

CONSOLE is a 1024 byte program designed to allow the Sol TERMINAL/COMPUTER to operate as a standard CRT terminal and to provide access to the essential computer capabilities of the Sol. Using CONSOLE, self test and small diagnostic programs can be entered to system memory and executed. This is in addition to providing facilities for of correct system operation helps in finding errors in case of a malfunction.

In addition, CONSOLE contains standardized entry points for all normal I/O operations. These routines are common with all Sol system software allowing each personality module in the Sol to be interface with external programs in an almost identical manner.

SECTION IX

SOFTWARE

Sol TERMINAL COMPUTERTM

**Processor
Technology**

9.1 CONSOL

CONSOL is a 1024 byte program designed to allow the Sol TERMINAL/COMPUTER to operate as a standard CRT terminal and to provide access to the essential computer capabilities of the Sol. Using CONSOL, self test and small diagnostic programs can be entered to system memory and executed. This in addition to providing verification of correct system operation helps in finding errors in case of a malfunction.

In addition, CONSOL contains standardized entry points for all normal I/O operations. These routines are common with all Sol System Software allowing each personality module in the Sol line to interface with external programs in an almost identical manner.

A cassette read routine is also resident in the CONSOL module allowing Sol Software to be loaded and run in a system with additional memory. Sol System Software as of November 1976 includes BASIC, FOCAL, a Scientific Calculator and numerous "game" packages including a 8K assembly language version of STARTREK called TREK80.

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 SELECT causes an immediate change in the operation of the unit.

When the MODE key is depressed CONSOL issues a prompt character (>) and waits for a command line to be input. 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 MODE SELECT key to start new command line

9.1.1 DUmP (addr) (addr)

The DUmP command displays memory data on the screen in a 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 a error displayed on the screen. If two addresses are input then all values from the first address to the last will be displayed.

DUMP Ø EF

Up to ten blanks may be inserted between each parameter without forcing an error condition. Errors are indicated 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".

9.1.2 ENter addr

The ENter command places sequential bytes into memory beginning at the specified address. Data, represented as hexadecimal values, are input from the keyboard for entry to memory. All CONSOL commands except MODE SELECT are executed when the RETURN key is pressed. After the ENTER, (address), RETURN sequence the Sol Displays a colon (:) prompt character. Values are then input one line at a time with each line terminated by a carriage return or linefeed. The ENTER function itself is terminated with a slash (/) and the Sol goes back to the command mode when the slash is encountered.

With all command functions of CONSOL, 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 to the left using the "cursor-left" key and the erroneous characters changed. A linefeed would then be used as a terminator since LF does not erase the line prior to processing the characters. This is particularly useful when using the ENter command since the input line can be visually scanned and errors corrected prior to the actual entry of input data to memory.

9.1.3 TLoad (speed)

Included within CONSOL are routines to read standardized cassette tape Software which is recorded 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 turns on the cassette player and waits for the next header, then uses the header information and loads the file to memory. The cassette recorder must be in play mode and properly connected before executing the TLoad command.

After loading the data, CONSOL returns to the command mode where the EXEC command can be used to execute the just loaded program. Also, a return can normally be made to the command mode by pressing the MODE SELECT key. Space limitations again limited escape during the header search, so if the system locks up in this routine the standard Sol restart must be used. To restart the Sol press UPPER CASE and REPEAT keys simultaneously.

The CUTS 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. (1200 and 300 bits per second respectively). If no parameter is given CONSOL will default to high speed operation as all standard Processor Technology Sol-System Software is recorded at this speed.

9.1.4 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 an 8080 CALL instruction so the program can return to the command mode using a standard 8080 RET instruction if normal stack operations are used.

9.1.5 BASic

The BASic command is provided for executing programs whose starting address is 0, such as Sol-BASIC5.

9.2 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 Communications Channel 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.

--- CONSOL ---
COPYRIGHT 1976

PROCESSOR TECHNOLOGY CORP.
6200 HOLLIS STREET
EMERYVILLE, CALIF. 94608

C000	0001	*
C000	0002	*
C000	0003	*
C000	0004	*
C000	0005	*
C000	0006	*
C000	0007	*
C000	0008	*
C000	0009	*
C000	0010	*
C000	0011	*
C000	0012	*
C000	0013	*
C000	0014	*
C000	0015	*
C000	0016	*
C000	0017	*
C000	0018	*
C000	0019	*
C000	0020	*
C000	0021	*
C000	0022	*
C000	0023	*
C000	0024	*
C000	0025	*
C000	0026	*
C000	0027	*
C000	0028	*
C000	0029	*
C000	0030	*
C000	0031	*
C000	0032	*
C000	0033	*
C000	0034	*
C000	0035	*
C000	0036	*
C000	0037	*
C000	0038	*
C000	0039	*
C000	0040	*
C000	0041	*
C000	0042	*
C000	0043	*
C000	0044	*
C000	0045	*
C000	0046	*
C000	0047	*
C000	0048	B
C004	0049	*
C004	0050	*
C004	0051	*

SYSTEM SOFTWARE

<<>> CONSOL <<>>

VERSION: 1.0:5
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 CONSOLE 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.

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 COPYRIGHT 1976 BY PROCESSOR TECHNOLOGY AND ALL RIGHTS THEREIN ARE RESERVED.

