMINAL MODE
C265 AF STRTA: XRA A
C266 4F      MOV C,A
C267 2100C8  LXI H,SYSRAM
*
C26A 77 CLERA: MOV M,A
C26B 23      INX H
C26C 0C      INR C
C26D C26AC2  JNZ CLERA
*
* TERMINAL MODE
* GET CHAR FROM KEYBOARD, AND PUT TO MONITOR
* COMMAND KEYS ARE NOT PUT TO TERMINAL,
* BUT ARE INTERPRETED AS DIRECT SOL COMMANDS
* THE MODE COMMAND PUTS SOL IN THE COMMAND MODE
C270 3100CC TERM: LXI SP,SYSTP
C273 CDEDC3      CALL TOFF
C276 CDC5C0      CALL PERSE
*

```

```

C279 CD22C0      KIN:      CALL  KSTAT
C27C C290C2      JNZ   TIN
C27F DBFC        IN    KDATA
C281 47          MOV   B,A
                *        ANI   80H
                *        JZ    TOUT
C282 FE06        cpi   06      ;Command Key?
C284 C28DC2      jnz   TOUT
C287 CD4BC0      CALL  VDMOT ;Process it
C28A C390C2      JMP   TIN
                *
C28D CD41C0      TOUT:   CALL  SDROT
C290 CD31C0      TIN:   CALL  SSTAT
C293 CA79C2      JZ    KIN
C296 DBF9        IN    SDATA
C298 E67F        ANI   7FH
C29A 47          MOV   B,A
C29B CD4BC0      CALL  VDMOT
C29E C379C2      JMP   KIN
                *
                *
                *   DUMP COMMAND
                *   DUMP ALL CHARS FROM MEMORY TO THE
                *   CURRENT OUTPUT DEVICE
                *   ALL VALUES ARE DISPLAYED AS HEX
                *
                *   THE COMMAND IS AS FOLLOWS:
                *   DUMP ADDR1 ADDR2
                *   THE VALUES FROM ADDR1 TO ADDR2 ARE THEN OUTPUT TO THE
                *   OUTPUT DEVICE. IF ONLY ADDR1 IS GIVEN, THEN THE VALUE AT TH
C2A1 CD3EC2      DUMP:   CALL  SCONV
C2A4 E5          PUSH  H
C2A5 CD32C2      CALL  SCHR
C2A8 E1          POP   H
C2A9 CAB4C2      JZ    POVER
C2AC E5          PUSH  H
C2AD CD44C2      CALL  SHEX
C2B0 D1          POP   D
C2B1 C3B6C2      JMP   NPASS
                *
C2B4 54          POVER:  MOV   D,H
C2B5 5D          MOV   E,L
C2B6 EB          NPASS:  XCHG
                *
C2B7 CD1AC2      DLOOP:  CALL  CRLF
C2BA DBFC        IN    KDATA
C2BC FE06        CPI   MODE
C2BE CAB4C1      JZ    COMND
C2C1 CDDDC2      CALL  ADOUT
C2C4 CDE5C2      CALL  BOUT
C2C7 0E10        MVI   C,16
                *
C2C9 7E          DLP1:  MOV   A,M

```

```

C2CA C5          PUSH B
C2CB CDE2C2     CALL HBOUT
C2CE CD02C3     CALL ACOMP
C2D1 D2B4C1     JNC COMND
C2D4 C1         POP B
C2D5 23         INX H
C2D6 0D         DCR C
C2D7 C2C9C2     JNZ DLP1
C2DA C3B7C2     JMP DLOOP

*
* PUT HL AS 16 BIT HEX
C2DD 7C        ADOUT: MOV A,H
C2DE CDEAC2    CALL HeOUT
C2E1 7D        MOV A,L

*
C2E2 CDEAC2    HBOUT: CALL HEOUT
C2E5 0620     BOUT: MVI B,' '
C2E7 C34BC0   JMP VDMOT

*
C2EA 4F        HEOUT: MOV C,A
C2EB OFOF0F0F RRC!RRC!RRC!RRC
C2EF CDF3C2    CALL HEOU1
C2F2 79        MOV A,C

*
C2F3 E60F     HEOU1: ANI OFH
C2F5 C630     ADI '0'
C2F7 FE3A     CPI '9'+1
C2F9 DAFEC2   JC OUTH
C2FC C607     ADI 7
C2FE 47       OUTH: MOV B,A
C2FF C34BC0   JMP VDMOT

*
* COMPARE DE AND HL
C302 7D       ACOMP: MOV A,L
C303 93       SUB E
C304 7C       MOV A,H
C305 9A       SBB D
C306 C9       RET

