

# PROM-100

SD #7140044 REV. C  
AUGUST, 1981

**SDSYSTEMS**

A SYNTECH COMPANY

**OPERATIONS  
MANUAL**

**PROM-100  
PROM PROGRAMMER**

**COPYRIGHT © 1979  
BY SD SYSTEMS  
JULY 1979**

**REVISION C  
AUGUST, 1981**

## TABLE OF CONTENTS

SECTION	DESCRIPTION	PAGE
I-1	INTRODUCTION	1
I-2	GENERAL DESCRIPTION	1
I-3	PHYSICAL DESCRIPTION	1
I-4	SPECIFICATION	1
II	FUNCTIONAL DESCRIPTION	
2-1	INTRODUCTION	3
2-2	DATA OUT BUS	3
2-3	DATA IN BUS	3
2-4	A $\emptyset$ -A7	3
2-5	I/ $\emptyset$ CONTROL LINES	4
2-6	ADDRESS DECODER	4
2-7	OUTPUT PORT 68H	4
2-8	OUTPUT PORT 69H	4
2-9	OUTPUT PORT 6AH	4
2-10	INPUT PORT 68H	5
III	CONTROL SOFTWARE	
3-1	INTRODUCTION	6
3-2	2758/2716/2732 PROGRAMMING	6
3-3	2708 PROGRAMMING	7
3-4	LOAD SEQUENCE	9
3-5	READ SEQUENCE	9
3-6	PROGRAMMING SEQUENCE	9
3-7	CAUTION:	10
3-8	USING THE PROM PROGRAMMER	10
3-9	LOADING AN OBJECT FILE	11
3-10	READING A PROM	12
3-11	PROGRAMMING A PROM	12
3-12	COPYING A PROM	14
IV	CONSTRUCTION	
4-1	INTRODUCTION	15
4-2	ASSEMBLY PROCEDURE	15
4-3	VOLTAGE CHECK	16
V	SWITCH SELCTION	
5-1	INTRODUCTION	18
VI	CHECK OUT	
6-1	INTRODUCTION	19
6-2	I/O PORT VERIFICATION	19
VII	SOFTWARE REQUIREMENTS	
7-1	INTRODUCTION	20
	APPENDICES	
APPENDIX A	SCHEMATIC	
APPENDIX B	PART LIST	
APPENDIX C	ASSEMBLY DRAWING	
APPENDIX D	SOFTWARE LISTING	

## SECTION I

### 1-1 INTRODUCTION

The SD SYSTEMS Prom Programmer Board (PROM-100) provides a low cost means for programming read only memory capability for computers utilizing the S-100 bus structure.

### 1-2 GENERAL DESCRIPTION

The SD SYSTEMS PROM-100 board is a high performance circuit capable of programming the industry standard 2708 1K X 8 EPROM, Intel's 2758 1K X 8 EPROM, 2716 2K X 8 EPROM, 2732 4K X 8 EPROM and Texas Instruments 2516 2K X 8 EPROM. The PROM-100 is selected for the appropriate EPROM type through dip switches S1 and S2. Another notable feature of the PROM-100 is the 25V programming pulse generated on board, eliminating the need for an external 25V supply.

### 1-3 PHYSICAL

The SD SYSTEMS Prom Programmer is implemented on a single 7.00" X 10.0" X .065" printed circuit board and is interfaced to the System by connector J-1.

### 1-4 SPECIFICATION

Table 1-1 lists the overall specifications for the SD SYSTEMS Prom Programmer Board. Table 1-2 lists the pin usages of connector J-1 for the Prom Programmer Board.

TABLE 1-1

+8 to +10V	300 ma maximum
+16 to 18V	100 ma maximum
-16 to 18V	60 ma maximum
Operating temperature	0°C to 50°C
Interface levels	TTL Compatible
Programming time	100 seconds max for 16,389 bits

TABLE 1-2

PIN NO	SIGNAL NAME	DIRECTION	DESCRIPTION
1, 51	+8V to +10V	INPUT	POWER
2	+16V to +20V	INPUT	POWER
100, 50	GND		
79, 80, 81, 31, 30 29, 82, 83	A0-A7	INPUT	ADDRESS BUS
36, 35, 88, 89, 38 39, 40, 90	D0-0 to D0-7	INPUT	DATA BUS OUT
95, 94, 41, 42, 91, 92, 93, 43	DI-0 to DI-7	OUTPUT	DATA BUS IN
77	PWR		WRITE
78	PDBIN		DATA BUS IN
46	SINP		PORT INPUT CYCLE
45	SOUT	INPUT	PORT OUTPUT CYCLE
99	POC	INPUT	POWER ON CLEAR

## SECTION II

### FUNCTIONAL DESCRIPTION

#### 2-1 INTRODUCTION

Functionally, the PROM Programmer consist of two main portions; hardware, and the software. The hardware allows the computer to select the port bits, generate the programming pulse, and control the decoder. The software must direct the hardware in each of the operations described above. The major functions contained in the PROM Programmer hardware are shown in the block diagram (Figure 2-1).

#### 2-2 DATA OUT BUS

The 8-Bit Data Out Bus is the S-100 path for transferring data from the computer (CPU) to the output ports on the Prom Programmer board.

#### 2-3 DATA IN BUS

The 8-Bit Data In Bus is the S-100 path for transferring data from the input port on the Prom Programmer Board to the computer (CPU).

#### 2-4 A0-A7

The A0-A7 low order eight address lines are used by the computer (CPU) to select the various input/output ports through a decoder on the Prom Programmer Board.

## 2-5 I/O CONTROL LINES AND READ WRITE CONTROL

The I/O Control Lines consist of PWR, PD BIN, SINP and SOUT. These lines are used to control the input and output operations from/to the I/O ports on the board.

## 2-6 ADDRESS DECODER

The Address Decoder detects when a port address used by the Prom Programmer is present on (A0-A7). The output of the decoder is used to gate read and write pulses to the I/O ports.

## 2-7 OUTPUT PORT 68H

Output port 68H is an 8 bit data register that transfers the data from CPU to ROM.

## 2-8 OUTPUT PORT 69H

Output port 69H is an 8 bit address register that determines the location in the ROM for the data to be programmed or read.

## 2-9 OUTPUT PORT 6AH

Output port 6AH is an 8 bit control register with several control functions.

1. Bit 1 & 2 Two high order address bits A8 & A9 address inputs.
2. Bit 3 One high order bit A10 (address input for 2716).
3. Bit 4 One high order bit A11 (address input for 2732).
4. Bit 6 Program pulse for 2716/58.
5. Bit 7 Drive MOS clock drive.
6. Bit 8 Controls port 68H chip select for 2716/58/08.

2-10 INPUT PORT 68 H

Input port 68H is used to read the present data in the ROM.



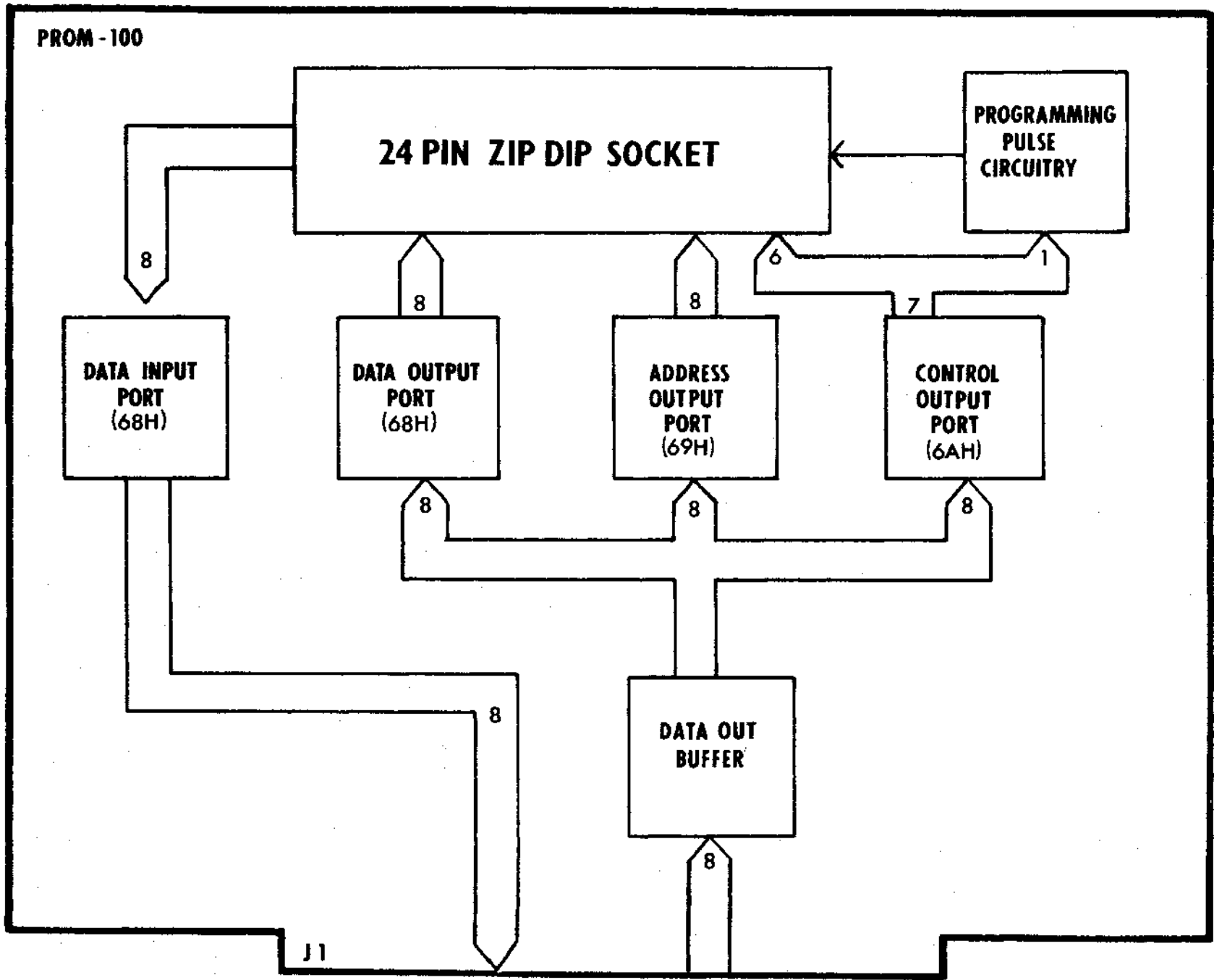


FIGURE 2-1

SECTION III  
CONTROL SOFTWARE

3-1 INTRODUCTION

Certain sequences must be executed to ensure proper programming of the prom.

3-2 2758/16/32 PROGRAMMING

Initially, and after each erasure, all bits of the 2716 are in the "high" state. Data is introduced by selectively programming "0's" into the required locations. Although only "0's" will be programmed both "1's" and 0's" can be presented in the data word. The only way to change a "0" to a "1" is by ultraviolet light erasure.

The 2716 is in the programming mode when the VPP power supply is 25V and  $\overline{CS}$  is at a low TTL level. The data to be programmed is applied 8 bits in parallel to the data output port. The levels required for the address and data inputs are TTL.

When the addresses and data are stable a 50 MSEC, active high, TTL program pulse is applied to the appropriate input pin on the EPROM.

A program pulse must be applied to each address location to be programmed. The user can program any location at any time - either individually, sequentially, or at random. The program pulse has a maximum width of 55 MSEC.

A verify is performed on the program bits to determine that they are correctly programmed. The verify may be performed with VPP at 25V.

### 3-3 2708 PROGRAMMING

Initially and after each erasure all 8192 bits are in high state (output high). Information is introduced by selectively programming "0" into the desired bit location. A programmed "0" can be changed to a "1" by UV. erasure.

The circuit is set up for programming operation by raising the CS/WE input (pin 20) to +12V. The word address is selected in the same manner as in the read mode. Data to be programmed is presented 8 bits in parallel to the data output lines (D0<sub>0</sub>-D0<sub>7</sub>) Logic levels for address and data lines and the supply voltages are the same as for the read mode.

After address and data set up, one program pulse per address is applied to the program input pin 18. One pass through an address is defined as a program loop. The number of loops (N) required is a function of the program pulse width (tpw) allowing to  $(N) \times (tpw) \geq 100$ . tpw in MSEC.

The width of the program pulse is from 0.1 to 1ms. The number of loops (N) is from a minimum of 100 (tpw=1ms) to greater than 1000 (tpw=0.1). There must be N successive loops through all 1024 addresses. It is not permitted to apply N program pulses to an address and then change to the next address to be programmed.

\*The PROM Programmer writes data received from the host computer into the PROM as guided by the address/control outputs supplied by the driver software. Basically, the PROM Programmer manipulates the data and the address/control inputs from the computer so that the subject PROM receives proper data, address and control inputs. The PROM-100 generates the programming pulse for the 2708 from the output of a step up switching voltage regulator provided on the board. The same pulse is used to provide VPP (25V) at pin 21 of 2716/58. This 25V is set up by the driver software of the PROM Programmer.

Driver software for PROM Programmer must meet three requirements. These are: loading the object modules (diskettes) into host computer memory, reading a PROM into memory and programming a PROM. Within the software, the executive system coordinates all three operations while each operation sequence is run by its respective routine. In regards to the PROM Programmer, only the read and program sequences are significant because they involve specific address/control inputs to the PROM Programmer. The requirements imply that, in addition to allocating a memory buffer for the driver software, the host computer must also supply a 1, 2 or 4K byte memory buffer for the data module to be loaded, read, and programmed.

### 3-4 LOAD SEQUENCE

The load sequence must be able to load data into memory at any location regardless of the routine load address contained in the file. Once the number of bytes to be loaded has been specified, the user may load and program a portion of the file and then return for the rest. For example, if a file to be programmed contains 4K bytes, the user could program 2K bytes into one 2716 PROM and then program the remaining 2K bytes of the module into a second PROM.

### 3-5 READ SEQUENCE

The read sequence is necessary in order to perform a PROM copy function. Since the PROM programmer contains only one 24 pin ZIP DIP socket, the master PROM must first be read into the 1K, 2K or 4K memory buffer (2708, 58 or 2716/32) and then programmed into an erased PROM. The host computer should offer memory editing to allow modification of the memory buffer before programming the new PROM.

### 3-6 PROGRAMMING SEQUENCE

The programming sequence actually involves a three step process.

1. Verify that PROM is erased.
2. Program the PROM.
3. Verify that the PROM contains correct data.

The entire PROM is read to verify erasure, that each location contains FF<sub>H</sub>. If unerased locations are found, the details are output to the console.

The programming of the PROM involves outputting the data for each of the 1K, 2K or 4K locations sequentially along with the PROM address and the 50ms program pulse for 2758/16/32 or 1ms program pulse for 2708. After this programming attempt is complete, the entire PROM must be read and compared with the memory buffer. If any locations do not match, the address and data should be output to the console.

### 3-7 CAUTION:

In order to prevent possible destruction of PROMS, PROMS must never be inserted into the ZIP DIP socket on the PROM Programmer until the driver software is running. If the PROM is inserted prior to starting the driver software, the PROM may be subjected to a continuous program pulse which will destroy the PROM. For the same reason, the host computer must not be reset while the PROM is in the socket.

### 3-8 USING THE PROM PROGRAMMER

The driver software can be used to load object files from diskettes, read object code from PROMS and program object code into PROMS. As the driver software starts, the console prints the start/options message. The sign-on portion of the message indicates that the PROM Programmer software is operative. The options portion consist of the first of three questions about which operations are

to be performed (load a file, read a PROM, and program a PROM), in that order. The user answers the questions by entering a Y (for Yes) or N (for No) at the end of each question. If a Y is entered the respective operation is performed after the necessary parameters are entered.

### 3-9 LOADING AN OBJECT FILE DISK FILE ONLY

Prior to answering the first question: READY TO LOAD A FILE (Y/N)? the user sets up the object file by specifying the file name, when loading the PROM-100 Software: A> PROM08 filename.Ext. When the user answers the question with a Y, the PROM Programmer software asks the user to enter two parameters (load start address and load size) by printing HEX LOAD address, #Bytes: The parameters must be separated by a comma or space and terminated with a carriage return. If no parameters are entered in front of the carriage return, the object file will be loaded into the RAM locations specified on the module. However, since the PROM Programmer driver software occupies the memory location  $\text{0100}_H$  through  $\text{5FF}_H$  the object file must not overlay these locations. This condition can be prevented by using the load address parameter which allows relocating the object data to any area of memory. The number of bytes parameter allows the loading of a limited number of bytes of the file into RAM so that object file may be loaded and programmed in blocks. If the load size parameter is not entered, the entire module is loaded. If a period is entered before the end of the parameter, the software repeats

the request for parameters allowing the user to correct the parameter. When the load is complete the PROM Programmer software asks: READY TO PROGRAM A PROM (Y/N)?

### 3-10 READING A PROM

In the console printout - READY TO READ A PROM (Y/N)? If a Y is entered, the PROM Programmer software asks the user to enter three parameters (MEM start address, MEM end address, PROM start address) by printing MEMORY START, MEMORY END, PROM START:

The MEM start and END parameters specify the memory space into which the data is to be read. The PROM START defines the first PROM location to be read. The three parameters must be separated by a comma or space and terminated with a carriage return. If a period is entered before the end of the parameter the software repeats the request for parameters allowing the user to correct the parameters. When the read is complete or when the user answers the question: READY TO READ A PROM (Y/N)? With an N, the PROM Programmer software asks: READY TO PROGRAM A PROM (Y/N)?

### 3-11 PROGRAMMING A PROM

When the user answers the question: READY TO PROGRAM A PROM (Y/N)? with a Y, the programmer software asks the user to enter three parameters MEM start address, MEM end address and PROM start address by printing MEMORY START, MEMORY END, PROM START: The MEMORY START and END parameters specify the memory space from which the data is to be programmed. The three parameters must be separated by a comma or space and terminated with a carriage return.



If a period is entered before the end of the parameters, the software repeats its request for parameters allowing the user to correct the parameters.

The PROM Programmer software first verifies that the PROM is erased (contains FF<sub>H</sub> in all locations). If the PROM Programmer software finds unerased locations, it prints the first unerased location. To check for other unerased locations the user enters a carriage return after each unerased location printed. If the PROM contains unerased location, the PROM must be erased again before starting programming. If there are unerased locations, the user may attempt programming by entering C. If there are no unerased locations, programming begins automatically. During the programming sequence, the PROGRAMMING LED COMES ON and stays on till the program ends. To stop programming during the sequence the user enters a period (.). The PROM Programmer software asks DO YOU WISH TO LOAD A MODULE (Y/N)? when the programming sequence is complete, the PROM Programmer reads the PROM to verify that the data has been programmed correctly. If the PROM Programmer software finds locations with BAD data, it prints the bad locations. To stop the error listing the user enters a period and the software returns to the start asking: READY TO LOAD A FILE (Y/N)? If the PROM contains bad locations it should be re-erased and re-programmed until it is correctly programmed. When programming is done the user can exit the PROM Programmer software by typing a period (.).

### 3-12 COPYING A PROM

Copying a master PROM into a subject PROM requires only that the master PROM be read as described in paragraph 3-10 and the subject PROM be programmed as described in paragraph 3-11.

SECTION IV  
CONSTRUCTION

4-1 INTRODUCTION

The SD SYSTEMS PROM Programmer board is intended for those people who have had some prior experience with kit building and digital electronics. If you do not fall into this category it is highly recommended that you find an experienced person to help you in assembly and checkout of the board. Appendix B shows the parts list for the SD SYSTEMS PROM Programmer board. Double check all parts against this parts list. If any differences are noted please contact SD SYSTEMS.