AUTO-STARTUP CODE

DB 0 FOUR PHASE WONDER
IMP 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 HAVE 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 - VDM 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      AOUT      LDA      OPORT      GET PORT NUMBER FROM MEMORY LOCATION
0052      0053      ANI      3          KEEP IT IN CONTROL
0054      0055      PUSH     H          WE'LL RESTORE IT LATER
0056      0057      LXI      H,OTAB    POINT TO TABLE
0058      0059      RLC          COMPUTE ADDRESS
0060      0061      MOV      L,A      WE HAVE IT
0062      0063      DISPT     GO TO HL...
0064      0065      JMP

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

SINP      AINP      LDA      IPORT    GET PORT NUMBER FROM MEMORY LOCATION
0066      0067      ANI      3        WE MUST BE REASONABLE
0068      0069      PUSH     H        SAVE H&L
0070      0071      LXI      H,ITAB   POINT TO TABLE
0072      0073      RLC          THE MATH
0074      0075      ADD      L        DONE
0076      0077      MOV      L,A
0078      0079      JMP

3A 03 C8
0080      0081      SOUT      AOUT
0082      0083      LDA      OPORT
0084      0085      ANI      3
0086      0087      PUSH     H
0088      0089      LXI      H,OTAB
0090      0091      RLC
0092      0093      MOV      L,A
0094      0095      DISPT
0096      0097      JMP