*
* ENTER COMMAND
* GET VALUES FROM KEYBOARD AND PUT INTO MEMORY
* THE INPUT VALUES ARE SCANNED FOLLOWING,
* A STANDARD 'GOLIN' INPUT SO ON SCREEN EDITING MAY TAKE
* PLACE PRIOR TO THE LINE TERMINATOR. A SLASH, '/',
* ENDS THE ENTER COMMAND AND RETURNS CONTROL TO
* THE COMMAND MODE
C307 CD3EC2   ENTER: CALL SCONV
C30A E5       PUSH H

*
C30B CD1AC2   ENLOP: CALL CRLF
C30E 063A     MVI B,':'
C310 CDCCC1   CALL CONT
C313 CD3DC1   CALL CREM

```

```

C316 OE01          MVI  C,1
C318 CD25C1       CALL VDAD2
C31B EB          XCHG
*
C31C OE03       ENL01: MVI  C,3
C31E CD34C2     CALL  SCHR1
C321 CA0BC3     JZ    ENLOP
*
C324 1A         ENL02: LDAX D
C325 FE2F       CPI  '/'
C327 CAB4C1     JZ    COMND
C32A CD44C2     CALL SHEX
C32D 7D         MOV  A,L
C32E E1         POP  H
C32F 77         MOV  M,A
C330 23         INX  H
C331 E5         PUSH H
C332 C31CC3     JMP  ENL01
*
* EXECUTE COMMAND
* GET THE FOLLOWING PARAMETER AND JUMP TO THE LOCATION.
* IF PROPER STACK OPERATIONS ARE USED WITHIN THE EXTERNAL
* PROGRAM,
* IT CAN DO A STANDARD 'RET' TO THE CONSOL COMMAND MODE
C335 CD3EC2     EXEC:  CALL SCONV
C338 E9         PCHL
*
* TAPE LOAD COMMAND
* READ FROM EITHER TAPE UNIT, PLACING THE DATA INTO MEMORY.
* WHILE SPACE WITHIN CONSOL DOES NOT ALLOW FOR 'STANDARD'
* TAPE ROUTINES, THIS COMMAND WILL LOAD SOL-BAS
C339 CD24C2     TLOAD: CALL SBLK
C33C CA4AC3     JZ    DFLT
C33F CD44C2     CALL SHEX
C342 7D         MOV  A,L
C343 E601       ANI  1
C345 3E20       MVI  A,' '
C347 C24BC3     JNZ  SETSP
*
C34A AF         DFLT:  XRA  A
C34B F6C0       SETSP: ORI  TAPE1+TAPE2
C34D D3FA       OUT  STAPT
C34F CDF1C3     CALL DELAY
C352 DBFB       IN   TDATA
*
C354 CD94C3     TLO01: CALL RHEAD
C357 C254C3     JNZ  TLO01
*
C35A 2A0CC8     LHLD BLOCK
C35D EB         XCHG
C35E 2A0EC8     LHLD LOADR
*
C361 7A         L0LOOP: MOV  A,D

```

```

C362 B3          ORA  E
C363 CAEDC3     JZ   TOFF
C366 0100FF    LXI  B,-256
C369 EB        XCHG
C36A 09        DAD  B
C36B D2E1C3    JNC  LBLK
C36E 0600      MVI  B,0
*
C370 0E00      RDBLK: MVI  C,0
C372 EB        XCHG
*
C373 CDC6C3    RTBYT: CALL TAPIN
C376 77        MOV  M,A
C377 23        INX  H
C378 A9        XRA  C
C379 2F        CMA
C37A 91        SUB  C
C37B 4F        MOV  C,A
C37C 05        DCR  B
C37D C273C3    JNZ  RTBYT
*
C380 CDC1C3    CALL CRCK
C383 CA61C3    JZ   LOLOOP
*
C386 0607      TERR:  MVI  B,'G'-40H
C388 CD4BC0    CALL VDMOT
C38B C3B4C1    JMP  COMND
*
*  CONSOL ERROR HANDLER
C38E EB        ERR1:  XCHG
C38F 363F      ERR2:  MVI  M,'?'
C391 C3B4C1    JMP  COMND
*
*  READ THE HEADER
C394 060A      RHEAD: MVI  B,10
C396 DBFA      RHEA1: IN   STAPT
C398 E640      ANI  TDR
C39A CA96C3    JZ   RHEA1
C39D DBFB      IN   TDATA
C39F B7        ORA  A
C3A0 C294C3    JNZ  RHEAD
C3A3 05        DCR  B
C3A4 C296C3    JNZ  RHEA1
*
*  WAIT FOR START CHAR
C3A7 CDC6C3    SOHL:  CALL TAPIN
C3AA 3D        DCR  A
C3AB C2A7C3    JNZ  SOHL
*
*  NOW GET THE HEADER
C3AE 2105C8    LXI  H,THEAD
C3B1 010010    LXI  B,HLEN*256
*

```