NOTE: General construction information assembly diagram and schematic diagram can be found in the appendices.

4-2 ASSEMBLY PROCEDURE (SEE APPENDIX C)

1. Install and solder the IC sockets in their proper locations.

- 8 Pin at U14
- 14 Pin at U2,U3,U5,U9,U10,U11
- 16 Pin at U4
- 20 Pin at U6,U7,U8,U12,U13

NOTE: DO NOT INSTALL SOCKETS AT LOCATIONS S1 and S2.

2. Install and solder the resistors as follows:

- A. R8            220 Ohm  $\frac{1}{2}$ W 5% (Red,Red,Brown)
- B. R4            22.1K Ohm 1/8W 1% (Precision Resistor)
- C. R5            1 Ohm  $\frac{1}{4}$ W 5% (Brown,Black,Gold)
- D. R6            270 Ohm  $\frac{1}{4}$ W 5% (Red,Violet,Brown)
- E. R7            100 Ohm  $\frac{1}{4}$ W 5% (Brown,Black,Brown)
- F. F1,2,3       1K Ohm  $\frac{1}{4}$ W 5% (Brown,Black,Red)

3. Install and solder diodes CR1 with banded end as shown on the PC board.
  - A. CR1 1N5803
  - B. CR2 Zener Diode 1N751 -5V
4. Install and solder the capacitors as follows:
  - A. C1-C3,C6-C11,C14,C15 0.1MF Ceramic
  - B. C12,C16-C18, 10MF 20V Tantalum
  - C. C13 .002MF Ceramic
  - D. C4 500MF 35V Dielectric Axial Leads
  - E. C5 300pf
5. Install and solder the voltage regulators with the heatsinks 6106-14 using the #6-32 hardware supplies.  
  
VR1 =5v 7805 or LM340T-5  
VR2 +12V 7812 or LM340T-12
6. Install and solder L1 coil.  
  
L1 J12044 500 Micro Henry
7. Install and solder dip switches S1 and S2.
8. Double check all solder connections for cold solder joints, unsoldered connections or shorted connections.

#### 4-3 VOLTAGE CHECK

1. Install the board in the computer and measure the output of +5V regulator, +12V regulator and -5V of CR1 respectively.
  - A. VR1 = +5V (Top Pin)
  - B. VR2 = +12V (Bottom Pin)
  - C. CR2 = -5V (Anode)

NOTE: Do not proceed with board checkout until all power supply voltages are correct. The TTL and MOS logic can be permanently damaged if improper voltages are applied.

2. Install the IC's in their sockets observing the Pin 1 designation on each socket marked on the PC Board.

A.	U1	24 Pin Zip Dip socket (Pin 1 lever in lower left.)
B.	U2	TL497ACN
C.	U3	74LS30
D.	U4	74LS155
E.	U6, U8	74LS273
F.	U5, U10	74LS14
G.	U7	74LS373
H.	U9	7406
I.	U11	74LS08
J.	U12, 13	74LS244
K.	U14	DS0025CN

3. Double check all IC's for proper orientation and location.
4. Refer the switch selection option and position then as described in Section V.

SECTION V  
SWITCH SELECTION

5-1 INTRODUCTION

PROM Programmer has been designed to accomodate a variety of PROMS. Care must be taken to select the right positions of the switches for the users specific PROM selection. Table 5-1 lists the proper switch positions for each PROM type.

PROM TYPE	S1	S2
2708 ( ) 1K x 8	All other	2708
2758 (5V)	All other	All other
2716 (5V)	All other	All other
2732 (5V)	2732	All other

Table 5-1

## SECTION VI

### CHECK OUT

#### 6-1 INTRODUCTION

This section will describe some basic checks that should be made on the PROM Programmer.

NOTE: It is assumed at this point that the voltage checks described in Section 4 have been previously made. The following checks require that the CPU board also be plugged into the bus.

#### 6-2 I/O PORT WRITE/READ VERIFICATION

Using the monitor in the system or a short program. Write data to output port 68 and read it back. Verify that the data read back is the same as that was written. This can be done as follows if using the monitor.

```
.0 6A 80 (cr) - enable Port 68
.0 68 dd (cr) - output data (dd) to port 68
.I 68 (cr) - input from port 68
68 dd - Data Read should be identical to data output
```

SECTION VII  
SOFTWARE REQUIREMENTS

6-1 INTRODUCTION

The SD Systems PROM-100 Programming Board software is available on disk on Intel 2716 EPROM.

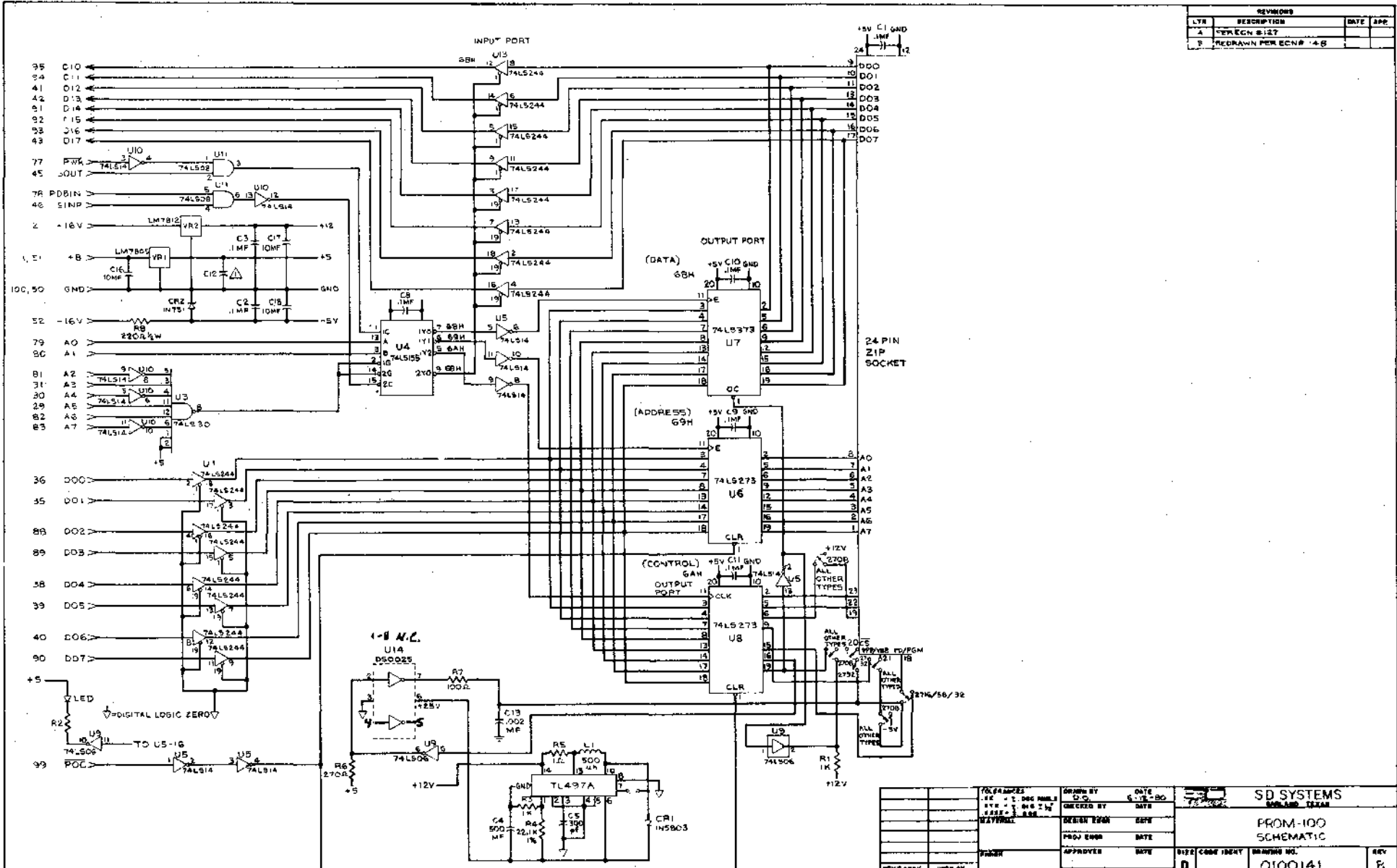
A listing of the software in EPROM is included in Appendix D, while the disk version is included on the disk. (.PRN extension)

The program in PROM must be moved to 100H prior to execution. PROM 08, the 2708 programming utility, is located in the first 1K section of the PROM (0-3FFH). PROM 16, the Intel 2758, 2716, 2732 programming utility is located in the latter 1K section of the PROM (400<sub>H</sub>-7FFH).



APPENDIX A  
SCHEMATIC

REVISIONS			
LYR	DESCRIPTION	DATE	APP
1	FOR ECN# 8-17		
2	REDRAWN PER ECN# 4-8		



NOTES: CAPACITORS C6, C7, C14 / C15 ARE .1 MF.

TOLERANCES		DRAWN BY	DATE	SD SYSTEMS SULLY, TEXAS
.1% - 1.0% RNDL		D.C.	6-18-80	
.1% - .5% 2% .5% - 1.0%		CHECKED BY	DATE	PROM-100 SCHEMATIC
MATERIAL		DESIGN ENGR	DATE	
FINISH		APPROVER	DATE	SIZE CODE IDENT
NEXTassy		USED ON		
APPLICATION				DRAWING NO. 0100141
				REV B

**APPENDIX B**

**PART LIST**

# SD Systems

P.O. Box 28810 • Dallas, Texas 75228 214-271-4667

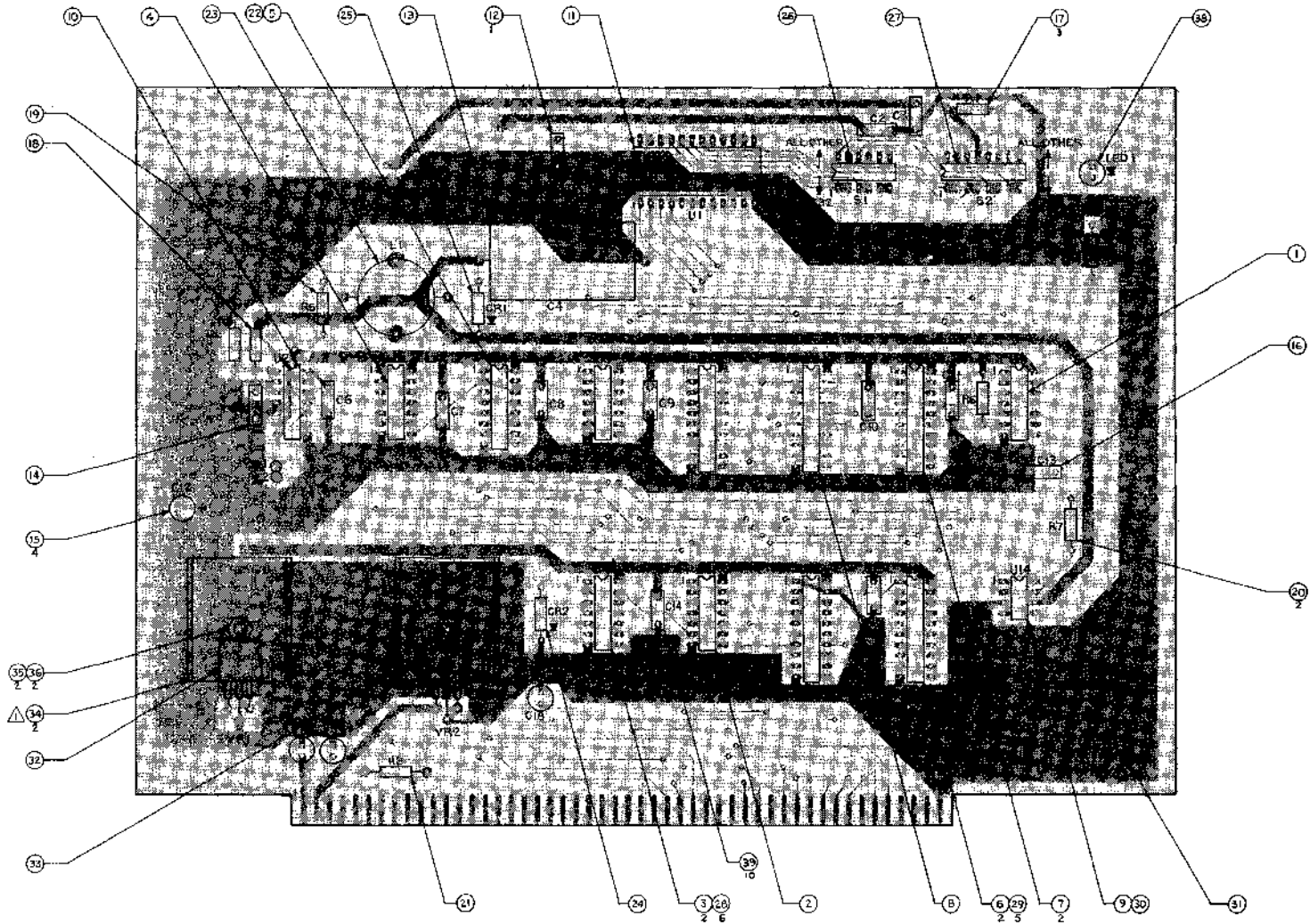
## BILL OF MATERIALS

Title:			PL No.	Rev.	
PROM-100			0100140	B	
Date Released:		Approved:		Sheet	
5/14/79		JRA ROB [Signature] CLP		1 of 2	
Item No	Qty	SD-P/N	Description	Unit Cost	Extension
1	1	7010007	7406, U9		
2	1	7010166	74LS08, U11		
3	2	7010172	74LS14, U5, U10		
4	1	7010180	74LS30, U3		
5	1	7010226	74LS155, U4		
6	2	7010264	74LS244, U12, U13		
7	2	7010276	74LS273, U6, U8		
8	1	7010304	74LS373, U7		
	1	7010353	DS0025CN, U14		
10	1	7010354	TL497ACN, U2		
11	1	7060010	24 PIN ZIP DIP, U1		
12	1	7030007	.1 MF C11		
13	1	7030022	500MF 35V, C4		
14	1	7030006	300 pf, C5		
15	4	7030009	10 MF 20V, C12, C16-18		
16	1	7030034	.002 MF Ceramic, C13		
17	3	7020073	1K Ohm 1/4W 5%, R1-R3		
18	1	7020175	22.1K Ohm, 1%, R4		
19	1	7020001	1 Ohm 1/4W 5%, R5		
20	1	7020059	270 Ohm 1/4W 5%, R6		
21	1	7020177	220 Ohm 1/2W, 5%, R8		
	1	7060003	16 Pin Socket, U4		
23	1	7120001	L1 Inductor 500 uh		
24	1	7040003	1N751, CR2		



APPENDIX C  
PROM-100  
ASSEMBLY DRAWING

REVISIONS			
LTR	DESCRIPTION	DATE	APP.



NOTES: 2. DEGREASE BOARD.  
 △ APPLY THERMAL COMPOUND P.N. 12-8 TO VR1 & 2.

TOLERANCES UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE TO CENTER UNLESS OTHERWISE SPECIFIED FRACTIONS TO 1/16"		DRAWN BY E.C.	DATE 5-8-80	SD SYSTEMS DALLAS, TEXAS	
MATERIAL		CHECKED BY J.P.	DATE	PROM-100	
FINISH		DESIGN ENGR J.P.	DATE	ASSEMBLY DRAWING	
NEXT ASSY USED ON		PROJ ENGR J.P.	DATE	SIZE CODE IDENT D	DRAWING NO. 0100140
APPLICATION		APPROVED	DATE	SCALE 1" = 1"	REVISIONS 1 OF 3

**APPENDIX D**  
**SOFTWARE LISTINGS**



STMT SOURCE STATEMENT

```

0001      NAME      PROM08
0002 ;
0003 ;
0004 ; 2708 PROM PROGRAMMER CONTROL PROGRAM
0005 ; SDOS ENVIRONMENT
0006 ; 9/27/79
0007 ;
0008 ; VERSION 1.3          CP/M COMPATIBLE          8/26/80
0009 ;
0010 ;
0011 ; SYSTEM EQUATES
0012 ;
0013 ;
>0000    0014 EXIT      EQU      0H
>0100    0015 TBASE     EQU      100H      ; BASE OF TRANSIENT AREA
>0068    0016 DATA     EQU      68H      ; DATA PORT
>0069    0017 LADD      EQU      69H      ; LOW ADDRESS PORT
>006A    0018 HACTL     EQU      6AH      ; HIGH ADDRESS + CONTROL PORT
>0000    0019 RBOUT     EQU      0
>0005    0020 FDOS      EQU      5
>005C    0021 FCB       EQU      5CH
>0080    0022 BUFFER    EQU      80H
>000F    0023 OPEN      EQU      15
>0010    0024 CLOSE     EQU      16
>0014    0025 READ      EQU      20
>001A    0026 SETDMA    EQU      26
>0080    0027 NBYTES    EQU      128
>0064    0028 LOOPS     EQU      100
0029 ;
0030 ;
0031 ;
0032 ;
0033 ;
0034 ;
0035      PSECT      ABS
0036 ;
>0100    0037      ORG      TBASE      ; START AT 100H
0038 ;
0039 ;
0040 ;
0041 ; PROGRAM PROM ENTRY POINT
0042 ;
0043 ;
0044 ;
0100    319607    0045 PPG      LD      SP,SPVAL
0103    218602    0046      LD      HL,MSG1
0106    CDA005    0047      CALL   PTXT      ; "PROM08 VERSION 1.3"
0109    1E80      0048      LD      E,80H
010B    CD4D02    0049      CALL   CONIN
010E    FE31      0050      CP      31H
0110    3806      0051      JR      C,SETDLY
0112    1EBF      0052      LD      E,0BFH
0114    2802      0053      JR      Z,SETDLY
0116    1E00      0054      LD      E,0
0118    7B        0055 SETDLY: LD      A,E
0119    328C06    0056      LD      (DLAY),A
011C    3E80      0057      LD      A,NBYTES
011E    329306    0058      LD      (BCOUNT),A