3A 04 C8
0098      0099      SINP      AINP
0100      0101      LDA      IPORT
0102      0103      ANI      3
0104      0105      PUSH     H
0106      0107      LXI      H,ITAB
0108      0109      RLC
0110      0111      ADD      L
0112      0113      MOV      L,A
0114      0115      JMP

```


--- CONSOL ---
COPYRIGHT 1976

ADDRESS	INSTR	OP	COND	DATA	COMMENT
0105	JMP				WE HAVE THE ADDRESS...GO TO HEAVEN
0106					
0107					
0108					KEYBOARD INPUT STATUS CHECK
0109					
0110					THIS ROUTINE TESTS THE KEYBOARD STATUS AND RETURNS
0111					WITH THE TEST BITS SET.
0112					
0113	IN	STAPT			GET STATUS WORD
0114	ANI	KDR			TEST KEYBOARD BIT
0115	RET	.			FLAGS ARE SET
0116					
0117					KEYBOARD DATA INPUT
0118					
0119					
0120					THIS ROUTINE, ENTRY AT KREAD, GETS THE DATA FROM THE
0121					KEYBOARD. ON RETURN THE CHARACTER IS IN REGISTER 'A'.
0122					
0123	POP	H			JUMP TABLE ENTRY POINT
0124	CALL	KSTAT			CHECK STATUS
0125	JNZ	KREAD			WAIT FOR INPUT
0126	IN	KDATA			GET DATA
0127	RET	.			GO BACK WITH IT
0128					
0129					SERIAL INPUT STATUS CHECK
0130					
0131					
0132	IN	SERST			GET SERIAL STATUS WORD
0133	ANI	SDR			TEST FOR SERIAL DATA READY
0134	RET	.			FLAGS ARE SET
0135					
0136					SERIAL DATA INPUT
0137					
0138					
0139	POP	H			RESTORE HL FROM JUMP TABLE ENTRY
0140	CALL	SSTAT			NORMAL ENTRY POINT
0141	JZ	SREAD			WAIT FOR INPUT
0142	IN	SDATA			GET DATA BYTE
0143	RET	.			WE HAVE IT
0144					
0145					SERIAL DATA OUTPUT
0146					
0147					
0148	POP	H			JUMP TABLE ENTRY POINT
0149	IN	SERST			GET PORT STATUS
0150	RAL	.			PUT HIGH BIT IN CARRY
0151	JNC	SDROT			LOOP UNTIL TRANSMITTER BUFFER IS EMPTY
0152	MOV	A,B			GET THE CHARACTER BACK
0153	OUT	SDATA			SEND IT OUT
0154	RET	.			AND WE'RE DONE
0155					
0156					
0157					

== CONSOL ==
 COPYRIGHT 1976

PROCESSOR TECHNOLOGY CORP.
 6200 HOLLIS STREET
 EMERYVILLE, CALIF. 94608

C04B	0158	*
C04B	0159	*
C04B	0160	*
C04B	0161	*
C04B	0162	*
C04B	0163	*
C04B	0164	*
C04B	0165	*
C04B	0166	*
C04B	0167	*
C04B	0168	*
C04B	0169	*
C04B	0170	*
C04B	0171	*
C04B	0172	*
C04B	0173	*
C04B	0174	*
C04B	0175	*
C04B	0176	*
C04B	0177	VDMOT
C04C	0178	VDMOI
C04D	0179	
C04E	0180	
C04F	0181	
C050	0182	
C053	0183	
C056	0184	*
C056	0185	GOBACK
C059	0186	
C05A	0187	
C05C	0188	
C05D	0189	GOBK
C05E	0190	
C05F	0191	
C060	0192	
C061	0193	
C062	0194	*
C062	0195	*
C062	0196	TSRCH
C063	0197	
C064	0198	
C067	0199	
C068	0200	
C069	0201	
C06C	0202	
C06C	0203	
C06D	0204	
C070	0205	*
C071	0206	*
C071	0207	*
C071	0208	*
C071	0209	*
C071	0210	*

VIDEO DISPLAY DRIVER ROUTINES

THESE ROUTINES ALLOW FOR STANDARD VIDEO TERMINAL OPERATIONS. ON ENTRY, THE CHARACTER FOR OUTPUT IS IN REGISTER B AND ALL REGISTERS ARE UNALTERED ON RETURN.

THE 'CONSOL' VERSION OF THIS ROUTINE IS A MINIMUM IMPLEMENTATION OF ROUTINES ORIGINATED BY:

IAN KETTLEBOROUGH
 OF
 COLLEGE STATION, TEXAS

SOLOS AND SOLED CONTAIN THE ESC SEQUENCES AND OTHER FULL IMPLEMENTATION FEATURES.

PUSH	H	SAVE EVERYBODY
PUSH	D	ENTRY FROM DEVICE SELECT
PUSH	B	
PUSH	PSW	
MOV	A,B	SAVE IN B...STRIP PARITY BEFORE SCREEN!
LXI	H,TBL	
CALL	TSRCH	GO PROCESS

GET SCREEN ADDRESS

CURSOR IS BACK ON

RESTORE ALL REGISTERS

EXIT FROM VDMOT

GET CHR FROM TABLE

ZERO IS THE LAST
 TEST THE CHR
 POINT FORWARD

FOUND ONE...SAVE ADDRESS
 REMOVE CURSOR

THIS ROUTINE DISPATCHES TO THE ADDRESS POINTED TO BY THE HL REGISTER PAIR. THE RETURN ADDRESS IS THE LAST ENTRY ON THE STACK.

PROCESSOR TECHNOLOGY CORP.
6200 HOLLIS STREET
EMERYVILLE, CALIF. 94608

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 *			
C077 23	0217 *			
C078 23	0218 NEXT	INX	H	GO TO NEXT
C079 23	0219	INX	H	
C080 C3 62 C0	0220	JMP	TSRCH	
C081	0221 *			
C082	0222 *			
C083 78	0223 CHAR	MOV	A,B	GET CHARACTER
C084 B7	0224	ORA	A	
C085 7F	0225	RZ	.	RETURN IF A NULL
C086 FE 7F	0226	CPI	7FH	IS IT A DEL?
C087 C8	0227	RZ	.	GO BACK IF SO
C088	0228 *			
C089	0229 *			
C090	0230 *			
C091 CD 21 C1	0231 OCHAR	CALL	VDADD	GET SCREEN ADDRESS
C092 78	0232	MOV	A,B	GET CHARACTER FOR OUTPUT