```

C3B4 CDC6C3    RHED1:  CALL TAPIN
C3B7 77        MOV  M,A
C3B8 23        INX  H
C3B9 A9        XRA  C
C3BA 2F        CMA
C3BB 91        SUB  C
C3BC 4F        MOV  C,A
C3BD 05        DCR  B
C3BE C2B4C3    JNZ  RHED1
*
*  GET THE NEXY BYTE AND COMPARE IT
*  TO THE VALUE IN REGISTER C.
*  THE FLAGS ARE SET ON RETURN
C3C1 CDC6C3    CRCCK:  CALL TAPIN
C3C4 B9        CMP  C
C3C5 C9        RET
*
*  GET THE NEXT AVAILABLE BYTE FROM THE TAPE.
*  WHILE WAITING, TEST THE KEYBOARD FOR A "MODE" COMMAND
*  IF RECEIVED, STOP THE TAPE LOAD AND RETURN TO THE
*  COMMAND MODE
C3C6 DBFA      TAPIN:  IN   STAPT
C3C8 E640      ANI  TDR
C3CA C2D7C3    JNZ  TREDY
C3CD DBFC      IN   KDATA
C3CF FE06      CPI  MODE
C3D1 CAB4C1    JZ   comnd
C3D4 C3C6C3    JMP  TAPIN
*
C3D7 DBFA      TREDY:  IN   STAPT
C3D9 E618      ANI  TFE+TOE
C3DB C286C3    JNZ  TERR
C3DE DBFB      IN   TDATA
C3E0 C9        RET
*
*  CALCULATE THE LENGTH OF THE LAST BLOCK
C3E1 01FFFF    LBLK:  LXI  B,-1
C3E4 09        DAD  B
C3E5 23        INX  H
C3E6 45        MOV  B,L
C3E7 210000    LXI  H,0
C3EA C370C3    JMP  RDBLK
*
*  TURN THE TAPES OFF
C3ED AF        TOFF:  XRA  A
C3EE D3FA      OUT  STAPT
C3F0 C9        RET
*
C3F1 110000    DELAY:  LXI  D,0
C3F4 1B        DLOP1: DCX  D
C3F5 7A        MOV  A,D
C3F6 B3        ORA  E
C3F7 C2F4C3    JNZ  DLOP1

```

```

C3FA C9          RET
*
* SOL SYSTEM EQUATES
* DISPLAY PARAMETERS
CC00 = VDMEM EQU 0C00H
*
* KEYBOARD SPECIAL KEYS
009A = DOWN EQU 9AH
0097 = UP EQU 97H
0081 = LEFT EQU 81H
0093 = RIGHT EQU 93H
008C = LOADK EQU 8CH
0006 = MODE EQU 06H
0004 = CLEAR EQU 04H
0002 = HOME EQU 02H
0008 = BACKS EQU 08H
000A = LF EQU 10
000D = CR EQU 13
0020 = BLANK EQU ' '
0020 = SPACE EQU BLANK
0030 = CX EQU 'X'-40
*
* PORT ASSIGNMENTS
00FA = STAPT EQU 0FAH
00F8 = SERST EQU 0F8H
00F9 = SDATA EQU 0F9H
00FB = TDATA EQU 0FBH
00FC = KDATA EQU 0FCH
00FD = PDATA EQU 0FDH
00FE = DSTAT EQU 0FEH
00FF = SENSE EQU 0FFH
*
* BIT ASSIGNMENT MASKS
0001 = SCD EQU 1
0002 = SDSK EQU 2
0004 = SPE EQU 4
0008 = SFE EQU 8
0010 = SOE EQU 16
0020 = SCTS EQU 32
0040 = SDR EQU 64
0080 = STBE EQU 128
*
0001 = KDR EQU 1
0002 = PDR EQU 2
0004 = PXDR EQU 4
0008 = TFE EQU 8
0010 = TOE EQU 16
0040 = TDR EQU 64
0080 = TTBE EQU 128
*
0001 = SOK EQU 1
*
0040 = TAPE1 EQU 64

```

```

0080 =    TAPE2    EQU 128
          *
          * SOL SYSTEM GLOBAL AREA
C800      ORG    OC800H
          *
C800 =    SYSRAM  EQU    $
CC00 =    SYSTP   EQU   $+1024
          *
          * CONSOL PARAMETER AREA
C800      NCHAR   DS 1
C801      LINE    DS 1
C802      BOT     DS 1
C803      OPORT   DS 1
C804      IPORT   DS 1
          *
C805      THEAD   DS 5
C80A      DS      DS 1
C80B      HTYPE   DS 1
C80C      BLOCK   DS 2
C80E      LOADR   DS 2
C810      XEQAD   DS 2
C812      ASPR    DS 3
          *
0010 =    HLEN    equ $-THEAD
          *
          *

```