```

ADDR	CODE	STMT	SOURCE	STATEMENT
0121	AF	0059		XOR A
0122	329206	0060		LD (FOPEN),A
0125	319607	0061	PPG0	LD SP,SPVAL
0128	AF	0062		XOR A ; CLEAR ACC
0129	D36A	0063		OUT (HACTL),A
012B	CD9105	0064		CALL CRLF
012E	21EB02	0065		LD HL,MSG2
0131	CDA005	0066		CALL PTXT ; "READY TO LOAD A FILE ?"
0134	210000	0067		LD HL,0
0137	228D06	0068		LD (SCR1),HL ;CLEAR BYTE COUNTER
013A	CD2D02	0069		CALL YORN
013D	D2FD03	0070		JP NC,READ2 ; IF NOT,CHK READ
0140	CD9105	0071	PPG0B	CALL CRLF
0143	3A9206	0072		LD A,(FOPEN)
0146	B7	0073		OR A
0147	2015	0074		JR NZ,PPG0C-\$
0149	0E0F	0075		LD C,OPEN
014B	115C00	0076		LD DE,FCB
014E	CD0500	0077		CALL FDOS
0151	FEFF	0078		CP 255
0153	2009	0079		JR NZ,PPG0C-\$
0155	218303	0080		LD HL,MSG10
0158	CDA005	0081		CALL PTXT
015B	C36602	0082		JP RENTRY
015E	0E1A	0083	PPG0C	LD C,SETDMA
0160	118000	0084		LD DE,BUFFER
0163	CD0500	0085		CALL FDOS
0166	210703	0086		LD HL,MSG3
0169	CDA005	0087		CALL PTXT ; "HEX LOAD ADDRESS, # BYTES
016C	CD1806	0088		CALL SCAN ;COLLECT DATA
016F	3A9106	0089		LD A,(NXCHR)
0172	FE2E	0090		CP ' '
0174	28CA	0091		JR Z,PPG0B-\$
0176	21B603	0092		LD HL,MSG12
0179	CDA005	0093		CALL PTXT
017C	2A9606	0094		LD HL,(OPR1)
017F	CD7E02	0095		CALL PADDR
0182	CD9105	0096		CALL CRLF
0185	CD0D02	0097		CALL THDR ; READ ADDR/#BYTES 1ST LINE
0188	E5	0098		PUSH HL
0189	ED5B9606	0099		LD DE,(OPR1) ; LOAD ADDRESS ENTERED
018D	3A9006	0100		LD A,(OPCNT)
0190	A7	0101		AND A
0191	2804	0102		JR Z,PPG0A-\$ ; IF NO OPR'S JUMP
0193	ED52	0103		SBC HL,DE ; COMPUTE OFFSET
0195	E5	0104		PUSH HL
0196	D1	0105		POP DE ; OFFSET REG
0197	E1	0106	PPG0A	POP HL
0198	ED538F06	0107		LD (SCR3),DE ; SAVE OFFSET
019C	1803	0108		JR PPG2-\$
019E	CD0D02	0109	PPG1	CALL THDR ; LOAD ADDR/# BYTES
01A1	ED5B8F06	0110	PPG2	LD DE,(SCR3)
01A5	B7	0111		OR A ; CLEAR CARRY
01A6	ED52	0112		SBC HL,DE ; SUBTRACT OFFSET
01A8	CD6405	0113	PPG3	CALL TWDG ; READ A BYTE
01AB	77	0114		LD (HL),A ; STORE IT
01AC	229406	0115		LD (LAST),HL
01AF	23	0116		INC HL

ADDR	CODE	STMT	SOURCE	STATEMENT
01B0	E5	0117		PUSH HL
01B1	2A8D06	0118		LD HL,(SCR1) ; BYTE COUNTER
01B4	23	0119		INC HL
01B5	228D06	0120		LD (SCR1),HL
01B8	E1	0121		POP HL
01B9	10ED	0122		DJNZ PPG3-\$ ; READ ENTIRE LINE
01BB	CD6405	0123		CALL TWDG ; READ CHECKSUM
01BE	AF	0124		XOR A
01BF	81	0125		ADD A,C
01C0	280B	0126		JR Z,PPG4-\$
01C2	E5	0127		PUSH HL
01C3	21CB03	0128		LD HL,MSG13
01C6	CDA005	0129		CALL PTXT
01C9	CD7E02	0130		CALL PADDR ; CHECKSUM ERROR
01CC	E1	0131		POP HL
01CD	2A8D06	0132	PPG4	LD HL,(SCR1)
01D0	ED4B9806	0133		LD BC,(OPR2) ; # OF BYTES TO READ
01D4	2B	0134		DEC HL ; IF 0,THEN FFFF
01D5	0B	0135		DEC BC
01D6	A7	0136		AND A
01D7	ED42	0137		SBC HL,BC
01D9	DA9E01	0138		JP C,PPG1 ; IF MORE,JUMP
01DC	21E803	0139	PPG10	LD HL,MSG14
01DF	CDA005	0140		CALL PTXT
01E2	2A9406	0141		LD HL,(LAST)
01E5	CD7E02	0142		CALL PADDR
01E8	CD9105	0143	PPG10A	CALL CRLF
01EB	212303	0144		LD HL,MSG4
01EE	CDA005	0145		CALL PTXT ; "READY TO PROG A PROM ?"
01F1	CD2D02	0146		CALL YORN
01F4	D26602	0147		JP NC,RETRY
01F7	CD9105	0148	PPG11	CALL CRLF
01FA	214203	0149		LD HL,MSG5
01FD	CDA005	0150		CALL PTXT ; "MEM START,MEM END,PROM START
0200	CD1806	0151		CALL SCAN
0203	3A9106	0152		LD A,(NXCHR)
0206	FE2E	0153		CP '.' ; IF '.' EXIT
0208	28ED	0154		JR Z,PPG11-\$
020A	C34C04	0155		JP PROG
		0156		;
		0157		;
		0158		;
020D	CD5A06	0159	THDR	CALL FREAD ; LOOK FOR ':'
0210	FE3A	0160		CP ':'
0212	20F9	0161		JR NZ,THDR-\$
0214	AF	0162		XOR A ; CLEAR CHECKSUM
0215	4F	0163		LD C,A
0216	CD6405	0164		CALL TWDG
0219	A7	0165		AND A ; IS RECORD LENGTH 0?
021A	280D	0166		JR Z,THDR1-\$
021C	47	0167		LD B,A
021D	CD6405	0168		CALL TWDG ; GET FIRST BYTE OF ADDRESS
0220	67	0169		LD H,A
0221	CD6405	0170		CALL TWDG ; GET SECOND BYTE
0224	6F	0171		LD L,A
0225	CD6405	0172		CALL TWDG ; GET RECORD TYPE
0228	C9	0173		RET
0229	F1	0174	THDR1	POP AF

ADDR	CODE	STMT	SOURCE	STATEMENT
022A	C3DC01	0175		JP PPG10
		0176		;
		0177		;
022D	CD4D02	0178	YORN	CALL ECHO
0230	4F	0179		LD C,A
0231	FE2E	0180		CP '.'
0233	CA6602	0181		JP Z,REENTRY
0236	B7	0182		OR A
0237	FE4E	0183		CP 'N'
0239	C8	0184		RET Z
023A	FE59	0185		CP 'Y'
023C	20EF	0186		JR NZ,YORN-\$
023E	37	0187		SCF
023F	C9	0188		RET
		0189		;
		0190		;
0240	C5	0191	CONST	PUSH BC ; CALL SDOS CONSOLE STATUS
0241	D5	0192		PUSH DE
0242	E5	0193		PUSH HL
0243	0E0B	0194		LD C,0BH
0245	CD0500	0195		CALL 5
0248	B7	0196		OR A
0249	E1	0197		POP HL
024A	D1	0198		POP DE
024B	C1	0199		POP BC
024C	C9	0200		RET
		0201		;
>024D		0202	ECHO:	
024D	C5	0203	CONIN	PUSH BC ; CALL SDOS CONSOLE INPUT
024E	D5	0204		PUSH DE
024F	E5	0205		PUSH HL
0250	0E01	0206		LD C,1
0252	CD0500	0207		CALL 5
0255	E1	0208		POP HL
0256	D1	0209		POP DE
0257	C1	0210		POP BC
0258	C9	0211		RET
		0212		;
0259	D5	0213	CONOUT:	PUSH DE
025A	C5	0214		PUSH BC
025B	E5	0215		PUSH HL
025C	59	0216		LD E,C
025D	0E02	0217		LD C,02
025F	CD0500	0218		CALL 5
0262	E1	0219		POP HL
0263	C1	0220		POP BC
0264	D1	0221		POP DE
0265	C9	0222		RET
		0223		;
0266	AF	0224	REENTRY	XOR A
0267	D36A	0225		OUT (HACTL),A
0269	CD9105	0226		CALL CRLF
026C	3A9206	0227		LD A,(FOPEN)
026F	B7	0228		OR A
0270	CA0000	0229		JP Z,EXIT
0273	0E10	0230		LD C,CLOSE
0275	115C00	0231		LD DE,FCB
0278	CD0500	0232		CALL FDOS

ADDR	CODE	STMT	SOURCE	STATEMENT
027B	C30000	0233		JP EXIT
		0234	;	
		0235	;	
027E	7C	0236	PADDR	LD A,H
027F	CDAC05	0237		CALL PACC
0282	7D	0238		LD A,L
0283	C38505	0239		JP PASP
		0240	;	
		0241	;	
		0242	;	
0286	50524F4D 2D303820 56455253 494F4E20 312E33	0243	MSG1	DEFM 'PROM-08 VERSION 1.3'
0299	0D0A0D0A	0244		DEFW 0A0DH,0A0DH
029D	49532054 48495320 53595354 454D2052 554E4E49 4E472032 2E302C20 322E352C 204F5220 342E3020 4D485A3F	0245		DEFM 'IS THIS SYSTEM RUNNING 2.0, 2.5, OR 4.0'
02C9	0D0A	0246		DEFW 0A0DH
02CB	28322E30 202D2030 202C2032 2E35202D 2031202C 20342E30 202D2032 29	0247		DEFM '(2.0 - 0 , 2.5 - 1 , 4.0 - 2)'
02E8	0A0D	0248		DEFW 0D0AH
02EA	03	0249		DEFB 03H
02EB	52454144 5920544F 204C4F41 44204120 46494C45 2028592F 4E293F	0250	MSG2	DEFM 'READY TO LOAD A FILE (Y/N)?'
0306	03	0251		DEFB 03H
0307	48455820 4C4F4144 20414444 52455353 2C202320 42595445 533A20	0252	MSG3	DEFM 'HEX LOAD ADDRESS, # BYTES: '
0322	03	0253		DEFB 03H
0323	52454144 5920544F 2050524F 4752414D	0254	MSG4	DEFM 'READY TO PROGRAM A PROM (Y/N)?'

ADDR	CODE	STMT	SOURCE	STATEMENT
	20412050			
	524F4D20			
	28592F4E			
	293F			
0341	03	0255		DEFB 03H
0342	4D454D4F	0256	MSG5	DEFM 'MEMORY START, MEMORY END, PROM START: '
	52592053			
	54415254			
	2C4D454D			
	4F525920			
	454E442C			
	50524F4D			
	20535441			
	52543A20			
0366	03	0257		DEFB 03H
0367	52454144	0258	MSG6	DEFM 'READY TO READ A PROM (Y/N)?'
	5920544F			
	20524541			
	44204120			
	50524F4D			
	2028592F			
	4E293F			
0382	03	0259		DEFB 03H
0383	46494C45	0260	MSG10	DEFM 'FILE NOT FOUND'
	204E4F54			
	20464F55			
	4E44			
0391	03	0261		DEFB 03H
0392	41545445	0262	MSG11	DEFM 'ATTEMPTING TO READ PAST END OF FILE'
	4D505449			
	4E472054			
	4F205245			
	41442050			
	41535420			
	454E4420			
	4F462046			
	494C45			
03B5	03	0263		DEFB 03H
03B6	4C4F4144	0264	MSG12	DEFM 'LOAD START ADDRESS= '
	20535441			
	52542041			
	44445245			
	53533D20			
03CA	03	0265		DEFB 03H
03CB	43484543	0266	MSG13	DEFM 'CHECKSUM ERROR AT LOCATIONS '
	4B53554D			
	20455252			
	4F522041			
	54204C4F			
	43415449			
	4F4E5320			
03E7	03	0267		DEFB 03H
03E8	4C4F4144	0268	MSG14	DEFM 'LOAD END ADDRESS= '
	2020454E			
	44202041			
	44445245			
	53533D20			
03FC	03	0269		DEFB 03H

ADDR	CODE	STMT	SOURCE	STATEMENT
		0270	;	
		0271	;	
		0272	;	
		0273	;	
		0274	;	PROM READ SEQUENCE
		0275	;	
		0276	;	
		0277	;	
03FD	CD9105	0278	READ2	CALL CRLF ; READ PROM INTO MEMORY
0400	216703	0279		LD HL,MSG6
0403	CDA005	0280		CALL PTXT ; "READY TO READ (Y/N)?"
0406	CD2D02	0281		CALL YORN
0409	D2E801	0282		JP NC,PPG10A ; NO, THEN PROGRAM
040C	CD9105	0283	READ0	CALL CRLF
040F	214203	0284		LD HL,MSG5
0412	CDA005	0285		CALL PTXT ; "MEM START, MEM END, PROM START
0415	CD1806	0286		CALL SCAN ; INPUT PARAMETERS
0418	3A9106	0287		LD A,(NXCHR)
041B	FE2E	0288		CP ' '
041D	28ED	0289		JR Z,READ0-\$
041F	3E40	0290		LD A,40H
0421	D36A	0291		OUT (HACTL),A ; TURN ON PROM VOLTAGE
0423	2A9606	0292		LD HL,(OPR1) ; MEM START ADDR
0426	ED5B9A06	0293		LD DE,(OPR3) ; PROM START ADDR
042A	CD1705	0294	READ1	CALL PREAD ; READ A BYTE OF PROM
042D	77	0295		LD (HL),A ; SAVE BYTE IN MEM
042E	CD3904	0296		CALL ADCMP ; INC ADDR & COMPARE
0431	20F7	0297		JR NZ,READ1-\$ ; IF NOT END, THEN JUMP
0433	AF	0298		XOR A
0434	D36A	0299		OUT (HACTL),A
0436	C3E801	0300		JP PPG10A ; WHEN COMPLETE, RETURN
		0301	;	
		0302	;	
		0303	;	
		0304	;	
		0305	;	INC HL & DE THEN COMPARE HL WITH (IX)
		0306	;	EXIT WITH ZERO FLAG SET IF EQUAL
		0307	;	
		0308	;	
0439	13	0309	ADCMP	INC DE ; INC PROM ADDR
043A	7A	0310		LD A,D
043B	E607	0311		AND 7H ; MAX PROM ADDR=7FF
043D	57	0312		LD D,A
043E	23	0313		INC HL
043F	D5	0314		PUSH DE
0440	E5	0315		PUSH HL ; SAVE HL & DE
0441	ED5B9806	0316		LD DE,(OPR2) ; GET RAM END ADDR
0445	13	0317		INC DE
0446	A7	0318		AND A
0447	ED52	0319		SBC HL,DE ; COMPARE
0449	E1	0320		POP HL
044A	D1	0321		POP DE
044B	C9	0322		RET
		0323	;	
		0324	;	
		0325	;	
		0326	;	
		0327	;	PROGRAMMING SEQUENCE

ADDR	CODE	STMT	SOURCE	STATEMENT
		0328		;
		0329		;
044C	2A9606	0330	PROG	LD HL,(OPR1) ; MEM START ADDRESS
044F	ED5B9A06	0331		LD DE,(OPR3) ; PROM START ADDRESS
0453	CD1705	0332	PROG1	CALL PREAD ; READ A BYTE (PROM)
0456	FEFF	0333		CP OFFH ; CHECK FOR ERASED BYTES
0458	2007	0334		JR NZ,PROG2-\$ ; IF NOT ERASED JUMP
045A	CD3904	0335	PRG1A	CALL ADCMP ; INC ADDR & CHECK FOR END
045D	20F4	0336		JR NZ,PROG1-\$ ; LOOP UNTIL END
045F	1822	0337		JR PROG3-\$ ; GO PROGRAM PROM
0461	E5	0338	PROG2	PUSH HL
0462	214505	0339		LD HL,MSG7
0465	CDA005	0340		CALL PTXT ; "NOT ERASED"
0468	E1	0341		POP HL
0469	DB68	0342		IN A,(DATA)
046B	CD2605	0343		CALL PRTER ; PRINT BAD LOCATION
046E	CD4D02	0344		CALL ECHO ; READ FROM CONSOLE
0471	4F	0345		LD C,A
0472	FE2E	0346		CP '.' ; CHECK FOR PERIOD
0474	2806	0347		JR Z,PRG2A-\$ ; EXIT
0476	FE43	0348		CP 'C' ; CHECK FOR 'C'
0478	2809	0349		JR Z,PROG3-\$ ; CONTINUE IF 'C' ENTERED
047A	18DE	0350		JR PRG1A-\$ ; IF NOT ,PRINT NEXT ERROR
047C	3E00	0351	PRG2A	LD A,0
047E	D36A	0352		OUT (HACTL),A
0480	C32501	0353		JP PPG0
0483	CD9105	0354	PROG3	CALL CRLF
0486	0665	0355		LD B,LOOPS+1
0488	2A9606	0356	PROG4	LD HL,(OPR1) ; SET MEM START ADDR
048B	ED5B9A06	0357		LD DE,(OPR3) ; SET PROM START ADDR
048F	05	0358		DEC B
0490	2818	0359		JR Z,PROG6-\$
0492	C5	0360	PROG5	PUSH BC
0493	CDEE04	0361		CALL PPROG ; PROGRAM ONE BYTE
0496	C1	0362		POP BC
0497	CD4002	0363		CALL CONST
049A	2807	0364		JR Z,PROG5A-\$
049C	CD4D02	0365		CALL CONIN
049F	FE2E	0366		CP '.'
04A1	2843	0367		JR Z,PROG9-\$
04A3	CD3904	0368	PROG5A	CALL ADCMP ; INC ADDR & CHECK END
04A6	20EA	0369		JR NZ,PROG5-\$ ; FINISH THIS LOOP
04A8	18DE	0370		JR PROG4-\$
04AA	3E00	0371	PROG6	LD A,0
04AC	D36A	0372		OUT (HACTL),A ; TURN OFF ALL CONTROL LINES
04AE	1614	0373		LD D,20
04B0	AF	0374	PROG6B	XOR A
04B1	CD0B05	0375		CALL DELAY
04B4	15	0376		DEC D
04B5	20F9	0377		JR NZ,PROG6B-\$
04B7	2A9606	0378	PROG6A	LD HL,(OPR1) ; MEM START ADDR
04BA	ED5B9A06	0379		LD DE,(OPR3) ; PROM START ADDR
04BE	CD1705	0380	PROG7	CALL PREAD ; READ PROM DATA
04C1	BE	0381		CP (HL)
04C2	2007	0382		JR NZ,PROG8-\$ ; IF NOT MATCH,JUMP
04C4	CD3904	0383	PRG7A	CALL ADCMP ; CHECK FOR END
04C7	20F5	0384		JR NZ,PROG7-\$
04C9	18B1	0385		JR PRG2A-\$ ; EXIT,COMPLETE



ADDR	CODE	STMT	SOURCE	STATEMENT
04CB	F5	0386	PROG8	PUSH AF
04CC	E5	0387		PUSH HL
04CD	215205	0388		LD HL,MSG8
04D0	CDA005	0389		CALL PTXT ; "BAD LOCATION"
04D3	E1	0390		POP HL
04D4	CD4002	0391		CALL CONST
04D7	2807	0392		JR Z,PROG8A-\$
04D9	CD4D02	0393		CALL CONIN
04DC	FE2E	0394		CP '.'
04DE	289C	0395		JR Z,PRG2A-\$
04E0	F1	0396	PROG8A	POP AF
04E1	CD2605	0397		CALL PRTER ; PRINT ERROR MESSAGE
04E4	18DE	0398		JR PRG7A-\$ ; PRINT ALL BAD LOCATIONS
04E6	215C05	0399	PROG9	LD HL,MSG9
04E9	CDA005	0400		CALL PTXT ; ABORTED
04EC	188E	0401		JR PRG2A-\$
		0402		;
		0403		;
		0404		;
		0405		; PULSE ONE PROM LOCATION
		0406		;
		0407		;
04EE	CD2205	0408	PProg	CALL PADD ; LATCH LSB OF ADDRESS
04F1	7A	0409		LD A,D
04F2	F680	0410		OR 080H
04F4	D36A	0411		OUT (HACTL),A
04F6	7E	0412		LD A,(HL) ; FETCH DATA
04F7	D368	0413		OUT (DATA),A ; OUTPUT DATA
04F9	7A	0414		LD A,D
04FA	F6C0	0415		OR 0C0H ; TURN ON PROG PULSE
04FC	D36A	0416		OUT (HACTL),A
04FE	3E01	0417		LD A,1
0500	CD0B05	0418		CALL DELAY ; 1 MILLISEC DELAY
0503	7A	0419		LD A,D
0504	F680	0420		OR 080H
0506	E687	0421		AND 087H ; PROG PULSE OFF
0508	D36A	0422		OUT (HACTL),A
050A	C9	0423		RET
		0424		;
050B	F5	0425	DELAY:	PUSH AF
050C	3A8C06	0426		LD A,(DLAY) ; 2.0 MHZ - 80H
050F	47	0427		LD B,A ; 2.5 MHZ - 0BFH
0510	F1	0428		POP AF ; 4.0 MHZ - 00H
0511	10FE	0429	DELAY1	DJNZ DELAY1-\$
0513	3D	0430		DEC A
0514	20F5	0431		JR NZ,DELAY-\$
0516	C9	0432		RET
		0433		;
		0434		;
		0435		;
		0436		;
		0437		; READ ONE LOCATION OF PROM
		0438		;
		0439		;
0517	CD2205	0440	PREAD	CALL PADD ; LATCH LSB OF ADDRESS
051A	7A	0441		LD A,D
051B	E607	0442		AND 7
051D	D36A	0443		OUT (HACTL),A