C093 E6 7F	0233	ANI	7FH	NO HIGH BITS GO PAST HERE
C094 77	0234	MOV	M,A	PUT CHR ON SCREEN
C095 3A 00 C8	0235	LDA	NCHAR	GET CHARACTER POSITION
C096 FE 3F	0236	CPI	63	END OF LINE?
C097 DA AD C0	0237	JC	OK	
C098 3A 01 C8	0238	LDA	LINE	
C099 FE 0F	0239	CPI	15	END OF SCREEN?
C100 C2 AD C0	0240	JNZ	OK	
C101	0241 *			
C102	0242 *			END OF SCREEN...ROLL UP ONE LINE
C103	0243 *			
C104 AF	0244 SCROLL	XRA	A	BACK TO FIRST CHAR POSITION
C105 32 00 C8	0245	STA	NCHAR	
C106 4F	0246	MOV	C,A	
C107 CD 28 C1	0247	CALL	VDAD	CALCULATE LINE TO BE BLANKED
C108	0248	XRA	A	CLEAR IT
C109 CD E9 C0	0249	CALL	CLINI	
C110 3A 02 C8	0250	LDA	BOT	
C111 3C	0251	INR	A	
C112 E6 0F	0252	ANI	0FH	
C113 C3 DD C0	0253	JMP	ERAS3	
C114	0254 *			
C115	0255 *			INCREMENT LINE COUNTER IF NECESSARY
C116	0256 *			
C117 3A 00 C8	0257 OK	LDA	NCHAR	GET CHR POSITION
C118 3C	0258	INR	A	
C119 32 00 C8	0259	STA	NCHAR	STORE THE NEW
C120 FE 40	0260	CPI	64	
C121 D8	0261	RC		
C122 AF	0262	XRA	A	WE'RE PAST THE END... REWIND THE COUNT
C123 32 00 C8	0263	STA	NCHAR	

PROCESSOR TECHNOLOGY CORP.
6200 HOLLIS STREET
EMERYVILLE, CALIF. 94608

== CONSOL ==
COPYRIGHT 1976

C0BB 3A 01 C8	0264	LDA	LINE	GET THE LINE COUNT
C0BE 3C	0265	INR	A	
C0BF E6 0F	0266	ANI	0FH	
C0C1 32 01 C8	0267	STA	LINE	MOD 15 INCREMENT
C0C4 C9	0268	RET		STORE THE NEW
C0C5	0269	*		
C0C5	0270	*	ERASE SCREEN	
C0C5	0271	*		
C0C5 21 00 CC	0272	PERSE		
C0C8 36 A0	0273	LXI	H,VDMEM	POINT TO SCREEN
C0CA	0274	MVI	M,80H+	THIS IS THE CURSOR
C0CA 23	0275	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	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	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	CALL	VDADD	GET CURRENT SCREEN ADDRESS
C0E6 3A 00 C8	0292	LDA	NCHAR	CURRENT CURSOR POSITION
C0E9 FE 40	0293	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	CLINI	LOOP TO END OF LINE
C0F3	0299	*		
C0F3	0300	*	HOME CURSOR	
C0F3	0301	*		
C0F3 AF	0302	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	LINE	
C0FD FE 0F	0309	LDA	15	
C0FF C8	0310	CPI	.	HOW FAR IS DOWN?
C100 3C	0311	RZ	A	
C101 C3 C1 C0	0312	INR	CUR	
C104	0313	JMP		
C104	0314	*		
C104	0315	*	ROUTINE TO MOVE THE CURSOR UP ONE LINE	
C104 3A 01 C8	0316	PUP	LINE	GET LINE COUNT

== CONSOL ==
 COPYRIGHT 1976

PROCESSOR TECHNOLOGY CORP.
 6200 HOLLIS STREET
 EMERYVILLE, CALIF. 94608

0370	MOV	A,M	STRIP OFF THE CURSOR
0371	ANI	7FH	
0372	MOV	M,A	
0373	RET		
0374 *	ROUTINE TO BACKSPACE		
0375 *			
0376 *			
0377	CALL	PLEFT	
0378	CALL	VDADD	GET SCREEN ADDRESS
0379	MVI	M,	PUT A BLANK THERE
0380	RET		
0381 *	ROUTINE TO PROCESS A CARRIAGE RETURN		
0382 *			
0383 *			
0384	CALL	CLINE	CLEAR FROM CURRENT CURSOR TO END OF LINE
0385	XRA	A	REWIND IT
0386	JMP	PCUR	AND STORE THE NEW VALUE
0387 *	ROUTINE TO PROCESS LINEFEED		
0388 *			
0389 *			
0390	LDA	LINE	GET LINE COUNT
0391	CPI	15	ARE WE AT THE BOTTOM?
0392	JNC	SC	
0393	INR	A	
0394	JMP	CUR	ONE MORE LINE UP
0395 *	ROUTINE TO PROCESS LINEFEED		
0396	XRA	A	
0397	JMP	SROL	
0398 *			
0399 *			
0400 *			
0401 *			
0402 *			
0403 *			
0404 *			
0405	DB	CLEAR	SCREEN
0406	DW	PERSE	
0407	DW	UP	CURSOR
0408	DW	PUP	"
0409	DW	DOWN	"
0410	DW	PDOWN	"
0411	DW	LEFT	"
0412	DW	PLEFT	"
0413	DW	RIGHT	"
0414	DW	PRIT	"
0415	DW	HOME	
0416	DW	PHOME	CARRIAGE RETURN
0417	DW	CR	
0418	DW	PCR	
0419	DW	LF	LINE FEED
0420	DW	PLF	
0421	DW	BACKS	BACK SPACE
0422	DW	PBACK	

THIS TABLE DEFINES THE CHARACTERS FOR SPECIAL
 PROCESSING. IF THE CHARACTER IS NOT IN THE TABLE IT
 GOES TO THE SCREEN.

PROCESSOR TECHNOLOGY CORP.
6200 HOLLIS STREET
EMERYVILLE, CALIF. 94608

--- CONSOL ---
COPYRIGHT 1976

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 *			
C184	0428 *			
C184 4C C0	0429 OTAB	DW	VDMOL	VDM 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 *			
C18C	0435 *			
C18C 27 C0	0436 ITAB	DW	KREAL	KEYBOARD INPUT
C18E 36 C0	0437	DW	SREAL	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 *			
C194	0443 *			
C194	0444 *			
C194	0445 *			
C194 54 45	0446	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'	
C1AA 00 00	0457	DW	0	
C1AC 00	0458	DB	0	
C1AD	0459 *			
C1AD	0460 *			
C1AD	0461 *			
C1AD	0462 *			
C1AD AF	0463 ERROT	XRA	A	
C1AE 32 04 C8	0464	STA	IPOINT	DEFAULT TO SCREEN
C1B1 32 03 C8	0465	STA	OPOINT	DEFAULT TO SCREEN
C1B4	0466 *			
C1B4	0467 *			
C1B4	0468 *			
C1B4	0469 *			
C1B4	0470 *			
C1B4	0471 *			
C1B4	0472 *			
C1B4	0473 *			
C1B4	0474 *			
C1B4 31 00 CC	0475	COMND	LXI	SP,SYSTP SET STACK POINTER

THIS TABLE DESCRIBES THE VALID COMMANDS FOR CONSOL

COMMAND TABLE

CONTAB ASC 'TE'
DW TERM
ASC 'DU'
DW DUMP
ASC 'EN'
DW ENTER
ASC 'EX'
DW EXEC
ASC 'TL'
DW TLOAD
ASC 'BA'
DW 0
DB 0

SPECIAL COMMAND TO EXECUTE 0

END OF TABLE MARK

CONSOL PORT ERROR HANDLER

--- COMMAND MODE ---

THIS ROUTINE GETS AND PROCESSES COMMANDS

== CONSOL ==
 COPYRIGHT 1976

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	*		
C1C3		0484	*		
C1C3		0485	*		
C1C3		0486	*		
C1C3		0487	*		
C1C3		0488	*		
C1C3		0489	*		
C1C3		0490	*		
C1C3	CD 28 C0	0491	GCLIN	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	CALL	VDMOT	
C1CF	C3 C3 C1	0496	JMP	GCLIN	
C1D2		0497	*		
C1D2		0498	*		
C1D2		0499	*		
C1D2	FE 0D	0500	PROCS	CR	
C1D4	CA DD C1	0501	JZ	CRPRC	
C1D7	FE 0A	0502	CPI	LF	
C1D9	C8	0503	RZ	.	
C1DA	C3 C3 C1	0504	JMP	GCLIN	
C1DD		0505	*		
C1DD		0506	*		
C1DD	CD E3 C0	0507	CRPRC		
C1E0	C9	0508	CALL	CLINE	
C1E1		0509	RET	.	
C1E1		0510	*		
C1E1		0511	*		
C1E1		0512	*		
C1E1	CD 3D C1	0513	COPRC	CREM	
C1E4	0E 01	0514	MVI	C,1	
C1E6	CD 25 C1	0515	CALL	VDAD2	
C1E9	EB	0516	XCHG		
C1EA	CD 32 C2	0517	CALL	SCHR	
C1ED	CA 8E C3	0518	JZ	ERRI	
C1F0	EB	0519	XCHG	.	
C1F1	11 94 C1	0520	LXI	D,COMTAB	
C1F4		0521	*		
C1F4		0522	*		
C1F4		0523	*		
C1F4		0524	*		
C1F4		0525	*		
C1F4		0526	*		
C1F4	1A	0527	FDCOM	LDAX	
C1F5	B7	0528	ORA	A	

THIS ROUTINE READS A COMMAND LINE FROM THE SYSTEM
 KEYBOARD AND PROCESSES IT TO THE SCREEN.

C/R TERMINATES THE SEQUENCE ERASING ALL CHARS TO THE
 RIGHT OF THE CURSOR
 L/F TERMINATES THE SEQUENCE
 MODE RESTARTS THE COMMAND LINE.

ERASE THE REMAINING LINE PRIOR TO RETURN
 IF SO GO PROCESS
 NO CONTROL CHARS TO SCREEN

CLEAR REMAINING LINE
 NOW PROCESS

REMOVE THE CURSOR
 SET FOR CHARACTER POSITION
 GET SCREEN ADDRESS

SCAN PAST BLANKS
 NO COMMAND?
 HL HAS FIRST CHR
 POINT TO COMMAND TABLE

THIS ROUTINE SEARCHES THROUGH A TABLE, POINTED TO
 BY 'DE', FOR A DOUBLE CHARACTER MATCH OF THE 'HL'
 MEMORY CONTENT. IF NO MATCH IS FOUND THE SCAN ENDS
 BY PLACING A QUESTION MARK WITHIN THE SEARCH STRING.

TEST FOR TABLE END

--= CONSOL ==-
COPYRIGHT 1976

** ALS-8 PROGRAM DEVELOPMENT SYSTEM **

PROCESSOR TECHNOLOGY CORP.
6200 HOLLIS STREET
EMERYVILLE, CALIF. 94608

PAGE 11

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	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		
C215 06 3E	0557	CALL	CRLF	
C217 C3 4B C0	0558	MVI	B,'>'	THE PROMPT
C21A	0559 *	JMP	VDMOT	PUT IT ON THE SCREEN
C21A 06 0A	0560	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	C,12	MAXIMUM COMMAND STRING
C226 1A	0569	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	SCAN TO FIRST NON BLANK CHR WITHIN 10
			C,10	

== CONSOL ==
COPYRIGHT 1976

PROCESSOR TECHNOLOGY CORP.
6200 HOLLIS STREET
EMERYVILLE, CALIF. 94608

C234 1A	0582	SCHRI	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	SCHRI		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	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,SYSTRAM		POINT TO SYSTEM RAM
C26A	0632	*			
C26A 77	0633	CLERA	MOV	M,A	
C26B 23	0634	INX	H		

PROCESSOR TECHNOLOGY CORP.
6200 HOLLIS STREET
EMERYVILLE, CALIF. 94608

C26C 0C	0635	INR	C	
C26D C2 6A C2	0636	JNZ	CLERA	CLEAR FIRST 256 BYTES
C270	0637 *			
C270	0638 *			
C270	0639 *			
C270	0640 *			
C270	0641 *			
C270	0642 *			
C270	0643 *			
C270	0644 *			
C270	0645 *			
C270	0646 *			
C270	0647 *			
C270	0648 *			
C270	0649 *			
C270	0650 *			
C270 31 00 CC	0651	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	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	CALL	SDROT	OUTPUT IT TO THE SERIAL PORT
C290 CD 31 C0	0665	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 *			