ADDR	CODE	STMT	SOURCE	STATEMENT
051F	DB68	0444		IN A, (DATA)
0521	C9	0445		RET
		0446		;
		0447		;
		0448		; LATCH LOWER 8 BITS OF PROM ADDRESS
		0449		;
		0450		;
0522	7B	0451	PADD	LD A,E
0523	D369	0452		OUT (LADD),A
0525	C9	0453		RET
		0454		;
		0455		;
		0456		;
		0457		;
		0458		;
		0459		; PRINTS ADDR/BAD DATA/GOOD DATA
		0460		; OF ERROR LOCATION
		0461		;
		0462		;
0526	47	0463	PRTER	LD B,A
0527	7A	0464		LD A,D
0528	CDAC05	0465		CALL PACC ; PRINT MSB OF ADDR
052B	7B	0466		LD A,E
052C	CDAC05	0467		CALL PACC ; PRINT LSB OF ADDR
052F	0E20	0468		LD C,20H
0531	CD5902	0469		CALL CONOUT
0534	78	0470		LD A,B
0535	CDAC05	0471		CALL PACC ; PRINT BAD DATA
0538	0E20	0472		LD C,20H
053A	CD5902	0473		CALL CONOUT
053D	7E	0474		LD A,(HL)
053E	CDAC05	0475		CALL PACC ; PRINT GOOD DATA
0541	CD9105	0476		CALL CRLF
0544	C9	0477		RET
		0478		;
		0479		;
		0480		;
		0481		;
0545	4E4F5420 45524153 45442020	0482	MSG7	DEFM 'NOT ERASED '
0551	03	0483		DEFB 03H
0552	42414420 4C4F4320 20	0484	MSG8	DEFM 'BAD LOC '
055B	03	0485		DEFB 03H
055C	41424F52 544544	0486	MSG9	DEFM 'ABORTED'
0563	03	0487		DEFB 03H
		0488		;
		0489		;
		0490		;
		0491		; TWDG-READ AND CONVERT 2 DIGITS TO BINARY
		0492		;
		0493		;
0564	CD5A06	0494	TWDG	CALL FREAD ; READ A CHAR, 1ST DIGIT
0567	CD7D05	0495		CALL ASBIN ; ASCII TO BINARY
056A	07	0496		RLCA

ADDR	CODE	STMT	SOURCE	STATEMENT
056B	07	0497		RLCA
056C	07	0498		RLCA
056D	07	0499		RLCA
056E	C5	0500		PUSH BC ; SAVE C REG (CKSM)
056F	4F	0501		LD C,A ; SAVE 1ST DIGIT
0570	CD5A06	0502		CALL FREAD ; 2ND DIGIT
0573	CD7D05	0503		CALL ASBIN
0576	B1	0504		OR C ; OR IN MSB'S
0577	C1	0505		POP BC ; RESTORE CKSM
0578	F5	0506		PUSH AF ; SAVE ACC
0579	81	0507		ADD A,C ; ADD IN 2 DIGITS TO CKSM
057A	4F	0508		LD C,A
057B	F1	0509		POP AF ; RESTORE ACC
057C	C9	0510		RET
		0511		;
		0512		;
		0513		;
		0514		;
		0515		; ASCII TO BINARY CONVERSION
		0516		;
		0517		;
057D	D630	0518	ASBIN	SUB 030H
057F	FE0A	0519		CP 10
0581	F8	0520		RET M
0582	D607	0521		SUB 7
0584	C9	0522		RET
		0523		;
		0524		;
		0525		;
		0526		;
		0527		;

ADDR	CODE	STMT	SOURCE	STATEMENT
		0529		;
		0530		;
0585	C5	0531	PASP	PUSH BC
0586	CDAC05	0532		CALL PACC
0589	CD9B05	0533		CALL SPACE
058C	CD9105	0534		CALL CRLF
058F	C1	0535		POP BC
0590	C9	0536		RET
		0537		;
		0538		;
		0539		;
0591	0E0D	0540	CRLF	LD C,0DH
0593	CD5902	0541		CALL CONOUT
0596	0E0A	0542		LD C,0AH
0598	C35902	0543		JP CONOUT
		0544		;
		0545		;
059B	0E20	0546	SPACE	LD C, ' '
059D	C35902	0547		JP CONOUT
		0548		;
		0549		;
		0550		;
		0551		;
		0552		;
		0553		; PRINT TEXT
		0554		;
		0555		;
05A0	7E	0556	PTXT	LD A,(HL) ; FETCH A BYTE
05A1	FE03	0557		CP 3
05A3	C8	0558		RET Z
05A4	4F	0559		LD C,A
05A5	CD5902	0560		CALL CONOUT
05A8	23	0561		INC HL
05A9	C3A005	0562		JP PTXT
		0563		;
		0564		;
		0565		; PRINT ACCUMULATOR
		0566		;
		0567		;
05AC	F5	0568	PACC	PUSH AF
05AD	0F	0569		RRCA
05AE	0F	0570		RRCA
05AF	0F	0571		RRCA
05B0	0F	0572		RRCA
05B1	CDB505	0573		CALL PRVAL
05B4	F1	0574		POP AF
		0575		;
		0576		;
05B5	E60F	0577	PRVAL	AND 0FH
05B7	C690	0578		ADD A,90H
05B9	27	0579		DAA
05BA	CE40	0580		ADC A,40H
05BC	27	0581		DAA
05BD	4F	0582		LD C,A
05BE	C35902	0583		JP CONOUT ; PRINT IT
		0584		;
		0585		;

ADDR	CODE	STMT	SOURCE	STATEMENT
		0586	;	
		0587	;	
		0588	;	CHECK FOR VALID HEX CHARACTER
		0589	;	
		0590	;	
05C1	FE30	0591	AORN	CP '0'
05C3	DAD705	0592		JP C,AORN2 ; JUMP IF < 30H
05C6	FE3A	0593		CP '9'+1
05C8	DAD505	0594		JP C,AORN1 ; JUMP IF < 3AH
05CB	FE40	0595		CP 'A'-1
05CD	DAD705	0596		JP C,AORN2 ; JUMP IF < 'A'
05D0	FE47	0597		CP 'F'+1
05D2	D2D705	0598		JP NC,AORN2 ; JUMP IF < 'G'
05D5	AF	0599	AORN1	XOR A
05D6	C9	0600		RET ; VALID DATA RET
05D7	AF	0601	AORN2	XOR A
05D8	3C	0602		INC A
05D9	C9	0603		RET ; NOT HEX CHAR
		0604	;	
		0605	;	
		0606	;	CHECK FOR TERMINATOR
		0607	;	
		0608	;	SPACE, COMMA, OR CARRIAGE RETURN
		0609	;	
		0610	;	
05DA	FE20	0611	TERMCK	CP ' '
05DC	C8	0612		RET Z
05DD	FE2C	0613		CP ','
05DF	C8	0614		RET Z
05E0	FE2E	0615		CP '. '
05E2	CAE805	0616		JP Z,TCHK0
05E5	FE0D	0617		CP 0DH
05E7	C0	0618		RET NZ
05E8	C5	0619	TCHK0	PUSH BC
05E9	CD9105	0620		CALL CRLF
05EC	C1	0621		POP BC
05ED	AF	0622		XOR A
05EE	C9	0623		RET
		0624	;	
		0625	;	
		0626	;	SCAN FOR OPERAND FROM KEYBOARD
		0627	;	
		0628	;	
		0629	;	EXIT WITH DATA IN HL, AND TERMINATOR
		0630	;	IN C. IF VALID DATA, RETURN WITH
		0631	;	ZERO FLAG SET. B CONTAINS # OF CHARACTERS ENTERED.
		0632	;	
		0633	;	
05EF	210000	0634	KEYIN	LD HL,0
05F2	45	0635		LD B,L
05F3	CD4D02	0636	KEY1	CALL ECHO
05F6	4F	0637		LD C,A
05F7	04	0638		INC B ; INC CHAR COUNT
05F8	CDDA05	0639		CALL TERMCK
05FB	C8	0640		RET Z ; IF TERMINATOR, RETURN
05FC	CDC105	0641		CALL AORN ; VALID DATA CHECK
05FF	C0	0642		RET NZ ; IF NOT RETURN
0600	79	0643		LD A,C

ADDR	CODE	STMT	SOURCE	STATEMENT
0601	CD7D05	0644	CALL	ASBIN ; CONVERT TO BINARY
0604	29	0645	ADD	HL,HL
0605	29	0646	ADD	HL,HL
0606	29	0647	ADD	HL,HL
0607	29	0648	ADD	HL,HL ; SHIFT 4 BITS
0608	85	0649	ADD	A,L
0609	6F	0650	LD	L,A
060A	C3F305	0651	JP	KEY1
		0652		;
		0653		;
		0654		;
		0655		;
060D	0E3F	0656	INVCMD	LD C,'?'
060F	CD5902	0657	CALL	CONOUT
0612	3E2E	0658	LD	A,'.'
0614	329106	0659	LD	(NXCHR),A
0617	C9	0660	RET	
		0661		;
		0662		;
		0663		;
		0664		;
		0665		;SCAN -- SCANS THE OPERANDS INPUT FROM THE
		0666		CONSOLE. THE OPERANDS ARE SAVED IN OPI,
		0667		AS THE ARGUMENTS TO EXECUTE THEIR COMMANDS.
		0668		;
0618	AF	0669	SCAN	XOR A ;INITIALIZE
0619	219606	0670	LD	HL,OPRS ;CLEAR OPERANDS
061C	E5	0671	PUSH	HL
061D	DDE1	0672	POP	IX ;IX=POINTER TO OPERANDS
061F	77	0673	LD	(HL),A
0620	010900	0674	LD	BC,9
0623	119706	0675	LD	DE,OPRS+1
0626	EDB0	0676	LDIR	
0628	329006	0677	LD	(OPCNT),A
062B	CDEF05	0678	SCAN1	CALL KEYIN ; GET ONE OPERAND
062E	C20D06	0679	JP	NZ,INVCMD
0631	79	0680	LD	A,C
0632	329106	0681	LD	(NXCHR),A
0635	FE20	0682	CP	' '
0637	2806	0683	JR	Z,SCAN2-\$
0639	FE2C	0684	CP	' '
063B	2802	0685	JR	Z,SCAN2-\$
063D	05	0686	DEC	B
063E	C8	0687	RET	Z ; IF NO DATA, RET
063F	DD7500	0688	SCAN2	LD (IX),L
0642	DD7401	0689	LD	(IX+1),H
0645	3A9006	0690	LD	A,(OPCNT)
0648	3C	0691	INC	A
0649	329006	0692	LD	(OPCNT),A
064C	DD23	0693	INC	IX
064E	DD23	0694	INC	IX ; POINT TO NEXT OPR
0650	79	0695	LD	A,C
0651	FE20	0696	CP	' '
0653	28D6	0697	JR	Z,SCAN1-\$
0655	FE2C	0698	CP	' '
0657	28D2	0699	JR	Z,SCAN1-\$
0659	C9	0700	RET	
		0701		;

ADDR	CODE	STMT	SOURCE	STATEMENT
		0702	;	
		0703	;	
065A	E5	0704	FREAD	PUSH HL
065B	D5	0705		PUSH DE
065C	C5	0706		PUSH BC
065D	3A9306	0707		LD A, (BCOUNT)
0660	FE80	0708		CP NBYTES
0662	200C	0709		JR NZ, FREAD1-\$
0664	0E14	0710		LD C, READ
0666	115C00	0711		LD DE, FCB
0669	CD0500	0712		CALL FDOS ; LEAD FILE
066C	B7	0713		OR A
066D	2011	0714		JR NZ, FERROR-\$
066F	AF	0715		XOR A
0670	218000	0716	FREAD1	LD HL, BUFFER
0673	5F	0717		LD E, A
0674	3C	0718		INC A
0675	329306	0719		LD (BCOUNT), A
0678	1600	0720		LD D, 0
067A	19	0721		ADD HL, DE ; POINT TO NEXT DATA
067B	7E	0722		LD A, (HL)
067C	C1	0723		POP BC
067D	D1	0724		POP DE
067E	E1	0725		POP HL
067F	C9	0726		RET
0680	C1	0727	FERROR	POP BC
0681	D1	0728		POP DE
0682	E1	0729		POP HL
0683	219203	0730		LD HL, MSG11
0686	CDA005	0731		CALL PTXT ; ATTEMPT TO READ PAST EOF
0689	C36602	0732		JP RENTRY
		0733	;	
>068C		0734	DLAY	DEFS 1
>068D		0735	SCR1	DEFS 1
>068E		0736	SCR2	DEFS 1
>068F		0737	SCR3	DEFS 1
>0690		0738	OPCNT	DEFS 1
>0691		0739	NXCHR	DEFS 1
>0692		0740	FOPEN	DEFS 1
>0693		0741	BCOUNT	DEFS 1
>0694		0742	LAST	DEFS 2
>0696		0743	OPRS	DEFS 6
>0696		0744	OPR1	EQU OPRS
>0698		0745	OPR2	EQU OPRS+2
>069A		0746	OPR3	EQU OPRS+4
>0796		0747	SPVAL	EQU OPRS+100H
		0748	;	
		0749	;	
		0750	;	

ADDR CODE STMT SOURCE STATEMENT

## CROSS REFERENCE LISTING

SYMBOL	VALUE	TYPE	STMT	STATEMENT REFERENCES								
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----
ADCMP	0439		0309	0383	0368	0335	0296					
AORN	05C1		0591	0641								
AORN1	05D5		0599	0594								
AORN2	05D7		0601	0598	0596	0592						
ASBIN	057D		0518	0644	0503	0495						
BCOUNT	0693		0741	0719	0707	0058						
BUFFER	0080		0022	0716	0084							
CLOSE	0010		0024	0230								
CONIN	024D		0203	0393	0365	0049						
CONOUT	0259		0213	0657	0583	0560	0547	0543	0541	0473	0469	
CONST	0240		0191	0391	0363							
CRLF	0591		0540	0620	0534	0476	0354	0283	0278	0226	0148	
				0143	0096	0071	0064					
DATA	0068		0016	0444	0413	0342						
DELAY	050B		0425	0431	0418	0375						
DELAY1	0511		0429	0429								
DLAY	068C		0734	0426	0056							
ECHO	024D		0202	0636	0344	0178						
EXIT	0000		0014	0233	0229							
FCB	005C		0021	0711	0231	0076						
FDOS	0005		0020	0712	0232	0085	0077					
FERROR	0680		0727	0714								
FOPEN	0692		0740	0227	0072	0060						
FREAD	065A		0704	0502	0494	0159						
FREAD1	0670		0716	0709								
HACTL	006A		0018	0443	0422	0416	0411	0372	0352	0299	0291	
				0225	0063							
INVCMD	060D		0656	0679								
KEY1	05F3		0636	0651								
KEYIN	05EF		0634	0678								
LADD	0069		0017	0452								
LAST	0694		0742	0141	0115							
LOOPS	0064		0028	0355								
MSG1	0286		0243	0046								
MSG10	0383		0260	0080								
MSG11	0392		0262	0730								
MSG12	03B6		0264	0092								
MSG13	03CB		0266	0128								
MSG14	03E8		0268	0139								
MSG2	02EB		0250	0065								
MSG3	0307		0252	0086								
MSG4	0323		0254	0144								
MSG5	0342		0256	0284	0149							
MSG6	0367		0258	0279								
MSG7	0545		0482	0339								
MSG8	0552		0484	0388								
MSG9	055C		0486	0399								
NBYTES	0080		0027	0708	0057							
NXCHR	0691		0739	0681	0659	0287	0152	0089				
OPCNT	0690		0738	0692	0690	0677	0100					
OPEN	000F		0023	0075								
OPR1	0696		0744	0378	0356	0330	0292	0099	0094			
OPR2	0698		0745	0316	0133							





ADDR	CODE	STMT	SOURCE	STATEMENT
TBASE	0100	0015	0037	
TCHK0	05E8	0619	0616	
TERMCK	05DA	0611	0639	
THDR	020D	0159	0161	0109 0097
THDR1	0229	0174	0166	
TWDG	0564	0494	0172	0170 0168 0164 0123 0113
YORN	022D	0178	0281	0186 0146 0069

ERRORS=0000

ADDR	CODE	STMT	SOURCE	STATEMENT
		0001	NAME	PROM16
		0002	;	
		0003	;	
		0004	;	2716/2758 PROM PROGRAMMER CONTROL PROGRAM
		0005	;	SDOS ENVIRONMENT
		0006	;	9/27/79
		0007	;	
		0008	;	VERSION 1.3 CP/M COMPATIBLE 8/26/80
		0009	;	
		0010	;	
		0011	;	SYSTEM EQUATES
		0012	;	
		0013	;	
>0000		0014	EXIT EQU	0H
>0100		0015	TBASE EQU	100H ; BASE OF TRANSIENT AREA
>0068		0016	DATA EQU	68H ; DATA PORT
>0069		0017	LADD EQU	69H ; LOW ADDRESS PORT
>006A		0018	HACTL EQU	6AH ; HIGH ADDRESS + CONTROL PORT
>0000		0019	RBOU	0
>0005		0020	FDOS EQU	5
>005C		0021	FCB EQU	5CH
>0080		0022	BUFFER EQU	80H
>000F		0023	OPEN EQU	15
>0010		0024	CLOSE EQU	16
>0014		0025	READ EQU	20
>001A		0026	SETDMA EQU	26
>0080		0027	NBYTES EQU	128
>0064		0028	LOOPS EQU	100
		0029	;	
		0030	;	
		0031	;	
		0032	;	
		0033	;	
		0034	;	
		0035	PSECT ABS	
		0036	;	
>0100		0037	ORG TBASE	; START AT 100H
		0038	;	
		0039	;	
		0040	;	
		0041	;	PROGRAM PROM ENTRY POINT
		0042	;	
		0043	;	
		0044	;	
0100	318507	0045	PPG LD	SP,SPVAL
0103	218602	0046	LD	HL,MSG1
0106	CD8F05	0047	CALL	PTXT ; "PROM16 VERSION 1.3"
0109	1E25	0048	LD	E,25H
010B	CD4D02	0049	CALL	CONIN
010E	FE31	0050	CP	31H
0110	3806	0051	JR	C,SETDLY
0112	1E32	0052	LD	E,32H
0114	2802	0053	JR	Z,SETDLY
0116	1E4A	0054	LD	E,4AH
0118	7B	0055	SETDLY: LD	A,E
0119	327B06	0056	LD	(DLAY),A
011C	3E80	0057	LD	A,NBYTES
011E	328206	0058	LD	(BCOUNT),A