C2A1	0676 *			
C2A1	0677 *			
C2A1	0678 *			
C2A1	0679 *			
C2A1	0680 *			
C2A1	0681 *			
C2A1	0682 *			
C2A1	0683 *			
C2A1	0684 *			
C2A1	0685 *			
C2A1	0686 *			
C2A1	0687 *			

THIS ROUTINE GETS CHARACTERS FROM THE SYSTEM KEYBOARD AND OUTPUTS THEM TO THE SERIAL OUTPUT PORT. IT IS INTENDED TO CONFIGURE THE SOL AS A STANDARD VIDEO TERMINAL. COMMAND KEYS ARE NOT OUTPUT TO THE OUTPUT PORT BUT ARE INTERPRETED AS DIRECT SOL COMMANDS. THE MODE COMMAND, RECEIVED BY THE KEYBOARD, PUTS THE SOL IN THE COMMAND MODE.

THIS ROUTINE DUMPS CHARACTERS FROM MEMORY TO THE CURRENT OUTPUT DEVICE. (WITH CONSOL ALL OUTPUT GOES TO THE SCREEN). ALL VALUES ARE DISPLAYED AS ASCII HEX.

THE COMMAND FORM IS AS FOLLOWS:

Dump addr1 addr2

THE VALUES FROM ADDR1 TO ADDR2 ARE THEN OUTPUT TO THE OUTPUT DEVICE. IF ONLY ADDR1 IS SPECIFIED THEN THE VALUE AT THAT ADDRESS IS OUTPUT.

PROCESSOR TECHNOLOGY CORP.
6200 HOLLIS STREET
EMERYVILLE, CALIF. 94608

== CONSOL ==-
COPYRIGHT 1976

0688 *	DUMP	CALL	SCONV	SCAN TO FIRST ADDRESS AND CONVERT IT
0689		PUSH	H	SAVE THE VALUE
0690		CALL	SCHR	GET THE NEXT
0691		POP	H	
0692		JZ	POVER	NO SECOND VALUE
0693		PUSH	H	
0694		CALL	SHEX	GET SECOND
0695		POP	D	THIS IS THE FIRST
0696		JMP	NPASS	MIND BENDERS
0697		MOV	D,H	NO SECOND PARAMETER COPY FIRST TO DE
0698 *	POVER	MOV	E,L	
0699		MOV	.	HL HAS START, DE HAS END
0700	NPASS	XCHG		
0701		CALL	CRLF	
0702 *	DLOOP	IN	KDATA	'MODE KEY' WILL ESCAPE THE DUMP
0703		CPI	MODE	
0704		JZ	COMND	
0705		CALL	ADOUT	OUTPUT ADDRESS
0706		CALL	BOUT	ANOTHER SPACE TO KEEP IT PRETTY
0707		MVI	C,16	VALUES PER LINE
0708		MOV	A,M	GET THE CHR
0709 *	DLP1	PUSH	B	SAVE VALUE COUNT
0710		CALL	HBOUT	SEND IT OUT WITH A BLANK
0711		CALL	ACOMP	COMPARE ADDRESSES
0712		JNC	COMND	ALL DONE
0713		POP	B	VALUES PER LINE
0714		INX	H	
0715		DCR	C	BUMP THE LINE COUNT
0716		JNZ	DLP1	NOT ZERO IF MORE FOR THIS LINE
0717		JMP	DLOOP	DO A LFCR BEFORE THE NEXT
0718				
0719				
0720				
0721 *				
0722 *				
0723 *				
0724	ADOUT	MOV	A,H	H FIRST
0725		CALL	HEOUT	
0726		MOV	A,L	THEN L FOLLOWED BY A SPACE
0727 *				
0728	HBOUT	CALL	HEOUT	
0729	BOUT	MVI	B,	
0730		JMP	VDMOT	CONSOL PUTS IT ON THE SCREEN
0731 *				
0732	HEOUT	MOV	C,A	GET THE CHARACTER
0733		RRC	.	MOVE THE HIGH FOUR DOWN
0734		RRC		
0735		RRC		
0736		RRC		
0737		CALL	HEOUT	PUT THEM OUT
0738 *		MOV	A,C	THIS TIME THE LOW FOUR
0739				
0740	HEOUT	ANI	0FH	FOUR ON THE FLOOR

== CONSOL ==-
COPYRIGHT 1976

PROCESSOR TECHNOLOGY CORP.
6200 HOLLIS STREET
EMERYVILLE, CALIF. 94608

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	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	*		
C307	0758	*	ENTER COMMAND	
C307	0759	*		
C307	0760	*	THIS ROUTINE GETS VALUES FROM THE KEYBOARD AND ENTERS	
C307	0761	*	THEM INTO MEMORY. THE INPUT VALUES ARE SCANNED FOLLOWING	
C307	0762	*	A STANDARD 'GCLIN' INPUT SO ON SCREEN EDITING MAY TAKE	
C307	0763	*	PLACE PRIOR TO THE LINE TERMINATOR. A BACK SLASH '/'	
C307	0764	*	ENDS THE ROUTINE AND RETURNS CONTROL TO THE COMMAND MODE.	
C307 CD 3E C2	0765	ENTER	SCONV	SCAN OVER CHARS AND GET ADDRESS
C30A E5	0766	PUSH	H	SAVE ADDRESS
C30B	0767	*		
C30B CD 1A C2	0768	ENLOP	CALL	
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	C,3	NO MORE THAN THREE SPACES BETWEEN VALUES
C31E CD 34 C2	0000	CALL	SCHRI	SCAN TO NEXT VALUE
C321 CA 0B C3	0778	JZ	ENLOP	LAST ENTRY FOUND START NEW LINE
C324 1A	0779	ENLO2	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	*		
C335	0793	*	EXECUTE COMMAND	

== CONSOL ==
COPYRIGHT 1976

C374	*	THIS ROUTINE GETS THE FOLLOWING PARAMETER AND DOES A
C375	*	PROGRAM JUMP TO THE LOCATION GIVEN BY IT. IF PROPER
C376	*	STACK OPERATIONS ARE USED WITHIN THE EXTERNAL PROGRAM
C377	*	IT CAN DO A STANDARD 'RET'URN TO THE CONSL COMMAND MODE.
C378	*	
C379	*	
C380	EXEC	SCAN PAST BLANKS AND GET PARAMETER
C381		GO.....(AMD TAKE NOTE)
C382	*	
C383	*	
C384	*	
C385	*	
C386	*	
C387	*	
C388	*	THIS ROUTINE READS FROM EITHER TAPE UNIT PLACING
C389	*	THE READ DATA INTO MEMORY. WHILE SPACE WITHIN CONSOL