ADDR	CODE	STMT	SOURCE	STATEMENT
0121	AF	0059		XOR A
0122	328106	0060		LD (FOPEN),A
0125	318507	0061	PPG0	LD SP,SPVAL
0128	AF	0062		XOR A ; CLEAR ACC
0129	D36A	0063		OUT (HACTL),A
012B	CD8005	0064		CALL CRLF
012E	21EB02	0065		LD HL,MSG2
0131	CD8F05	0066		CALL PTXT ; "READY TO LOAD A FILE ?"
0134	210000	0067		LD HL,0
0137	227C06	0068		LD (SCR1),HL ; CLEAR BYTE COUNTER
013A	CD2D02	0069		CALL YORN
013D	D2FD03	0070		JP NC,READ2 ; IF NOT,CHK READ
0140	CD8005	0071	PPG0B	CALL CRLF
0143	3A8106	0072		LD A,(FOPEN)
0146	B7	0073		OR A
0147	2015	0074		JR NZ,PPG0C-\$
0149	0E0F	0075		LD C,OPEN
014B	115C00	0076		LD DE,FCB
014E	CD0500	0077		CALL FDOS
0151	FEFF	0078		CP 255
0153	2009	0079		JR NZ,PPG0C-\$
0155	218303	0080		LD HL,MSG10
0158	CD8F05	0081		CALL PTXT
015B	C36602	0082		JP RENTRY
015E	0E1A	0083	PPG0C	LD C,SETDMA
0160	118000	0084		LD DE,BUFFER
0163	CD0500	0085		CALL FDOS
0166	210703	0086		LD HL,MSG3
0169	CD8F05	0087		CALL PTXT ; "HEX LOAD ADDRESS, # BYTES
016C	CD0706	0088		CALL SCAN ; COLLECT DATA
016F	3A8006	0089		LD A,(NXCHR)
0172	FE2E	0090		CP ' . '
0174	28CA	0091		JR Z,PPG0B-\$
0176	21B603	0092		LD HL,MSG12
0179	CD8F05	0093		CALL PTXT
017C	2A8506	0094		LD HL,(OPR1)
017F	CD7E02	0095		CALL PADDR
0182	CD8005	0096		CALL CRLF
0185	CD0D02	0097		CALL THDR ; READ ADDR/#BYTES 1ST LINE
0188	E5	0098		PUSH HL
0189	ED5B8506	0099		LD DE,(OPR1) ; LOAD ADDRESS ENTERED
018D	3A7F06	0100		LD A,(OPCNT)
0190	A7	0101		AND A
0191	2804	0102		JR Z,PPG0A-\$ ; IF NO OPR'S JUMP
0193	ED52	0103		SBC HL,DE ; COMPUTE OFFSET
0195	E5	0104		PUSH HL
0196	D1	0105		POP DE ; OFFSET REG
0197	E1	0106	PPG0A	POP HL
0198	ED537E06	0107		LD (SCR3),DE ; SAVE OFFSET
019C	1803	0108		JR PPG2-\$
019E	CD0D02	0109	PPG1	CALL THDR ; LOAD ADDR/# BYTES
01A1	ED5B7E06	0110	PPG2	LD DE,(SCR3)
01A5	B7	0111		OR A ; CLEAR CARRY
01A6	ED52	0112		SBC HL,DE ; SUBTRACT OFFSET
01A8	CD5305	0113	PPG3	CALL TWDG ; READ A BYTE
01AB	77	0114		LD (HL),A ; STORE IT
01AC	228306	0115		LD (LAST),HL
01AF	23	0116		INC HL

ADDR	CODE	STMT	SOURCE	STATEMENT
01B0	E5	0117		PUSH HL
01B1	2A7C06	0118		LD HL,(SCR1) ; BYTE COUNTER
01B4	23	0119		INC HL
01B5	227C06	0120		LD (SCR1),HL
01B8	E1	0121		POP HL
01B9	10ED	0122		DJNZ PPG3-\$ ; READ ENTIRE LINE
01BB	CD5305	0123		CALL TWDG ; READ CHECKSUM
01BE	AF	0124		XOR A
01BF	81	0125		ADD A,C
01C0	280B	0126		JR Z,PPG4-\$
01C2	E5	0127		PUSH HL
01C3	21CB03	0128		LD HL,MSG13
01C6	CD8F05	0129		CALL PTXT
01C9	CD7E02	0130		CALL PADDR ; CHECKSUM ERROR
01CC	E1	0131		POP HL
01CD	2A7C06	0132	PPG4	LD HL,(SCR1)
01D0	ED4B8706	0133		LD BC,(OPR2) ; # OF BYTES TO READ
01D4	2B	0134		DEC HL ; IF 0,THEN FFFF
01D5	0B	0135		DEC BC
01D6	A7	0136		AND A
01D7	ED42	0137		SBC HL,BC
01D9	DA9E01	0138		JP C,PPG1 ; IF MORE,JUMP
01DC	21E803	0139	PPG10	LD HL,MSG14
01DF	CD8F05	0140		CALL PTXT
01E2	2A8306	0141		LD HL,(LAST)
01E5	CD7E02	0142		CALL PADDR
01E8	CD8005	0143	PPG10A	CALL CRLF
01EB	212303	0144		LD HL,MSG4
01EE	CD8F05	0145		CALL PTXT ; "READY TO PROGRAM A PROM ?"
01F1	CD2D02	0146		CALL YORN
01F4	D26602	0147		JP NC,REENTRY
01F7	CD8005	0148	PPG11	CALL CRLF
01FA	214203	0149		LD HL,MSG5
01FD	CD8F05	0150		CALL PTXT ; "MEM START,MEM END,PROM START"
0200	CD0706	0151		CALL SCAN
0203	3A8006	0152		LD A,(NXCHR)
0206	FE2E	0153		CP '.' ; IF '.' EXIT
0208	28ED	0154		JR Z,PPG11-\$
020A	C34C04	0155		JP PROG
		0156		;
		0157		;
		0158		;
020D	CD4906	0159	THDR	CALL FREAD ; LOOK FOR ':'
0210	FE3A	0160		CP ':'
0212	20F9	0161		JR NZ,THDR-\$
0214	AF	0162		XOR A ; CLEAR CHECKSUM
0215	4F	0163		LD C,A
0216	CD5305	0164		CALL TWDG
0219	A7	0165		AND A ; IS RECORD LENGTH 0?
021A	280D	0166		JR Z,THDR1-\$
021C	47	0167		LD B,A
021D	CD5305	0168		CALL TWDG ; GET FIRST BYTE OF ADDRESS
0220	67	0169		LD H,A
0221	CD5305	0170		CALL TWDG ; GET SECOND BYTE
0224	6F	0171		LD L,A
0225	CD5305	0172		CALL TWDG ; GET RECORD TYPE
0228	C9	0173		RET
0229	F1	0174	THDR1	POP AF

ADDR	CODE	STMT	SOURCE	STATEMENT
022A	C3DC01	0175		JP PPG10
		0176		;
		0177		;
022D	CD4D02	0178	YORN	CALL ECHO
0230	4F	0179		LD C,A
0231	FE2E	0180		CP '.'
0233	CA6602	0181		JP Z,REENTRY
0236	B7	0182		OR A
0237	FE4E	0183		CP 'N'
0239	C8	0184		RET Z
023A	FE59	0185		CP 'Y'
023C	20EF	0186		JR NZ,YORN-\$
023E	37	0187		SCF
023F	C9	0188		RET
		0189		;
		0190		;
0240	C5	0191	CONST	PUSH BC ; CALL SDOS CONSOLE STATUS
0241	D5	0192		PUSH DE
0242	E5	0193		PUSH HL
0243	0E0B	0194		LD C,0BH
0245	CD0500	0195		CALL 5
0248	B7	0196		OR A
0249	E1	0197		POP HL
024A	D1	0198		POP DE
024B	C1	0199		POP BC
024C	C9	0200		RET
		0201		;
>024D		0202	ECHO:	
024D	C5	0203	CONIN:	PUSH BC
024E	D5	0204		PUSH DE
024F	E5	0205		PUSH HL
0250	0E01	0206		LD C,1 ; CALL SDOS CONSOLE INPUT
0252	CD0500	0207		CALL 5
0255	E1	0208		POP HL
0256	D1	0209		POP DE
0257	C1	0210		POP BC
0258	C9	0211		RET
		0212		;
0259	D5	0213	CONOUT:	PUSH DE
025A	C5	0214		PUSH BC
025B	E5	0215		PUSH HL
025C	59	0216		LD E,C
025D	0E02	0217		LD C,02
025F	CD0500	0218		CALL 5
0262	E1	0219		POP HL
0263	C1	0220		POP BC
0264	D1	0221		POP DE
0265	C9	0222		RET
		0223		;
0266	AF	0224	REENTRY	XOR A
0267	D36A	0225		OUT (HACTL),A
0269	CD8005	0226		CALL CRLF
026C	3A8106	0227		LD A,(FOPEN)
026F	B7	0228		OR A
0270	CA0000	0229		JP Z,EXIT
0273	0E10	0230		LD C,CLOSE
0275	115C00	0231		LD DE,FCB
0278	CD0500	0232		CALL FDOS

ADDR	CODE	STMT	SOURCE	STATEMENT
027B	C30000	0233		JP EXIT
		0234		;
		0235		;
027E	7C	0236	PADDR	LD A,H
027F	CD9B05	0237		CALL PACC
0282	7D	0238		LD A,L
0283	C37405	0239		JP PASP
		0240		;
		0241		;
		0242		;
0286	50524F4D	0243	MSG1	DEFM 'PROM-16 VERSION 1.3'
	2D313620			
	56455253			
	494F4E20			
	312E33			
0299	0D0A0D0A	0244		DEFW 0A0DH,0A0DH
029D	49532054	0245		DEFM 'IS THIS SYSTEM RUNNING 2.0, 2.5, OR 4.0
	48495320			
	53595354			
	454D2052			
	554E4E49			
	4E472032			
	2E302C20			
	322E352C			
	204F5220			
	342E3020			
	4D485A3F			
02C9	0D0A	0246		DEFW 0A0DH
02CB	28322E30	0247		DEFM '(2.0 - 0 , 2.5 - 1 , 4.0 - 2)'
	202D2030			
	202C2032			
	2E35202D			
	2031202C			
	20342E30			
	202D2032			
	29			
02E8	0A0D	0248		DEFW 0D0AH
02EA	03	0249		DEFB 03H
02EB	52454144	0250	MSG2	DEFM 'READY TO LOAD A FILE (Y/N)?'
	5920544F			
	204C4F41			
	44204120			
	46494C45			
	2028592F			
	4E293F			
0306	03	0251		DEFB 03H
0307	48455820	0252	MSG3	DEFM 'HEX LOAD ADDRESS, # BYTES: '
	4C4F4144			
	20414444			
	52455353			
	2C202320			
	42595445			
	533A20			
0322	03	0253		DEFB 03H
0323	52454144	0254	MSG4	DEFM 'READY TO PROGRAM A PROM (Y/N)?'
	5920544F			
	2050524F			
	4752414D			

ADDR	CODE	STMT	SOURCE	STATEMENT
	20412050			
	524F4D20			
	28592F4E			
	293F			
0341	03	0255		DEFB 03H
0342	4D454D4F	0256	MSG5	DEFM 'MEMORY START, MEMORY END, PROM START: '
	52592053			
	54415254			
	2C4D454D			
	4F525920			
	454E442C			
	50524F4D			
	20535441			
	52543A20			
0366	03	0257		DEFB 03H
0367	52454144	0258	MSG6	DEFM 'READY TO READ A PROM (Y/N)?'
	5920544F			
	20524541			
	44204120			
	50524F4D			
	2028592F			
	4E293F			
0382	03	0259		DEFB 03H
0383	46494C45	0260	MSG10	DEFM 'FILE NOT FOUND'
	204E4F54			
	20464F55			
	4E44			
0391	03	0261		DEFB 03H
0392	41545445	0262	MSG11	DEFM 'ATTEMPTING TO READ PAST END OF FILE
	4D505449			
	4E472054			
	4F205245			
	41442050			
	41535420			
	454E4420			
	4F462046			
	494C45			
03B5	03	0263		DEFB 03H
03B6	4C4F4144	0264	MSG12	DEFM 'LOAD START ADDRESS= '
	20535441			
	52542041			
	44445245			
	53533D20			
03CA	03	0265		DEFB 03H
03CB	43484543	0266	MSG13	DEFM 'CHECKSUM ERROR AT LOCATIONS '
	4B53554D			
	20455252			
	4F522041			
	54204C4F			
	43415449			
	4F4E5320			
03E7	03	0267		DEFB 03H
03E8	4C4F4144	0268	MSG14	DEFM 'LOAD END ADDRESS= '
	2020454E			
	44202041			
	44445245			
	53533D20			
03FC	03	0269		DEFB 03H



ADDR	CODE	STMT	SOURCE	STATEMENT
		0270		;
		0271		;
		0272		;
		0273		;
		0274		; PROM READ SEQUENCE
		0275		;
		0276		;
		0277		;
03FD	CD8005	0278	READ2	CALL CRLF ; READ PROM INTO MEMORY
0400	216703	0279		LD HL,MSG6
0403	CD8F05	0280		CALL PTXT ; "READY TO READ (Y/N)?"
0406	CD2D02	0281		CALL YORN
0409	D2E801	0282		JP NC,PPG10A ; NO, THEN PROGRAM
040C	CD8005	0283	READ0	CALL CRLF
040F	214203	0284		LD HL,MSG5
0412	CD8F05	0285		CALL PTXT ; "MEM START, MEM END, PROM START"
0415	CD0706	0286		CALL SCAN ; INPUT PARAMETERS
0418	3A8006	0287		LD A,(NXCHR)
041B	FE2E	0288		CP '.'
041D	28ED	0289		JR Z,READ0-\$
041F	3E40	0290		LD A,40H
0421	D36A	0291		OUT (HACTL),A ; TURN ON PROM VOLTAGE
0423	2A8506	0292		LD HL,(OPR1) ; MEM START ADDR
0426	ED5B8906	0293		LD DE,(OPR3) ; PROM START ADDR
042A	CD0605	0294	READ1	CALL PREAD ; READ A BYTE OF PROM
042D	77	0295		LD (HL),A ; SAVE BYTE IN MEM
042E	CD3904	0296		CALL ADCMP ; INC ADDR & COMPARE
0431	20F7	0297		JR NZ,READ1-\$ ; IF NOT END, THEN JUMP
0433	AF	0298		XOR A
0434	D36A	0299		OUT (HACTL),A
0436	C3E801	0300		JP PPG10A ; WHEN COMPLETE, RETURN
		0301		;
		0302		;
		0303		;
		0304		;
		0305		; INC HL & DE THEN COMPARE HL WITH (IX)
		0306		; EXIT WITH ZERO FLAG SET IF EQUAL
		0307		;
		0308		;
0439	13	0309	ADCMP	INC DE ; INC PROM ADDR
043A	7A	0310		LD A,D
043B	E607	0311		AND 7H ; MAX PROM ADDR=7FF
043D	57	0312		LD D,A
043E	23	0313		INC HL
043F	D5	0314		PUSH DE
0440	E5	0315		PUSH HL ; SAVE HL & DE
0441	ED5B8706	0316		LD DE,(OPR2) ; GET RAM END ADDR
0445	13	0317		INC DE
0446	A7	0318		AND A
0447	ED52	0319		SBC HL,DE ; COMPARE
0449	E1	0320		POP HL
044A	D1	0321		POP DE
044B	C9	0322		RET
		0323		;
		0324		;
		0325		;
		0326		;
		0327		; PROGRAMMING SEQUENCE

ADDR	CODE	STMT	SOURCE	STATEMENT
		0328		;
		0329		;
044C	3E40	0330	PROG	LD A,40H
044E	D36A	0331		OUT (HACTL),A ; TURN ON PROM VOLTAGE
0450	2A8506	0332		LD HL,(OPR1) ; MEM START ADDRESS
0453	ED5B8906	0333		LD DE,(OPR3) ; PROM START ADDRESS
0457	CD0605	0334	PROG1	CALL PREAD ; READ A BYTE (PROM)
045A	FEFF	0335		CP OFFH ; CHECK FOR ERASED BYTES
045C	2007	0336		JR NZ,PROG2-\$ ; IF NOT ERASED JUMP
045E	CD3904	0337	PRG1A	CALL ADCMP ; INC ADDR & CHECK FOR END
0461	20F4	0338		JR NZ,PROG1-\$ ; LOOP UNTIL END
0463	1822	0339		JR PROG3-\$ ; GO PROGRAM PROM
0465	E5	0340	PROG2	PUSH HL
0466	213405	0341		LD HL,MSG7
0469	CD8F05	0342		CALL PTXT ; "NOT ERASED"
046C	E1	0343		POP HL
046D	DB68	0344		IN A,(DATA)
046F	CD1505	0345		CALL PRTER ; PRINT BAD LOCATION
0472	CD4D02	0346		CALL ECHO ; READ FROM CONSOLE
0475	4F	0347		LD C,A
0476	FE2E	0348		CP '.' ; CHECK FOR PERIOD
0478	2806	0349		JR Z,PRG2A-\$ ; EXIT
047A	FE43	0350		CP 'C' ; CHECK FOR 'C'
047C	2809	0351		JR Z,PROG3-\$ ; CONTINUE IF 'C' ENTERED
047E	18DE	0352		JR PRG1A-\$ ; IF NOT ,PRINT NEXT ERROR
0480	3E00	0353	PRG2A	LD A,0
0482	D36A	0354		OUT (HACTL),A
0484	C32501	0355		JP PPG0
0487	CD8005	0356	PROG3	CALL CRLF
048A	2A8506	0357	PROG4	LD HL,(OPR1) ; SET MEM START ADDR
048D	ED5B8906	0358		LD DE,(OPR3) ; SET PROM START ADDR
0491	CDE004	0359	PROG5	CALL PPROG ; PROGRAM ONE BYTE
0494	CD4002	0360		CALL CONST
0497	2807	0361		JR Z,PROG5A-\$
0499	CD4D02	0362		CALL CONIN
049C	FE2E	0363		CP '.'
049E	2838	0364		JR Z,PROG9-\$
04A0	CD3904	0365	PROG5A	CALL ADCMP ; INC ADDR & CHECK END
04A3	20EC	0366		JR NZ,PROG5-\$ ; FINISH THIS LOOP
04A5	3E00	0367	PROG6	LD A,0
04A7	D36A	0368		OUT (HACTL),A ; TURN OFF ALL CONTROL LINES
04A9	2A8506	0369	PROG6A	LD HL,(OPR1) ; MEM START ADDR
04AC	ED5B8906	0370		LD DE,(OPR3) ; PROM START ADDR
04B0	CD0605	0371	PROG7	CALL PREAD ; READ PROM DATA
04B3	BE	0372		CP (HL)
04B4	2007	0373		JR NZ,PROG8-\$ ; IF NOT MATCH,JUMP
04B6	CD3904	0374	PRG7A	CALL ADCMP ; CHECK FOR END
04B9	20F5	0375		JR NZ,PROG7-\$
04BB	18C3	0376		JR PRG2A-\$ ; EXIT,COMPLETE
04BD	F5	0377	PROG8	PUSH AF
04BE	E5	0378		PUSH HL
04BF	214105	0379		LD HL,MSG8
04C2	CD8F05	0380		CALL PTXT ; "BAD LOCATION"
04C5	E1	0381		POP HL
04C6	CD4002	0382		CALL CONST
04C9	2807	0383		JR Z,PROG8A-\$
04CB	CD4D02	0384		CALL CONIN
04CE	FE2E	0385		CP '.'