C390	*	DOES NOT ALLOW FOR 'STANDARD' TAPE ROUTINES THIS
C391	*	COMMAND WILL LOAD SOL- BASIC5 AND OTHER STANDARD SOL
C392	*	SYSTEM SOFTWARE FOR DIRECT EXECUTION.
C393	*	
C394	*	
C395	TLOAD	SCAN TO SPEED PARAMETER
C396	JZ	DEFAULT TO HIGH SPEED IF NONE
C397	CALL	CONVERT IT
C398	MOV	GET VALUE
C399	ANI	ONLY BIT ZERO COUNTS
C400	MVI	PRETEND ITS SLOW
C401	JNZ	
C402	* DFLT	MAKE IT FAST
C403	SETSP	TAPEL+TAPE2 CONSOL STARTS BOTH TAPES
C404	OUT	START TAPES AND SELECT SPEED
C405	CALL	WAIT WHILE THE TAPE UNIT WINDS UP
C406	IN	CLEAR THE UART FLAGS
C407	* TLOD1	READ PAST HEADER
C408	JNZ	IF ERROR START OVER
C409	* LHL	GET BLOCK SIZE
C410	XCHG	...TO DE
C411	LHLD	GET LOAD ADDRESS
C412	* LOLOOP	GET COUNT
C413	ORA	E
C414	JZ	TOFF
C415	LXI	B,-256
C416	XCHG	.
C417	DAD	B
C418	JNC	LBLK
C419	MVI	B,0
C420	* RDBLK	C,0
C421	XCHG	.
C422		COUNT IS ZERO-TURN OFF TAPE AND RETURN
C423		THIS MANY PRIOR TO CRC TEST
C424		COUNT TO HL
C425		A LITTLE MATH
C426		NO CARRY, IT'S THE LAST BLOCK
C427		256 TO READ
C428		ZERO THE CRC
C429		ROUND ROBIN
C430	*	

== CONSOL ==-
COPYRIGHT 1976

PROCESSOR TECHNOLOGY CORP.
6200 HOLLIS STREET
EMERYVILLE, CALIF. 94608

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	STORE THE NEW
C37B	4F			0852		MOV	C,A	COUNT DOWN
C37C	05			0853		DCR	B	STILL MORE IF NOT ZERO
C37D	C2	73	C3	0854		JNZ	RTBYT	
C380				0855	*			
C380	CD	C1	C3	0856		CALL	CRCK	CHECK CRC AND FALL THROUGH TO ERROR IF NO GOOD
C383	CA	61	C3	0857		JZ	LOLOOP	TEST OK
C386				0858	*			
C386	06	07		0859	TERR	MVI	B,'G'-40H	BELL CHARACTER
C388	CD	4B	C0	0860		CALL	VDROT	PUT IT ON THE SCREEN
C38B	C3	B4	C1	0861		JMP	COMND	
C38E				0862	*			
C38E				0863	*			
C38E				0864	*			
C38E				0865	*			
C38E	EB			0866	ERR1	XCHG		GET SCAN ADDRESS
C38F	36	3F		0867	ERR2	MVI	M,'?'	PUT A QUESTION MARK THERE
C391	C3	B4	C1	0868		JMP	COMND	AND GO TO COMMAND MODE
C394				0869	*			
C394				0870	*			
C394				0871	*			
C394	06	0A		0872	RHEAD	MVI	B,10	FIND 10 NULLS
C396	DB	FA		0873	RHEAL	IN	STAPT	GET A BYTE
C398	E6	40		0874		ANI	TDR	
C39A	CA	96	C3	0875		JZ	RHEAL	
C39D	DB	FB		0876		IN	TDATA	
C39F	B7			0877		ORA	A	IGNORE ERROR CONDITIONS
C3A0	C2	94	C3	0878		JNZ	RHEAD	ZERO?
C3A3	05			0879		DCR	B	
C3A4	C2	96	C3	0880		JNZ	RHEAL	LOOP UNTIL 10 IN A ROW
C3A7				0881	*			
C3A7				0882	*			
C3A7				0883	*			
C3A7	CD	C6	C3	0884	SOHL	CALL	TAPIN	WAIT FOR THE START CHARACTER
C3AA	3D			0885		DCR	A	
C3AB	C2	A7	C3	0886		JNZ	SOHL	WAIT FOR A '1'
C3AE				0887	*			
C3AE				0888	*			
C3AE				0889	*			
C3AE	21	05	C8	0890		LXI	H,THEAD	POINT TO BUFFER
C3B1	01	00	10	0891		LXI	B,HLEN*256	LENGTH OF HEADER IN 'B',C<0
C3B4				0892	*			
C3B4	CD	C6	C3	0893	RHED1	CALL	TAPIN	GET BYTE
C3B7	77			0894		MOV	M,A	STORE IT
C3B8	23			0895		INX	H	INCREMENT ADDRESS
C3B9	A9			0896		XRA	C	NOW CALCULATE THE CRC
C3BA	2F			0897		CMA	.	INSIDE OUT AND UPSIDE DOWN
C3BB	91			0898		SUB	C	SQUEEZE IT

== CONSOL ==
COPYRIGHT 1976

C3BC 4F	0899	MOV	C,A	AND SAVE AGAIN
C3BD 05	0900	DCR	B	WHOLE HEADER YET?
C3BE C2 B4 C3	0901	JNZ	RHED1	LOOP UNTIL DONE
C3C1	0902	*		
C3C1	0903	*		THIS ROUTINE GETS THE NEXT BYTE AND COMPARES IT
C3C1	0904	*		TO THE VALUE IN REGISTER C. THE FLAGS ARE SET ON
C3C1	0905	*		RETURN.
C3C1	0906	*		
C3C1 CD C6 C3	0907	CRCK		
C3C4 B9	0908	CALL	TAPIN	GET CRC BYTE
C3C5 C9	0909	CMP	C	COMPARE IT WITH CALCULATED
C3C6	0910	RET		
C3C6	0911	*		
C3C6	0912	*		THIS ROUTINE GETS THE NEXT AVAILABLE BYTE FROM THE
C3C6	0913	*		TAPE. WHILE WAITING FOR THE BYTE THE KEYBOARD IS TESTED
C3C6	0914	*		FOR A 'MODE' COMMAND. IF RECEIVED THE TAPE LOAD IS
C3C6	0915	*		TERMINATED AND A RETURN TO THE COMMAND MODE IS MADE.
C3C6	0916	*		
C3C6 DB FA	0917	TAIN		
C3C8 E6 40	0918	IN	STAPT	CHECK STATUS
C3CA C2 D7 C3	0919	ANI	TDR	
C3CD DB FC	0920	JNZ	TREDY	ONE IS AVAILABLE
C3CF FE 80	0921	IN	KDATA	CHECK FOR MODE WHILE WE'RE WAITING
C3D1 CA B4 C1	0922	CPI	MODE	
C3D4 C3 C6 C3	0923	JZ	COMND	MODE WAS GIVEN..ABORT OPERATION
C3D7	0924	JMP	TAPIN	NOT MODE...STAY IN LOOP
C3D7 DB FA	0925	IN	STAPT	
C3D9 E6 18	0926	ANI	TFE+TOE	DATA ERROR?
C3DB C2 86 C3	0927	JNZ	TERR	IF FRAMING OR OVERRUN ERROR
C3DE DB FB	0928	IN	TDATA	GET THE DATA
C3E0 C9	0929	RET		
C3E1	0930	*		
C3E1	0931	*		THIS ROUTINE CALCULATES THE LENGTH OF THE LAST BLOCK
C3E1	0932	*		
C3E1 01 FF FF	0933	LBK		
C3E4 09	0934	LXI	B,-1	COMPLEMENT HL
C3E5 23	0935	DAD	B	TWO'S
C3E6 45	0936	INX	H	LENGTH TO REGISTER B
C3E7 21 00 00	0937	MOV	B,L	TELL DE WE'RE DONE
C3EA C3 70 C3	0938	LXI	H,0	ONWARD TO THE END
C3ED	0939	JMP	RDBLK	
C3ED	0940	*		THIS ROUTINE TURNS THE TAPE UNITS OFF
C3ED	0941	*		
C3ED AF	0942	TOFF		
C3EE D3 FA	0943	XRA	A	
C3F0 C9	0944	OUT	STAPT	GIVE COMMAND
C3F1	0945	RET	.	AND GRIND TO A SLOW STOP
C3F1	0946	*		
C3F1 11 00 00	0947	DELAY		
C3F4 1B	0948	DLOP1		
C3F5 7A	0949	LXI	D,0	START LOOP
C3F6 B3	0950	DCX	D	DOWN COUNT
C3F7 C2 F4 C3	0951	MOV	A,D	
		ORA	E	TEST FOR ZERO
		JNZ	DLOP1	IF NOT

PROCESSOR TECHNOLOGY CORP.
6200 HOLLIS STREET
EMERYVILLE, CALIF. 94608

== CONSOL ==
COPYRIGHT 1976

C3FB	1005 *	EQU	1	KEYBOARD DATA READY
C3FB	1006 KDR	EQU	2	PARALLEL DATA READY
C3FB	1007 PDR	EQU	4	PARALLEL DEVICE READY
C3FB	1008 PXDR	EQU	8	TAPE FRAMING ERROR
C3FB	1009 TFE	EQU	16	TAPE OVERFLOW ERROR
C3FB	1010 TOE	EQU	64	TAPE DATA READY
C3FB	1011 TDR	EQU	128	TAPE TRANSMITTER BUFFER EMPTY
C3FB	1012 TTBE	EQU		
C3FB	1013 *	EQU	1	SCROLL OK FLAG
C3FB	1014 SOK	EQU	64	TAPE ONE 'ON' BIT
C3FB	1015 *	EQU	128	TAPE TWO
C3FB	1016 TAPE1	EQU		
C3FB	1017 TAPE2	EQU		
C3FB	1018 *			
C3FB	1019 *			
C3FB	1020 *			
C3FB	1021 *			
C3FB	1022 *			
C3FB	1023 *			
C3FB	1024 *			
C800	1025 *	ORG	0C800H	START OF 1K RAM AREA
C800	1026 SYSRAM	EQU	\$	START OF SYSTEM RAM
C800	1027 SYSIP	EQU	\$+1024	STACK IS AT THE TOP
C800	1028 *			
C800	1029 *			
C800	1030 *			
C800	1031 *			
C800	1032 NCHAR	DS	1	CURRENT CHARACTER POSITION
C801	1033 LINE	DS	1	CURRENT LINE POSITION
C802	1034 BOT	DS	1	BEGINNING OF TEXT DISPLACEMENT
C803	1035 OPORT	DS	1	OUTPUT PORT
C804	1036 IPORT	DS	1	INPUT PORT
C805	1037 *			
C805	1038 *			
C805	1039 *			
C805	1040 THEAD	DS	5	NAME
C80A	1041	DS	1	THIS BYTE MUST BE ZERO
C80B	1042 HTYPE	DS	1	TYPE
C80C	1043 BLOCK	DS	2	BLOCK SIZE
C80E	1044 LOADR	DS	2	LOAD ADDRESS
C810	1045 XEQAD	DS	2	AUTO EXECUTE ADDRESS
C812	1046 HSPR	DS	3	SPARES
C815	1047 *			
C815	1048 HLEN	EQU	\$-THEAD	LENGTH OF HEADER
C815	1049 *			
C815	1050 *			
ACOMP	C302	ADOUT	C016	AOUT
BACKS	W05F	AINP	C007	C007
BOT	C802	BLANK	0020	BLOCK
CLERA	C26A	CHAR	C07B	CLEAR
COMTA	C194	COPRC	C0E3	COMND
CRCK	C3C1	CRLF	C1E1	CR
			C21A	CRPRC
			C1DD	C1DD

== CONSOL ==-
COPYRIGHT 1976

CUR C0C1 CX
DISPT C071 DLOOP
DOWN 009A DSTAT
ENLO2 C324 ENLOP
ERAS2 C0D6 ERAS3
ERROT C1AD EXEC
GOBAC C056 GOBK
HCOV1 C25B HE0U1
HOME 008E HSPR
ITAB C18C KDATA
KREAL C027 KREAD
LEFT 0081 LF
LOADR C00E LOLOO
NCOM C20B NEXT
OK C0AD OPORT
PBACK C145 PCR
PDOWN C0FA PDR
PLEFT C10D PLF
PROCS C1D2 PROMP
RDBLK C370 RHEAL
RIGHT 0093 RTBYT
SC C161 SCD
SCONV C23E SCROL
SDR 0040 SLROT
SEROT C040 SERST
SHEL C247 SHEX
SOHL C3A7 SOK
SPE 0004 SREAL
SSTAT C031 STAPT
SYSRA C800 SYSTP
TAPIN C3C6 TBL
TERM C270 TERR
TIN C290 TLOAD
TOFF C3ED TOUT
TTBE 0080 UP
VDADD C121 VDMEM
XEQAD C810 *

0018
C2B7
00FE
C30B
C0DD
C335
C05D
C2F3
C812
00FC
C028
000A
C361
C076
C803
C14E
0002
C155
C212
C396
C373
0001
C098
C041
00F8
C244
0001
C036
00FA
CC00
C165
C386
C339
C28D
0097
CC00

DELAY
DLOP1
DUMP
ENTER
ERR1
FDCOM
HBOUT
HEOUT
HTYPE
KDR
KSTAT
LINE
MODE
NPASS
OTAB
PCUR
PERSE
POVER
PUP
RHEAD
SBLK
SCHR
SCTS
SDSK
SETSP
SINP
SOUT
SREAD
STBE
TAPE1
TDATA
TFE
TLOD1
TREDY
VDAD
VDMO1

C3F1
C3F4
C2A1
C307
C38E
C1F4
C2E2
C2EA
C80B
0001
C022
C801
0080
C2B6
C184
C113
C0C5
C2B4
C104
C394
C224
C232
0020
0002
C34B
C013
C004
C037
0080
0040
00FB
0008
C354
C3D7
C128
C04C

DFLT
DLP1
ENLO1
ERAS1
ERR2
GCLIN
HCONV
HLEN
IPOINT
KIN
LBLK
LOADK
NCHAR
OCHAR
OUTH
PDATA
PHOME
PRIT
PXDR
RHED1
SBLK1
SCHR1
SDATA
SENSE
SFE
SOE
SPACE
SROL
STRTA
TAPE2
TDR
THREAD
TOE
TSRCH
VDAD2
VDMOT

C34A
C2C9
C31C
C0CA
C38F
C1C3
C24B
0010
C804
C279
C3E1
008C
C800
C081
C2FE
00FD
C0F3
C117
0004
C3B4
C226
C234
00F9
00FF
0008
0010
0020
C09C
C265
0080
0040
C805
0010
C062
C125
C04B