ADDR	CODE	STMT	SOURCE	STATEMENT
04D0	28AE	0386		JR Z,PRG2A-\$
04D2	F1	0387	PROG8A	POP AF
04D3	CD1505	0388		CALL PRTER ; PRINT ERROR MESSAGE
04D6	18DE	0389		JR PRG7A-\$ ; PRINT ALL BAD LOCATIONS
04D8	214B05	0390	PROG9	LD HL,MSG9
04DB	CD8F05	0391		CALL PTXT ; ABORTED
04DE	18A0	0392		JR PRG2A-\$
		0393		;
		0394		;
		0395		;
		0396		;
		0397		;
		0398		;
04E0	CD1105	0399	PProg	CALL PADD ; LATCH LSB OF ADDRESS
04E3	7A	0400		LD A,D
04E4	F6C0	0401		OR 0C0H
04E6	D36A	0402		OUT (HACTL),A
04E8	7E	0403		LD A,(HL) ; FETCH DATA
04E9	D368	0404		OUT (DATA),A ; OUTPUT DATA
04EB	7A	0405		LD A,D
04EC	F6E0	0406		OR 0E0H ; TURN ON PROG PULSE
04EE	D36A	0407		OUT (HACTL),A
04F0	3A7B06	0408		LD A,(DLAY) ; 2.0 MHZ - 25H
		0409		;
		0410		;
				;
04F3	CDFE04	0411		CALL DELAY ; 50 MILLISEC DELAY
04F6	7A	0412		LD A,D
04F7	F6C0	0413		OR 0C0H
04F9	E6C7	0414		AND 0C7H ; PROG PULSE OFF
04FB	D36A	0415		OUT (HACTL),A
04FD	C9	0416		RET
		0417		;
04FE	06BF	0418	DELAY	LD B,191
0500	10FE	0419	DELAY1	DJNZ DELAY1-\$
0502	3D	0420		DEC A
0503	20F9	0421		JR NZ,DELAY-\$
0505	C9	0422		RET
		0423		;
		0424		;
		0425		;
		0426		;
		0427		;
		0428		;
		0429		;
0506	CD1105	0430	PREAD	CALL PADD ; LATCH LSB OF ADDRESS
0509	7A	0431		LD A,D
050A	F640	0432		OR 40H
050C	D36A	0433		OUT (HACTL),A
050E	DB68	0434		IN A,(DATA)
0510	C9	0435		RET
		0436		;
		0437		;
		0438		;
		0439		;
		0440		;
		0441		;
0511	7B	0441	PADD	LD A,E
0512	D369	0442		OUT (LADD),A
0514	C9	0443		RET

ADDR	CODE	STMT	SOURCE	STATEMENT
		0444		;
		0445		;
		0446		;
		0447		;
		0448		;
		0449		; PRINTS ADDR/BAD DATA/GOOD DATA
		0450		; OF ERROR LOCATION
		0451		;
		0452		;
0515	47	0453	PRTER	LD B,A
0516	7A	0454		LD A,D
0517	CD9B05	0455		CALL PACC ; PRINT MSB OF ADDR
051A	7B	0456		LD A,E
051B	CD9B05	0457		CALL PACC ; PRINT LSB OF ADDR
051E	0E20	0458		LD C,20H
0520	CD5902	0459		CALL CONOUT
0523	78	0460		LD A,B
0524	CD9B05	0461		CALL PACC ; PRINT BAD DATA
0527	0E20	0462		LD C,20H
0529	CD5902	0463		CALL CONOUT
052C	7E	0464		LD A,(HL)
052D	CD9B05	0465		CALL PACC ; PRINT GOOD DATA
0530	CD8005	0466		CALL CRLF
0533	C9	0467		RET
		0468		;
		0469		;
		0470		;
		0471		;
0534	4E4F5420 45524153 45442020	0472	MSG7	DEFM 'NOT ERASED '
0540	03	0473		DEFB 03H
0541	42414420 4C4F4320 20	0474	MSG8	DEFM 'BAD LOC '
054A	03	0475		DEFB 03H
054B	41424F52 544544	0476	MSG9	DEFM 'ABORTED'
0552	03	0477		DEFB 03H
		0478		;
		0479		;
		0480		;
		0481		; TWDG-READ AND CONVERT 2 DIGITS TO BINARY
		0482		;
		0483		;
0553	CD4906	0484	TWDG	CALL FREAD ; READ A CHAR, 1ST DIGIT
0556	CD6C05	0485		CALL ASBIN ; ASCII TO BINARY
0559	07	0486		RLCA
055A	07	0487		RLCA
055B	07	0488		RLCA
055C	07	0489		RLCA
055D	C5	0490		PUSH BC ; SAVE C REG (CKSM)
055E	4F	0491		LD C,A ; SAVE 1ST DIGIT
055F	CD4906	0492		CALL FREAD ; 2ND DIGIT
0562	CD6C05	0493		CALL ASBIN
0565	B1	0494		OR C ; OR IN MSB'S
0566	C1	0495		POP BC ; RESTORE CKSM
0567	F5	0496		PUSH AF ; SAVE ACC

ADDR	CODE	STMT	SOURCE	STATEMENT
0568	81	0497	ADD	A,C ; ADD IN 2 DIGITS TO CKSM
0569	4F	0498	LD	C,A
056A	F1	0499	POP	AF ; RESTORE ACC
056B	C9	0500	RET	
		0501	;	
		0502	;	
		0503	;	
		0504	;	
		0505	;	ASCII TO BINARY CONVERSION
		0506	;	
		0507	;	
056C	D630	0508	ASBIN	SUB 030H
056E	FE0A	0509	CP	10
0570	F8	0510	RET	M
0571	D607	0511	SUB	7
0573	C9	0512	RET	
		0513	;	
		0514	;	
		0515	;	
		0516	;	
		0517	;	

ADDR	CODE	STMT	SOURCE	STATEMENT
		0519	;	
		0520	;	
0574	C5	0521	PASP	PUSH BC
0575	CD9B05	0522		CALL PACC
0578	CD8A05	0523		CALL SPACE
057B	CD8005	0524		CALL CRLF
057E	C1	0525		POP BC
057F	C9	0526		RET
		0527	;	
		0528	;	
		0529	;	
0580	0E0D	0530	CRLF	LD C,0DH
0582	CD5902	0531		CALL CONOUT
0585	0E0A	0532		LD C,0AH
0587	C35902	0533		JP CONOUT
		0534	;	
		0535	;	
058A	0E20	0536	SPACE	LD C, ' '
058C	C35902	0537		JP CONOUT
		0538	;	
		0539	;	
		0540	;	
		0541	;	
		0542	;	
		0543	;	PRINT TEXT
		0544	;	
		0545	;	
058F	7E	0546	PTXT	LD A,(HL) ; FETCH A BYTE
0590	FE03	0547		CP 3
0592	C8	0548		RET Z
0593	4F	0549		LD C,A
0594	CD5902	0550		CALL CONOUT
0597	23	0551		INC HL
0598	C38F05	0552		JP PTXT
		0553	;	
		0554	;	
		0555	;	PRINT ACCUMULATOR
		0556	;	
		0557	;	
059B	F5	0558	PACC	PUSH AF
059C	0F	0559		RRCA
059D	0F	0560		RRCA
059E	0F	0561		RRCA
059F	0F	0562		RRCA
05A0	CDA405	0563		CALL PRVAL
05A3	F1	0564		POP AF
		0565	;	
		0566	;	
05A4	E60F	0567	PRVAL	AND 0FH
05A6	C690	0568		ADD A,90H
05A8	27	0569		DAA
05A9	CE40	0570		ADC A,40H
05AB	27	0571		DAA
05AC	4F	0572		LD C,A
05AD	C35902	0573		JP CONOUT ; PRINT IT
		0574	;	
		0575	;	

ADDR	CODE	STMT	SOURCE	STATEMENT
		0576	;	
		0577	;	
		0578	;	CHECK FOR VALID HEX CHARACTER
		0579	;	
		0580	;	
05B0	FE30	0581	AORN	CP '0'
05B2	DAC605	0582		JP C,AORN2 ; JUMP IF < 30H
05B5	FE3A	0583		CP '9'+1
05B7	DAC405	0584		JP C,AORN1 ; JUMP IF < 3AH
05BA	FE40	0585		CP 'A'-1
05BC	DAC605	0586		JP C,AORN2 ; JUMP IF < 'A'
05BF	FE47	0587		CP 'F'+1
05C1	D2C605	0588		JP NC,AORN2 ; JUMP IF < 'G'
05C4	AF	0589	AORN1	XOR A
05C5	C9	0590		RET ; VALID DATA RET
05C6	AF	0591	AORN2	XOR A
05C7	3C	0592		INC A
05C8	C9	0593		RET ; NOT HEX CHAR
		0594	;	
		0595	;	
		0596	;	CHECK FOR TERMINATOR
		0597	;	
		0598	;	SPACE, COMMA, OR CARRIAGE RETURN
		0599	;	
		0600	;	
05C9	FE20	0601	TERMCK	CP ' '
05CB	C8	0602		RET Z
05CC	FE2C	0603		CP ', '
05CE	C8	0604		RET Z
05CF	FE2E	0605		CP '. '
05D1	CAD705	0606		JP Z,TCHK0
05D4	FE0D	0607		CP 0DH
05D6	C0	0608		RET NZ
05D7	C5	0609	TCHK0	PUSH BC
05D8	CD8005	0610		CALL CRLF
05DB	C1	0611		POP BC
05DC	AF	0612		XOR A
05DD	C9	0613		RET
		0614	;	
		0615	;	
		0616	;	SCAN FOR OPERAND FROM KEYBOARD
		0617	;	
		0618	;	
		0619	;	EXIT WITH DATA IN HL, AND TERMINATOR
		0620	;	IN C. IF VALID DATA, RETURN WITH
		0621	;	ZERO FLAG SET. B CONTAINS # OF CHARACTERS ENTERED.
		0622	;	
		0623	;	
05DE	210000	0624	KEYIN	LD HL,0
05E1	45	0625		LD B,L
05E2	CD4D02	0626	KEY1	CALL ECHO
05E5	4F	0627		LD C,A
05E6	04	0628		INC B ; INC CHAR COUNT
05E7	CDC905	0629		CALL TERMCK
05EA	C8	0630		RET Z ; IF TERMINATOR, RETURN
05EB	CDB005	0631		CALL AORN ; VALID DATA CHECK
05EE	C0	0632		RET NZ ; IF NOT RETURN
05EF	79	0633		LD A,C

ADDR	CODE	STMT	SOURCE STATEMENT
05F0	CD6C05	0634	CALL ASBIN ; CONVERT TO BINARY
05F3	29	0635	ADD HL,HL
05F4	29	0636	ADD HL,HL
05F5	29	0637	ADD HL,HL
05F6	29	0638	ADD HL,HL ; SHIFT 4 BITS
05F7	85	0639	ADD A,L
05F8	6F	0640	LD L,A
05F9	C3E205	0641	JP KEY1
		0642	;
		0643	;
		0644	;
		0645	;
05FC	0E3F	0646	INVCMD LD C,'?'
05FE	CD5902	0647	CALL CONOUT
0601	3E2E	0648	LD A,'.'
0603	328006	0649	LD (NXCHR),A
0606	C9	0650	RET
		0651	;
		0652	;
		0653	;
		0654	;
		0655	;SCAN -- SCANS THE OPERANDS INPUT FROM THE
		0656	; CONSOLE. THE OPERANDS ARE SAVED IN OPI,
		0657	; AS THE ARGUMENTS TO EXECUTE THEIR COMMANDS.
		0658	;
0607	AF	0659	SCAN XOR A ;INITIALIZE
0608	218506	0660	LD HL,OPRS ;CLEAR OPERANDS
060B	E5	0661	PUSH HL
060C	DDE1	0662	POP IX ;IX=POINTER TO OPERANDS
060E	77	0663	LD (HL),A
060F	010900	0664	LD BC,9
0612	118606	0665	LD DE,OPRS+1
0615	EDB0	0666	LDIR
0617	327F06	0667	LD (OPCNT),A
061A	CDDE05	0668	SCAN1 CALL KEYIN ; GET ONE OPERAND
061D	C2FC05	0669	JP NZ,INVCMD
0620	79	0670	LD A,C
0621	328006	0671	LD (NXCHR),A
0624	FE20	0672	CP ','
0626	2806	0673	JR Z,SCAN2-\$
0628	FE2C	0674	CP ','
062A	2802	0675	JR Z,SCAN2-\$
062C	05	0676	DEC B
062D	C8	0677	RET Z ; IF NO DATA, RET
062E	DD7500	0678	SCAN2 LD (IX),L
0631	DD7401	0679	LD (IX+1),H
0634	3A7F06	0680	LD A,(OPCNT)
0637	3C	0681	INC A
0638	327F06	0682	LD (OPCNT),A
063B	DD23	0683	INC IX
063D	DD23	0684	INC IX ; POINT TO NEXT OPR
063F	79	0685	LD A,C
0640	FE20	0686	CP ','
0642	28D6	0687	JR Z,SCAN1-\$
0644	FE2C	0688	CP ','
0646	28D2	0689	JR Z,SCAN1-\$
0648	C9	0690	RET
		0691	;



ADDR	CODE	STMT	SOURCE	STATEMENT
		0692	;	
		0693	;	
0649	E5	0694	FREAD	PUSH HL
064A	D5	0695		PUSH DE
064B	C5	0696		PUSH BC
064C	3A8206	0697		LD A,(BCOUNT)
064F	FE80	0698		CP NBYTES
0651	200C	0699		JR NZ,FREAD1-\$
0653	0E14	0700		LD C,READ
0655	115C00	0701		LD DE,FCB
0658	CD0500	0702		CALL FDOS ; LEAD FILE
065B	B7	0703		OR A
065C	2011	0704		JR NZ,FERROR-\$
065E	AF	0705		XOR A
065F	218000	0706	FREAD1	LD HL,BUFFER
0662	5F	0707		LD E,A
0663	3C	0708		INC A
0664	328206	0709		LD (BCOUNT),A
0667	1600	0710		LD D,0
0669	19	0711		ADD HL,DE ; POINT TO NEXT DATA
066A	7E.	0712		LD A,(HL)
066B	C1	0713		POP BC
066C	D1	0714		POP DE
066D	E1	0715		POP HL
066E	C9	0716		RET
066F	C1	0717	FERROR	POP BC
0670	D1	0718		POP DE
0671	E1	0719		POP HL
0672	219203	0720		LD HL,MSG11
0675	CD8F05	0721		CALL PTXT ; ATTEMPT TO READ PAST EOF
0678	C36602	0722		JP RENTRY
		0723	;	
>067B		0724	DLAY	DEFS 1
>067C		0725	SCR1	DEFS 1
>067D		0726	SCR2	DEFS 1
>067E		0727	SCR3	DEFS 1
>067F		0728	OPCNT	DEFS 1
>0680		0729	NXCHR	DEFS 1
>0681		0730	FOPEN	DEFS 1
>0682		0731	BCOUNT	DEFS 1
>0683		0732	LAST	DEFS 2
>0685		0733	OPRS	DEFS 6
>0685		0734	OPR1	EQU OPRS
>0687		0735	OPR2	EQU OPRS+2
>0689		0736	OPR3	EQU OPRS+4
>0785		0737	SPVAL	EQU OPRS+100H
		0738	;	
		0739	;	
		0740	;	

ADDR CODE STMT SOURCE STATEMENT

CROSS REFERENCE LISTING

SYMBOL	VALUE	TYPE	STMT	STATEMENT REFERENCES								
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----
ADCMP	0439		0309	0374	0365	0337	0296					
AORN	05B0		0581	0631								
AORN1	05C4		0589	0584								
AORN2	05C6		0591	0588	0586	0582						
ASBIN	056C		0508	0634	0493	0485						
BCOUNT	0682		0731	0709	0697	0058						
BUFFER	0080		0022	0706	0084							
CLOSE	0010		0024	0230								
CONIN	024D		0203	0384	0362	0049						
CONOUT	0259		0213	0647	0573	0550	0537	0533	0531	0463	0459	
CONST	0240		0191	0382	0360							
CRLF	0580		0530	0610	0524	0466	0356	0283	0278	0226	0148	
				0143	0096	0071	0064					
DATA	0068		0016	0434	0404	0344						
DELAY	04FE		0418	0421	0411							
DELAY1	0500		0419	0419								
DLAY	067B		0724	0408	0056							
ECHO	024D		0202	0626	0346	0178						
EXIT	0000		0014	0233	0229							
FCB	005C		0021	0701	0231	0076						
FDOS	0005		0020	0702	0232	0085	0077					
FERROR	066F		0717	0704								
FOPEN	0681		0730	0227	0072	0060						
FREAD	0649		0694	0492	0484	0159						
FREAD1	065F		0706	0699								
HACTL	006A		0018	0433	0415	0407	0402	0368	0354	0331	0299	
				0291	0225	0063						
INVCMD	05FC		0646	0669								
KEY1	05E2		0626	0641								
KEYIN	05DE		0624	0668								
LADD	0069		0017	0442								
LAST	0683		0732	0141	0115							
LOOPS	0064		0028									
MSG1	0286		0243	0046								
MSG10	0383		0260	0080								
MSG11	0392		0262	0720								
MSG12	03B6		0264	0092								
MSG13	03CB		0266	0128								
MSG14	03E8		0268	0139								
MSG2	02EB		0250	0065								
MSG3	0307		0252	0086								
MSG4	0323		0254	0144								
MSG5	0342		0256	0284	0149							
MSG6	0367		0258	0279								
MSG7	0534		0472	0341								
MSG8	0541		0474	0379								
MSG9	054B		0476	0390								
NBYTES	0080		0027	0698	0057							
NXCHR	0680		0729	0671	0649	0287	0152	0089				
OPCNT	067F		0728	0682	0680	0667	0100					
OPEN	000F		0023	0075								
OPR1	0685		0734	0369	0357	0332	0292	0099	0094			
OPR2	0687		0735	0316	0133							



ADDR	CODE	STMT	SOURCE	STATEMENT					
TCHK0	05D7	0609	0606						
TERMCK	05C9	0601	0629						
THDR	020D	0159	0161	0109	0097				
THDR1	0229	0174	0166						
TWDG	0553	0484	0172	0170	0168	0164	0123	0113	
YORN	022D	0178	0281	0186	0146	0069			

ERRORS=0000

```

0002          NAME      PROM32
0003 ;
0004 ;
0005 ; 2732 PROM PROGRAMMER CONTROL PROGRAM
0006 ; SDOS ENVIRONMENT
0007 ; 9/27/79
0008 ;
0009 ; VERSION 1.3          CP/M COMPATIBLE          8/26/80
0010 ;
0011 ;
0012 ; SYSTEM EQUATES
0013 ;
0014 ;
>0000 0015 EXIT      EQU      0H
>0100 0016 TBASE   EQU      100H      ; BASE OF TRANSIENT AREA
>0068 0017 DATA   EQU      68H      ; DATA PORT
>0069 0018 LADD    EQU      69H      ; LOW ADDRESS PORT
>006A 0019 HACTL   EQU      6AH      ; HIGH ADDRESS + CONTROL PORT
>0000 0020 RBOUT   EQU      0
>0005 0021 FDOS    EQU      5
>005C 0022 FCB     EQU      5CH
>0080 0023 BUFFER  EQU      80H
>000F 0024 OPEN    EQU      15
>0010 0025 CLOSE   EQU      16
>0014 0026 READ    EQU      20
>001A 0027 SETDMA  EQU      26
>0080 0028 NBYTES  EQU      128
>0064 0029 LOOPS   EQU      100
0030 ;
0031 ;
0032 ;
0033 ;
0034 ;
0035 ;
0036          PSECT    ABS
0037 ;
>0100 0038          ORG      TBASE      ; START AT 100H
0039 ;
0040 ;
0041 ;
0042 ; PROGRAM PROM ENTRY POINT
0043 ;
0044 ;
0045 ;
0100 318207 0046 PPG     LD      SP,SPVAL
0103 218802 0047       LD      HL,MSG1
0106 CD8C05 0048       CALL   PTXT      ; "PROM32 VERSION 1.3"
0109 1E25    0049       LD      E,25H
010B CD4F02 0050       CALL   CONIN
010E FE31    0051       CP      31H
0110 3806    0052       JR      C,SETDLY
0112 1E32    0053       LD      E,32H
0114 2802    0054       JR      Z,SETDLY
0116 1E4A    0055       LD      E,4AH
0118 7B      0056 SETDLY: LD      A,E
0119 327806 0057       LD      (DLAY),A
011C 3E80    0058       LD      A,NBYTES
011E 327F06 0059       LD      (BCOUNT),A
  
```

```

0121 AF          0060      XOR      A
0122 327E06     0061      LD       (FOPEN),A
0125 318207     0062 PPG0    LD       SP,SPVAL
0128 AF          0063      XOR      A          ; CLEAR ACC
0129 F620       0064      OR       20H        ; STROBE OFF
012B D36A        0065      OUT      (HACTL),A
012D CD7D05     0066      CALL    CRLF
0130 21EB02     0067      LD       HL,MSG2
0133 CD8C05     0068      CALL    PTXT      ; "READY TO LOAD A FILE ?"
0136 210000     0069      LD       HL,0
0139 227906     0070      LD       (SCR1),HL ;CLEAR BYTE COUNTER
013C CD2F02     0071      CALL    YORN
013F D2FD03     0072      JP       NC,READ2 ; IF NOT,CHK READ
0142 CD7D05     0073 PPG0B   CALL    CRLF
0145 3A7E06     0074      LD       A,(FOPEN)
0148 B7          0075      OR       A
0149 2015       0076      JR      NZ,PPG0C-$
014B 0E0F       0077      LD       C,OPEN
014D 115C00     0078      LD       DE,FCB
0150 CD0500     0079      CALL    FDOS
0153 FEFF       0080      CP       255
0155 2009       0081      JR      NZ,PPG0C-$
0157 218303     0082      LD       HL,MSG10
015A CD8C05     0083      CALL    PTXT
015D C36802     0084      JP       RENTRY
0160 0E1A       0085 PPG0C   LD       C,SETDMA
0162 118000     0086      LD       DE,BUFFER
0165 CD0500     0087      CALL    FDOS
0168 210703     0088      LD       HL,MSG3
016B CD8C05     0089      CALL    PTXT      ; "HEX LOAD ADDRESS, # BYTES"
016E CD0406     0090      CALL    SCAN      ;COLLECT DATA
0171 3A7D06     0091      LD       A,(NXCHR)
0174 FE2E       0092      CP       '.'
0176 28CA       0093      JR      Z,PPG0B-$
0178 21B603     0094      LD       HL,MSG12
017B CD8C05     0095      CALL    PTXT
017E 2A8206     0096      LD       HL,(OPR1)
0181 CD8002     0097      CALL    PADDR
0184 CD7D05     0098      CALL    CRLF
0187 CD0F02     0099      CALL    THDR      ; READ ADDR/#BYTES 1ST LINE
018A E5          0100      PUSH    HL
018B ED5B8206   0101      LD       DE,(OPR1) ; LOAD ADDRESS ENTERED
018F 3A7C06     0102      LD       A,(OPCNT)
0192 A7          0103      AND     A
0193 2804       0104      JR      Z,PPG0A-$ ; IF NO OPR'S JUMP
0195 ED52       0105      SBC    HL,DE      ; COMPUTE OFFSET
0197 E5          0106      PUSH    HL
0198 D1          0107      POP     DE        ; OFFSET REG
0199 E1          0108 PPG0A   POP     HL
019A ED537B06   0109      LD       (SCR3),DE ; SAVE OFFSET
019E 1803       0110      JR      PPG2-$
01A0 CD0F02     0111 PPG1    CALL    THDR      ; LOAD ADDR/# BYTES
01A3 ED5B7B06   0112 PPG2    LD       DE,(SCR3)
01A7 B7          0113      OR       A          ; CLEAR CARRY
01A8 ED52       0114      SBC    HL,DE      ; SUBTRACT OFFSET
01AA CD5005     0115 PPG3    CALL    TWDG      ; READ A BYTE
01AD 77         0116      LD       (HL),A    ; STORE IT
01AE 228006     0117      LD       (LAST),HL
  
```

ADDR	CODE	STMT	SOURCE	STATEMENT
01B1	23	0118		INC HL
01B2	E5	0119		PUSH HL
01B3	2A7906	0120		LD HL,(SCR1) ; BYTE COUNTER
01B6	23	0121		INC HL
01B7	227906	0122		LD (SCR1),HL
01BA	E1	0123		POP HL
01BB	10ED	0124		DJNZ PPG3-\$ ; READ ENTIRE LINE
01BD	CD5005	0125		CALL TWDG ; READ CHECKSUM
01C0	AF	0126		XOR A
01C1	81	0127		ADD A,C
01C2	280B	0128		JR Z,PPG4-\$
01C4	E5	0129		PUSH HL
01C5	21CB03	0130		LD HL,MSG13
01C8	CD8C05	0131		CALL PTXT
01CB	CD8002	0132		CALL PADDR ; CHECKSUM ERROR
01CE	E1	0133		POP HL
01CF	2A7906	0134	PPG4	LD HL,(SCR1)
01D2	ED4B8406	0135		LD BC,(OPR2) ; # OF BYTES TO READ
01D6	2B	0136		DEC HL ; IF 0,THEN FFFF
01D7	0B	0137		DEC BC
01D8	A7	0138		AND A
01D9	ED42	0139		SBC HL,BC
01DB	DAA001	0140		JP C,PPG1 ; IF MORE,JUMP
01DE	21E803	0141	PPG10	LD HL,MSG14
01E1	CD8C05	0142		CALL PTXT
01E4	2A8006	0143		LD HL,(LAST)
01E7	CD8002	0144		CALL PADDR
01EA	CD7D05	0145	PPG10A	CALL CRLF
01ED	212303	0146		LD HL,MSG4
01F0	CD8C05	0147		CALL PTXT ; "READY TO PROGRAM A PROM ?"
01F3	CD2F02	0148		CALL YORN
01F6	D26802	0149		JP NC,REENTRY
01F9	CD7D05	0150	PPG11	CALL CRLF
01FC	214203	0151		LD HL,MSG5
01FF	CD8C05	0152		CALL PTXT ; "MEM START,MEM END,PROM START"
0202	CD0406	0153		CALL SCAN
0205	3A7D06	0154		LD A,(NXCHR)
0208	FE2E	0155		CP '.' ; IF '.' EXIT
020A	28ED	0156		JR Z,PPG11-\$
020C	C33D04	0157		JP PROG
		0158		;
		0159		;
		0160		;
020F	CD4606	0161	THDR	CALL FREAD ; LOOK FOR ':'
0212	FE3A	0162		CP ':'
0214	20F9	0163		JR NZ,THDR-\$
0216	AF	0164		XOR A ; CLEAR CHECKSUM
0217	4F	0165		LD C,A
0218	CD5005	0166		CALL TWDG
021B	A7	0167		AND A ; IS RECORD LENGTH 0?
021C	280D	0168		JR Z,THDR1-\$
021E	47	0169		LD B,A
021F	CD5005	0170		CALL TWDG ; GET FIRST BYTE OF ADDRESS
0222	67	0171		LD H,A
0223	CD5005	0172		CALL TWDG ; GET SECOND BYTE
0226	6F	0173		LD L,A
0227	CD5005	0174		CALL TWDG ; GET RECORD TYPE
022A	C9	0175		RET

ADDR	CODE	STMT	SOURCE	STATEMENT	
022B	F1	0176	THDRI	POP	AF
022C	C3DE01	0177		JP	PPG10
		0178		;	
		0179		;	
022F	CD4F02	0180	YORN	CALL	ECHO
0232	4F	0181		LD	C,A
0233	FE2E	0182		CP	'.'
0235	CA6802	0183		JP	Z,REENTRY
0238	B7	0184		OR	A
0239	FE4E	0185		CP	'N'
023B	C8	0186		RET	Z
023C	FE59	0187		CP	'Y'
023E	20EF	0188		JR	NZ,YORN-\$
0240	37	0189		SCF	
0241	C9	0190		RET	
		0191		;	
		0192		;	
0242	C5	0193	CONST	PUSH	BC ; CALL SDOS CONSOLE STATUS
0243	D5	0194		PUSH	DE
0244	E5	0195		PUSH	HL
0245	0E0B	0196		LD	C,0BH
0247	CD0500	0197		CALL	5
024A	B7	0198		OR	A
024B	E1	0199		POP	HL
024C	D1	0200		POP	DE
024D	C1	0201		POP	BC
024E	C9	0202		RET	
		0203		;	
>024F		0204	ECHO:		
024F	C5	0205	CONIN	PUSH	BC ; CALL SDOS CONSOLE INPUT
0250	D5	0206		PUSH	DE
0251	E5	0207		PUSH	HL
0252	0E01	0208		LD	C,1
0254	CD0500	0209		CALL	5
0257	E1	0210		POP	HL
0258	D1	0211		POP	DE
0259	C1	0212		POP	BC
025A	C9	0213		RET	
		0214		;	
025B	D5	0215	CONOUT:	PUSH	DE
025C	C5	0216		PUSH	BC
025D	E5	0217		PUSH	HL
025E	59	0218		LD	E,C
025F	0E02	0219		LD	C,02
0261	CD0500	0220		CALL	5
0264	E1	0221		POP	HL
0265	C1	0222		POP	BC
0266	D1	0223		POP	DE
0267	C9	0224		RET	
		0225		;	
0268	AF	0226	REENTRY	XOR	A
0269	D36A	0227		OUT	(HACTL),A
026B	CD7D05	0228		CALL	CRLF
026E	3A7E06	0229		LD	A,(FOPEN)
0271	B7	0230		OR	A
0272	CA0000	0231		JP	Z,EXIT
0275	0E10	0232		LD	C,CLOSE
0277	115C00	0233		LD	DE,FCB



PROM32 ADDR	4K EPROM CODE	PROGRAMMING STMT	SOURCE	STATEMENT	SD SYSTEMS Z80 ASSEMBLER PAGE 0005
027A	CD0500	0234		CALL	FDOS
027D	C30000	0235		JP	EXIT
		0236 ;			
		0237 ;			
0280	7C	0238	PADDR	LD	A,H
0281	CD9805	0239		CALL	PACC
0284	7D	0240		LD	A,L
0285	C37105	0241		JP	PASP
		0242 ;			
		0243 ;			
		0244 ;			
0288	50524F4D 2D333220 56455253 494F4E20 312E33	0245	MSG1	DEFM	'PROM-32 VERSION 1.3'
029B	0D0A0D0A	0246		DEFW	0A0DH,0A0DH
029F	49532054 48495320 53595354 454D2052 554E4E49 4E472032 2E4F2C20 322E352C 204F5220 342E3020 4D485A3F	0247		DEFM	'IS THIS SYSTEM RUNNING 2.0, 2.5, OR 4.0
02CB	0D0A	0248		DEFW	0A0DH
02CD	28322E30 202D2030 2C20322E 35202D20 312C2034 2E30202D 203229	0249		DEFM	'(2.0 - 0, 2.5 - 1, 4.0 - 2)'
02E8	0A0D	0250		DEFW	0D0AH
02EA	03	0251		DEFB	03H
02EB	52454144 5920544F 204C4F41 44204120 46494C45 2028592F 4E293F	0252	MSG2	DEFM	'READY TO LOAD A FILE (Y/N)?'
0306	03	0253		DEFB	03H
0307	48455820 4C4F4144 20414444 52455353 2C202320 42595445 533A20	0254	MSG3	DEFM	'HEX LOAD ADDRESS, # BYTES: '
0322	03	0255		DEFB	03H
0323	52454144 5920544F 2050524F 4752414D	0256	MSG4	DEFM	'READY TO PROGRAM A PROM (Y/N)?'

```

20412050
524F4D20
28592F4E
293F
0341 03 0257 DEF 03H
0342 4D454D4F 0258 MSG5 DEF 'MEMORY START, MEMORY END, PROM START: '
52592053
54415254
2C4D454D
4F525920
454E442C
50524F4D
20535441
52543A20
0366 03 0259 DEF 03H
0367 52454144 0260 MSG6 DEF 'READY TO READ A PROM (Y/N)?'
5920544F
20524541
44204120
50524F4D
2028592F
4E293F
0382 03 0261 DEF 03H
0383 46494C45 0262 MSG10 DEF 'FILE NOT FOUND'
204E4F54
20464F55
4E44
0391 03 0263 DEF 03H
0392 41545445 0264 MSG11 DEF 'ATTEMPTING TO READ PAST END OF FILE'
4D505449
4E472054
4F205245
41442050
41535420
454E4420
4F462046
494C45
03B5 03 0265 DEF 03H
03B6 4C4F4144 0266 MSG12 DEF 'LOAD START ADDRESS= '
20535441
52542041
44445245
53533D20
03CA 03 0267 DEF 03H
03CB 43484543 0268 MSG13 DEF 'CHECKSUM ERROR AT LOCATIONS '
4B53554D
20455252
4F522041
54204C4F
43415449
4F4E5320
03E7 03 0269 DEF 03H
03E8 4C4F4144 0270 MSG14 DEF 'LOAD END ADDRESS= '
2020454E
44202041
44445245
53533D20
03FC 03 0271 DEF 03H
  
```

```

0272 ;
0273 ;
0274 ;
0275 ;
0276 ; PROM READ SEQUENCE
0277 ;
0278 ;
0279 ;
03FD CD7D05 0280 READ2 CALL CRLF ; READ PROM INTO MEMORY
0400 216703 0281 LD HL,MSG6
0403 CD8C05 0282 CALL PTXT ; "READY TO READ (Y/N)?"
0406 CD2F02 0283 CALL YORN
0409 D2EA01 0284 JP NC,PPG10A ; NO, THEN PROGRAM
040C CD7D05 0285 READ0 CALL CRLF
040F 214203 0286 LD HL,MSG5
0412 CD8C05 0287 CALL PTXT ; "MEM START, MEM END, PROM START"
0415 CD0406 0288 CALL SCAN ; INPUT PARAMETERS
0418 3A7D06 0289 LD A,(NXCHR)
041B FE2E 0290 CP '.'
041D 28ED 0291 JR Z,READ0-$
041F CDF004 0292 CALL RDSET
0422 CDFF04 0293 READ1 CALL PREAD ; READ A BYTE OF PROM
0425 77 0294 LD (HL),A ; SAVE BYTE IN MEM
0426 CD2E04 0295 CALL ADCMP ; INC ADDR & COMPARE
0429 20F7 0296 JR NZ,READ1-$ ; IF NOT END, THEN JUMP
042B C3EA01 0297 JP PPG10A ; WHEN COMPLETE, RETURN
0298 ;
0299 ;
0300 ;
0301 ;
0302 ; INC HL & DE THEN COMPARE HL WITH (IX)
0303 ; EXIT WITH ZERO FLAG SET IF EQUAL
0304 ;
0305 ;
042E 13 0306 ADCMP INC DE ; INC PROM ADDR
042F 23 0307 INC HL
0430 D5 0308 PUSH DE
0431 E5 0309 PUSH HL ; SAVE HL & DE
0432 ED5B8406 0310 LD DE,(OPR2) ; GET RAM END ADDR
0436 13 0311 INC DE
0437 A7 0312 AND A
0438 ED52 0313 SBC HL,DE ; COMPARE
043A E1 0314 POP HL
043B D1 0315 POP DE
043C C9 0316 RET
0317 ;
0318 ;
0319 ;
0320 ;
0321 ; PROGRAMMING SEQUENCE
0322 ;
0323 ;
043D CDF004 0324 PROG: CALL RDSET
0440 CDFF04 0325 PROG1 CALL PREAD ; READ A BYTE (PROM)
0443 FEFF 0326 CP OFFH ; CHECK FOR ERASED BYTES
0445 2007 0327 JR NZ,PROG2-$ ; IF NOT ERASED JUMP
0447 CD2E04 0328 PRG1A CALL ADCMP ; INC ADDR & CHECK FOR END
044A 20F4 0329 JR NZ,PROG1-$ ; LOOP UNTIL END

```

ADDR	CODE	STMT	SOURCE	STATEMENT
044C	1822	0330		JR PROG3-\$ ; GO PROGRAM FROM
044E	F5	0331	PROG2	PUSH AF
044F	E5	0332		PUSH HL
0450	213105	0333		LD HL,MSG7
0453	CD8C05	0334		CALL PTXT ; "NOT ERASED"
0456	E1	0335		POP HL
0457	F1	0336		POP AF
0458	CD1005	0337		CALL PRTER ; PRINT BAD LOCATION
045B	CD4F02	0338		CALL ECHO ; READ FROM CONSOLE
045E	4F	0339		LD C,A
045F	FE2E	0340		CP '.' ; CHECK FOR PERIOD
0461	2806	0341		JR Z,PRG2A-\$ ; EXIT
0463	FE43	0342		CP 'C' ; CHECK FOR 'C'
0465	2809	0343		JR Z,PROG3-\$ ; CONTINUE IF 'C' ENTERED
0467	18DE	0344		JR PRG1A-\$ ; IF NOT ,PRINT NEXT ERROR
0469	3E20	0345	PRG2A	LD A,20H
046B	D36A	0346		OUT (HACTL),A
046D	C32501	0347		JP PPG0
0470	CD7D05	0348	PROG3	CALL CRLF
0473	2A8206	0349	PROG4	LD HL,(OPR1) ; SET MEM START ADDR
0476	ED5B8606	0350		LD DE,(OPR3) ; SET PROM START ADDR
047A	7A	0351		LD A,D
047B	F6E0	0352		OR 0E0H
047D	57	0353		LD D,A
047E	7A	0354	PROG5:	LD A,D
047F	CBF7	0355		SET 6,A
0481	D36A	0356		OUT (HACTL),A
0483	CDCE04	0357		CALL PPROG ; PROGRAM ONE BYTE
0486	CD4202	0358		CALL CONST
0489	2807	0359		JR Z,PROG5A-\$
048B	CD4F02	0360		CALL CONIN
048E	FE2E	0361		CP '.'
0490	2834	0362		JR Z,PROG9-\$
0492	CD2E04	0363	PROG5A	CALL ADCMP ; INC ADDR & CHECK END
0495	20E7	0364		JR NZ,PROG5-\$ ; FINISH THIS LOOP
0497	3E20	0365	PROG6	LD A,20H
0499	D36A	0366		OUT (HACTL),A ; TURN OFF ALL CONTROL LINES
049B	CDF004	0367	PROG6A:	CALL RDSET
049E	CDFE04	0368	PROG7	CALL PREAD ; READ PROM DATA
04A1	BE	0369		CP (HL)
04A2	2007	0370		JR NZ,PROG8-\$ ; IF NOT MATCH,JUMP
04A4	CD2E04	0371	PRG7A	CALL ADCMP ; CHECK FOR END
04A7	20F5	0372		JR NZ,PROG7-\$
04A9	18BE	0373		JR PRG2A-\$ ; EXIT,COMPLETE
04AB	F5	0374	PROG8	PUSH AF
04AC	E5	0375		PUSH HL
04AD	213E05	0376		LD HL,MSG8
04B0	CD8C05	0377		CALL PTXT ; "BAD LOCATION"
04B3	E1	0378		POP HL
04B4	CD4202	0379		CALL CONST
04B7	2807	0380		JR Z,PROG8A-\$
04B9	CD4F02	0381		CALL CONIN
04BC	FE2E	0382		CP '.'
04BE	28A9	0383		JR Z,PRG2A-\$
04C0	F1	0384	PROG8A	POP AF
04C1	CD1005	0385		CALL PRTER ; PRINT ERROR MESSAGE
04C4	18DE	0386		JR PRG7A-\$ ; PRINT ALL BAD LOCATIONS
04C6	214805	0387	PROG9	LD HL,MSG9

```

04C9 CD8C05 0388 CALL PTXT ; ABORTED
04CC 189B 0389 JR PRG2A-$
0390 ;
0391 ;
0392 ;
0393 ; PULSE ONE PROM LOCATION
0394 ;
0395 ;
04CE 7B 0396 PPROG: LD A,E
04CF D369 0397 OUT (LADD),A ; LATCH LSB OF ADDR
04D1 7A 0398 LD A,D
04D2 D36A 0399 OUT (HACTL),A
04D4 7E 0400 LD A,(HL) ; FETCH DATA
04D5 D368 0401 OUT (DATA),A ; OUTPUT DATA
04D7 7A 0402 LD A,D
04D8 CBAF 0403 RES 5,A ; TURN ON PROG PULSE
04DA D36A 0404 OUT (HACTL),A
04DC 3A7806 0405 LD A,(DLAY) ; 2.0 MHZ - 25H
0406 ; 2.5 MHZ - 32H
0407 ; 4.0 MHZ - 4AH
04DF CDE804 0408 CALL DELAY ; 50 MILLISEC DELAY
04E2 7A 0409 LD A,D
04E3 CBEF 0410 SET 5,A ; PROGRAM PULSE OFF
04E5 D36A 0411 OUT (HACTL),A
04E7 C9 0412 RET
0413 ;
04E8 06BF 0414 DELAY LD B,191
04EA 10FE 0415 DELAY1 DJNZ DELAY1-$
04EC 3D 0416 DEC A
04ED 20F9 0417 JR NZ,DELAY-$
04EF C9 0418 RET
0419 ;
0420 ;
04F0 2A8206 0421 RDSET: LD HL,(OPR1)
04F3 ED5B8606 0422 LD DE,(OPR3)
04F7 7A 0423 LD A,D
04F8 E60F 0424 AND 0FH
04FA CBEF 0425 SET 5,A
04FC 57 0426 LD D,A
04FD C9 0427 RET
0428 ;
0429 ;
0430 ; READ ONE LOCATION OF PROM
0431 ;
0432 ;
04FE 7B 0433 PREAD: LD A,E
04FF D369 0434 OUT (LADD),A ; LATCH LSB OF ADDR
0501 7A 0435 LD A,D
0502 D36A 0436 OUT (HACTL),A
0504 CBAF 0437 RES 5,A
0506 D36A 0438 OUT (HACTL),A
0508 DB68 0439 IN A,(DATA)
050A F5 0440 PUSH AF
050B 7A 0441 LD A,D
050C D36A 0442 OUT (HACTL),A
050E F1 0443 POP AF
050F C9 0444 RET
0445 ;

```

```

0446 ; PRINTS ADDR/BAD DATA/GOOD DATA
0447 ; OF ERROR LOCATION
0448 ;
0449 ;
0510 47 0450 PRTER LD B,A
0511 7A 0451 LD A,D
0512 E60F 0452 AND 0FH
0514 CD9805 0453 CALL PACC ; PRINT MSB OF ADDR
0517 7B 0454 LD A,E
0518 CD9805 0455 CALL PACC ; PRINT LSB OF ADDR
051B 0E20 0456 LD C,20H
051D CD5B02 0457 CALL CONOUT
0520 78 0458 LD A,B
0521 CD9805 0459 CALL PACC ; PRINT BAD DATA
0524 0E20 0460 LD C,20H
0526 CD5B02 0461 CALL CONOUT
0529 7E 0462 LD A,(HL)
052A CD9805 0463 CALL PACC ; PRINT GOOD DATA
052D CD7D05 0464 CALL CRLF
0530 C9 0465 RET
0466 ;
0467 ;
0468 ;
0469 ;
0531 4E4F5420 0470 MSG7 DEFM 'NOT ERASED '
45524153
45442020
053D 03 0471 DEFB 03H
053E 42414420 0472 MSG8 DEFM 'BAD LOC '
4C4F4320
20
0547 03 0473 DEFB 03H
0548 41424F52 0474 MSG9 DEFM 'ABORTED'
544544
054F 03 0475 DEFB 03H
0476 ;
0477 ;
0478 ;
0479 ; TWDG--READ AND CONVERT 2 DIGITS TO BINARY
0480 ;
0481 ;
0550 CD4606 0482 TWDG CALL FREAD ; READ A CHAR, 1ST DIGIT
0553 CD6905 0483 CALL ASBIN ; ASCII TO BINARY
0556 07 0484 RLCA
0557 07 0485 RLCA
0558 07 0486 RLCA
0559 07 0487 RLCA
055A C5 0488 PUSH BC ; SAVE C REG (CKSM)
055B 4F 0489 LD C,A ; SAVE 1ST DIGIT
055C CD4606 0490 CALL FREAD ; 2ND DIGIT
055F CD6905 0491 CALL ASBIN
0562 B1 0492 OR C ; OR IN MSB'S
0563 C1 0493 POP BC ; RESTORE CKSM
0564 F5 0494 PUSH AF ; SAVE ACC
0565 81 0495 ADD A,C ; ADD IN 2 DIGITS TO CKSM
0566 4F 0496 LD C,A
0567 F1 0497 POP AF ; RESTORE ACC
0568 C9 0498 RET
  
```

ADDR	CODE	STMT	SOURCE	STATEMENT
		0499	;	
		0500	;	
		0501	;	
		0502	;	
		0503	;	ASCII TO BINARY CONVERSION
		0504	;	
		0505	;	
0569	D630	0506	ASBIN	SUB 030H
056B	FE0A	0507		CP 10
056D	F8	0508		RET M
056E	D607	0509		SUB 7
0570	C9	0510		RET
		0511	;	
		0512	;	
		0513	;	
		0514	;	
		0515	;	

ADDR	CODE	STMT	SOURCE	STATEMENT
		0517	;	
		0518	;	
0571	C5	0519	PASP	PUSH BC
0572	CD9805	0520		CALL PACC
0575	CD8705	0521		CALL SPACE
0578	CD7D05	0522		CALL CRLF
057B	C1	0523		POP BC
057C	C9	0524		RET
		0525	;	
		0526	;	
		0527	;	
057D	0E0D	0528	CRLF	LD C,0DH
057F	CD5B02	0529		CALL CONOUT
0582	0E0A	0530		LD C,0AH
0584	C35B02	0531		JP CONOUT
		0532	;	
		0533	;	
0587	0E20	0534	SPACE	LD C,' '
0589	C35B02	0535		JP CONOUT
		0536	;	
		0537	;	
		0538	;	
		0539	;	
		0540	;	
		0541	;	PRINT TEXT
		0542	;	
		0543	;	
058C	7E	0544	PTXT	LD A,(HL) ; FETCH A BYTE
058D	FE03	0545		CP 3
058F	C8	0546		RET Z
0590	4F	0547		LD C,A
0591	CD5B02	0548		CALL CONOUT
0594	23	0549		INC HL
0595	C38C05	0550		JP PTXT
		0551	;	
		0552	;	
		0553	;	PRINT ACCUMULATOR
		0554	;	
		0555	;	
0598	F5	0556	PACC	PUSH AF
0599	0F	0557		RRCA
059A	0F	0558		RRCA
059B	0F	0559		RRCA
059C	0F	0560		RRCA
059D	CDA105	0561		CALL PRVAL
05A0	F1	0562		POP AF
		0563	;	
		0564	;	
05A1	E60F	0565	PRVAL	AND 0FH
05A3	C690	0566		ADD A,90H
05A5	27	0567		DAA
05A6	CE40	0568		ADC A,40H
05A8	27	0569		DAA
05A9	4F	0570		LD C,A
05AA	C35B02	0571		JP CONOUT ; PRINT IT
		0572	;	
		0573	;	



```

0574 ;
0575 ;
0576 ; CHECK FOR VALID HEX CHARACTER
0577 ;
0578 ;
05AD FE30 0579 AORN CP '0'
05AF DAC305 0580 JP C,AORN2 ; JUMP IF < 30H
05B2 FE3A 0581 CP '9'+1
05B4 DAC105 0582 JP C,AORN1 ; JUMP IF < 3AH
05B7 FE40 0583 CP 'A'-1
05B9 DAC305 0584 JP C,AORN2 ; JUMP IF < 'A'
05BC FE47 0585 CP 'F'+1
05BE D2C305 0586 JP NC,AORN2 ; JUMP IF < 'G'
05C1 AF 0587 AORN1 XOR A
05C2 C9 0588 RET ; VALID DATA RET
05C3 AF 0589 AORN2 XOR A
05C4 3C 0590 INC A
05C5 C9 0591 RET ; NOT HEX CHAR
0592 ;
0593 ;
0594 ; CHECK FOR TERMINATOR
0595 ;
0596 ; SPACE, COMMA, OR CARRIAGE RETURN
0597 ;
0598 ;
05C6 FE20 0599 TERMCK CP ' '
05C8 C8 0600 RET Z
05C9 FE2C 0601 CP ','
05CB C8 0602 RET Z
05CC FE2E 0603 CP '.'
05CE CAD405 0604 JP Z,TCHK0
05D1 FE0D 0605 CP 0DH
05D3 C0 0606 RET NZ
05D4 C5 0607 TCHK0 PUSH BC
05D5 CD7D05 0608 CALL CRLF
05D8 C1 0609 POP BC
05D9 AF 0610 XOR A
05DA C9 0611 RET
0612 ;
0613 ;
0614 ; SCAN FOR OPERAND FROM KEYBOARD
0615 ;
0616 ;
0617 ; EXIT WITH DATA IN HL, AND TERMINATOR
0618 ; IN C. IF VALID DATA, RETURN WITH
0619 ; ZERO FLAG SET. B CONTAINS # OF CHARACTERS ENTERED.
0620 ;
0621 ;
05DB 210000 0622 KEYIN LD HL,0
05DE 45 0623 LD B,L
05DF CD4F02 0624 KEY1 CALL ECHO
05E2 4F 0625 LD C,A
05E3 04 0626 INC B ; INC CHAR COUNT
05E4 CDC605 0627 CALL TERMCK
05E7 C8 0628 RET Z ; IF TERMINATOR, RETURN
05E8 CDAD05 0629 CALL AORN ; VALID DATA CHECK
05EB C0 0630 RET NZ ; IF NOT RETURN
05EC 79 0631 LD A,C

```

ADDR	CODE	STMT	SOURCE	STATEMENT
05ED	CD6905	0632		CALL ASBIN ; CONVERT TO BINARY
05F0	29	0633		ADD HL,HL
05F1	29	0634		ADD HL,HL
05F2	29	0635		ADD HL,HL
05F3	29	0636		ADD HL,HL ; SHIFT 4 BITS
05F4	85	0637		ADD A,L
05F5	6F	0638		LD L,A
05F6	C3DF05	0639		JP KEY1
		0640		;
		0641		;
		0642		;
		0643		;
		0644		;
		0645		;
05F9	0E3F	0646	INVCMD	LD C,'?'
05FB	CD5B02	0647		CALL CONOUT
05FE	3E2E	0648		LD A,'.'
0600	327D06	0649		LD (NXCHR),A
0603	C9	0650		RET
		0651		;
		0652		;
		0653		;
		0654		;
		0655		;SCAN -- SCANS THE OPERANDS INPUT FROM THE
		0656		; CONSOLE. THE OPERANDS ARE SAVED IN OPI,
		0657		; AS THE ARGUMENTS TO EXECUTE THEIR COMMANDS.
		0658		;
0604	AF	0659	SCAN	XOR A ;INITIALIZE
0605	218206	0660		LD HL,OPRS ;CLEAR OPERANDS
0608	E5	0661		PUSH HL
0609	DDE1	0662		POP IX ;IX=POINTER TO OPERANDS
060B	77	0663		LD (HL),A
060C	010900	0664		LD BC,9
060F	118306	0665		LD DE,OPRS+1
0612	EDB0	0666		LDIR
0614	327C06	0667		LD (OPCNT),A
0617	CDD805	0668	SCAN1	CALL KEYIN ; GET ONE OPERAND
061A	C2F905	0669		JP NZ,INVCMD
061D	79	0670		LD A,C
061E	327D06	0671		LD (NXCHR),A
0621	FE20	0672		CP ' '
0623	2806	0673		JR Z,SCAN2-\$
0625	FE2C	0674		CP ' '
0627	2802	0675		JR Z,SCAN2-\$
0629	05	0676		DEC B
062A	C8	0677		RET Z ; IF NO DATA, RET
062B	DD7500	0678	SCAN2	LD (IX),L
062E	DD7401	0679		LD (IX+1),H
0631	3A7C06	0680		LD A,(OPCNT)
0634	3C	0681		INC A
0635	327C06	0682		LD (OPCNT),A
0638	DD23	0683		INC IX
063A	DD23	0684		INC IX ; POINT TO NEXT OPR
063C	79	0685		LD A,C
063D	FE20	0686		CP ' '
063F	28D6	0687		JR Z,SCAN1-\$
0641	FE2C	0688		CP ' '
0643	28D2	0689		JR Z,SCAN1-\$

ADDR	CODE	STMT	SOURCE	STATEMENT
0645	C9	0690		RET
		0691	;	
		0692	;	
		0693	;	
0646	E5	0694	FREAD	PUSH HL
0647	D5	0695		PUSH DE
0648	C5	0696		PUSH BC
0649	3A7F06	0697		LD A,(BCOUNT)
064C	FE80	0698		CP NBYTES
064E	200C	0699		JR NZ,FREAD1-\$
0650	0E14	0700		LD C,READ
0652	115C00	0701		LD DE,FCB
0655	CD0500	0702		CALL FDOS ; LEAD FILE
0658	B7	0703		OR A
0659	2011	0704		JR NZ,FERROR-\$
065B	AF	0705		XOR A
065C	218000	0706	FREAD1	LD HL,BUFFER
065F	5F	0707		LD E,A
0660	3C	0708		INC A
0661	327F06	0709		LD (BCOUNT),A
0664	1600	0710		LD D,0
0666	19	0711		ADD HL,DE ; POINT TO NEXT DATA
0667	7E	0712		LD A,(HL)
0668	C1	0713		POP BC
0669	D1	0714		POP DE
066A	E1	0715		POP HL
066B	C9	0716		RET
066C	C1	0717	FERROR	POP BC
066D	D1	0718		POP DE
066E	E1	0719		POP HL
066F	219203	0720		LD HL,MSG11
0672	CD8C05	0721		CALL PTXT ; ATTEMPT TO READ PAST EOF
0675	C36802	0722		JP RENTRY
		0723	;	
>0678		0724	DLAY	DEFS 1
>0679		0725	SCR1	DEFS 1
>067A		0726	SCR2	DEFS 1
>067B		0727	SCR3	DEFS 1
>067C		0728	OPCNT	DEFS 1
>067D		0729	NXCHR	DEFS 1
>067E		0730	FOPEN	DEFS 1
>067F		0731	BCOUNT	DEFS 1
>0680		0732	LAST	DEFS 2
>0682		0733	OPRS	DEFS 6
>0682		0734	OPR1	EQU OPRS
>0684		0735	OPR2	EQU OPRS+2
>0686		0736	OPR3	EQU OPRS+4
>0782		0737	SPVAL	EQU OPRS+100H
		0738	;	
		0739	;	
		0740	;	

CROSS REFERENCE LISTING

SYMBOL	VALUE	TYPE	STMT	STATEMENT REFERENCES								
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----
ADCMP	042E		0306	0371	0363	0328	0295					
AORN	05AD		0579	0629								
AORN1	05C1		0587	0582								
AORN2	05C3		0589	0586	0584	0580						
ASBIN	0569		0506	0632	0491	0483						
BCOUNT	067F		0731	0709	0697	0059						
BUFFER	0080		0023	0706	0086							
CLOSE	0010		0025	0232								
CONIN	024F		0205	0381	0360	0050						
CONOUT	025B		0215	0647	0571	0548	0535	0531	0529	0461	0457	
CONST	0242		0193	0379	0358							
CRLF	057D		0528	0608	0522	0464	0348	0285	0280	0228	0150	
				0145	0098	0073	0066					
DATA	0068		0017	0439	0401							
DELAY	04E8		0414	0417	0408							
DELAY1	04EA		0415	0415								
DLAY	0678		0724	0405	0057							
ECHO	024F		0204	0624	0338	0180						
EXIT	0000		0015	0235	0231							
FCB	005C		0022	0701	0233	0078						
FDOS	0005		0021	0702	0234	0087	0079					
FERROR	066C		0717	0704								
FOPEN	067E		0730	0229	0074	0061						
FREAD	0646		0694	0490	0482	0161						
FREAD1	065C		0706	0699								
HACTL	006A		0019	0442	0438	0436	0411	0404	0399	0366	0356	
				0346	0227	0065						
INVCMD	05F9		0646	0669								
KEY1	05DF		0624	0639								
KEYIN	05DB		0622	0668								
LADD	0069		0018	0434	0397							
LAST	0680		0732	0143	0117							
LOOPS	0064		0029									
MSG1	0288		0245	0047								
MSG10	0383		0262	0082								
MSG11	0392		0264	0720								
MSG12	03B6		0266	0094								
MSG13	03CB		0268	0130								
MSG14	03E8		0270	0141								
MSG2	02EB		0252	0067								
MSG3	0307		0254	0088								
MSG4	0323		0256	0146								
MSG5	0342		0258	0286	0151							
MSG6	0367		0260	0281								
MSG7	0531		0470	0333								
MSG8	053E		0472	0376								
MSG9	0548		0474	0387								
NBYTES	0080		0028	0698	0058							
NXCHR	067D		0729	0671	0649	0289	0154	0091				
OPCNT	067C		0728	0682	0680	0667	0102					
OPEN	000F		0024	0077								
OPR1	0682		0734	0421	0349	0101	0096					
OPR2	0684		0735	0310	0135							



ADDR	CODE	STMT	SOURCE	STATEMENT					
TCHK0	05D4	0607	0604						
TERMCK	05C6	0599	0627						
THDR	020F	0161	0163	0111	0099				
THDR1	022B	0176	0168						
TWDG	0550	0482	0174	0172	0170	0166	0125	0115	
YORN	022F	0180	0283	0188	0148	0071			

ERRORS=0000